Volume 42 Number 1 March 2018 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

AI in Slovenia

Anniversary Edition

Guest Editors:

Mitja Luštrek

Jure Žabkar

Marko Grobelnik

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

Introduction to the Special Anniversary Issue on "AI in Slovenia"

Slovenian Artificial Intelligence Society (SLAIS) celebrated its 25th anniversary last year. Along with 40th anniversary of the Informatica journal and 20th Information Society conference, the idea of dedicating a special issue to commemorate these events was born. Informatica has long been a publication venue for Slovenian artificial intelligence (AI) research, one of the Information Society subconferences is dedicated to AI, and the best papers from it are traditionally published in the Informatica journal, so commemorating them jointly is very fitting. We decided to invite papers presenting current work of Slovenian AI researchers, as well as position papers providing a (historical) perspective on some AI topic.

The global research on AI has had several ups and downs through the history. It started in the 1950s with Alan Turing's seminal paper "Computing Machinery and Intelligence" and the Dartmouth conference. These events sparked a golden age of discovery, which ended in the so-called first AI winter in the 1970s. However, it was soon succeeded by another boom in the 1980s, spurred by the Japanese fifth-generation computer project. The second AI winter followed, caused by the perception that AI is not fulfilling its promise. After that, we have seen steady progress, and right now we are at another peak of success: AI is applied in all areas of life and business, with examples ranging from self-driving-cars to the games of go and chess.

As recorded by Marko Bohanec during his tenure as SLAIS chair, Slovenian AI research started in 1972, at the end of the initial golden age. It began at the Computer Science Department at Jožef Stefan Institute (JSI), and later expanded to the Faculty of Computer and Information Science (FRI) of the University of Ljubljana. Initially, AI research in Slovenia was concerned with heuristic search, including knowledge-based approaches to computer chess. The emphasis then gradually shifted and expanded to the areas of machine learning, knowledge representation, computer-aided multi-attribute decision making, qualitative reasoning and modelling, and combinatorial optimisation. This provided a solid basis for later application projects. In 1982, the development and implementation of AI tools started and soon resulted in numerous practical applications. Most of these applications were based on Assistant Professional, an inductive learning system, and DEX, a computer-aided decision making system. Later, the research encompassed practically all major areas of AI – basic and applied – some of which are sampled in the papers composing this special issue.

SLAIS was established in 1992, when Slovenian AI research was already quite extensive, as evidenced by having over 60 members one year after establishment. The membership later rose to a peak of 157. SLAIS is a member of the European Association for Artificial Intelligence (EurAI), and three of its members were elected EurAI fellows. SLAIS and Slovenian AI is in most respects firmly embedded in the European and global AI community, mainly through participation in international research projects beginning in 1990 with the first European Framework Programme. We are looking to continue along this path, as well as strengthen the collaboration and sense of community within Slovenia with efforts such as this special issue.

The first paper of the special issue is by Igor Kononenko, presenting early research on machine learning in Slovenia starting in 1982. It is followed by a paper of the AI pioneer Ivan Bratko on computer chess – a topic dating back to the very beginning of AI research and recently again brought to prominence by the success of AlphaZero. After these we have a range of papers on various topic representing most major Slovenian groups engaged in AI research. In addition to Igor Kononenko, the Laboratory for Cognitive Modelling at FRI is represented by Marko Robnik-Šikonja with a paper on explaining the predictions of machine-learning models. The AI Laboratory at FRI contributed a paper on argumentation in interactive machine learning by Martin Možina. From the AI Laboratory at JSI we have a paper on semantic annotation of documents by Janez Brank et al. The Department of Intelligent Systems at JSI is represented by two papers on AI applications in the health domain: on continuous blood pressure estimation from PPG signal by Gašper Slapničar et al., and on psychological arousal recognition from physiological signals by Martin Gjoreski et al. The Department of Knowledge Technologies at JSI is represented by three papers: on assessing the quality of feature rankings by Ivica Slavkov et al., on computational creativity by Senja Pollak et al., and on a related topic of creatively blending software components by Matej Martinc et al. Finally, we have a paper on text understanding by Jure Zupan, who does not belong to any established Slovenian AI group. We regret that we have not been able to include any paper from the Bioinformatics Laboratory at FRI or from the University of Maribor, but since we are confident in continued success of AI research in Slovenia, we trust there will be further opportunities for this.

Mitja Luštrek
Jure Žabkar
Marko Grobelnik

Early Machine Learning Research in Ljubljana

Igor Kononenko
University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, Ljubljana, Slovenia
E-mail: igor.kononenko@fri.uni-lj.si

Keywords: machine learning, decision trees, naive Bayesian classifier, ReliefF

Received: October 17, 2017

We describe early machine learning research in Ljubljana, motivated by medical diagnostic problems, in the areas of building decision trees with Assistant, the development of Naïve and Semi-Naïve Bayesian classifier and its explanations of individual predictions, and the development of ReliefF and RReliefF algorithms for non-myopic evaluation of attributes in classification and regression, respectively.

Povzetek: V članku opišemo zgodnje raziskave na področju strojnega učenja v Ljubljani, ki so bile motivirane z medicinskimi diagnostičnimi problemi. Razvili smo sistem Asistent za gradnjo odločitvenih dreves, naivni in delno naivni Bayesov klasifikator in metodo razlage njunih napovedi ter algoritma ReliefF in RReliefF za nekratkovidno ocenjevanje atributov v klasifikaciji in regresiji.

1 Introduction

As a young researcher, I started my research in Machine learning (ML) in 1982 at the University of Ljubljana and with strong connection with the Artificial Intelligence (AI) group at Jožef Stefan Institute in Ljubljana, Slovenia. My supervisor Prof. Ivan Bratko suggested to me to use Quinlan's (1979) algorithm ID3 for learning medical diagnostic rules. My first data set, obtained from Ljubljana Institute of Oncology, was a description of 339 patients with known correct locations of the primary tumor in the body out of 22 possible locations. The diagnostic task was to determine the location of the primary tumor for new patients, given the description of patients' age, sex, tumor grade, and locations of detected metastases. We tested the classification accuracy of physicians-experts and they were able to correctly classify 42% of patients. The performance of ID3 on this hard diagnostic problem was not satisfactory (lower than 40%), that is why we started to research the possible deficiencies of ID3 and search for the methodologies which would circumvent them.

At that time only few researchers applied ML to medical diagnosis, see (Kononenko, 2001) for an overview. ID3 was developed in 1979 and was not yet applied to medical diagnosis, nobody was using Naïve Bayes (Good, 1950; 1964), which was yet to be rediscovered by us and subsequently by ML community, and more advanced ML approaches, such as multilayered neural networks, support vector machines and random forests were developed much later. Therefore, building decision trees with ID3 seemed to be a good starting point. Note also that there was no internet at that time and the spreading of news about scientific development was significantly slower compared to nowadays. For example, we became aware of system CART (Breiman et al. 1984) for building classification and regression trees several years after it was published.

2 Induction of decision trees with Assistant

Our first discovery was that Information gain, used by ID3 to evaluate the quality of attributes, was biased to overestimate the multivalued attributes, so normalization was required. Another observation was, that lower levels of the tree become unreliable due to small numbers of training examples, so a kind of pruning was needed. Also, at certain level of the tree, built by ID3, a null (empty) leaves could appear, indicating that there was no corresponding training instances for such a leaf, which required a technique to classify new instances which fall in such a leaf. Yet another problem was that ID3 was not able to deal with missing values of attributes. Introduction of an additional value "unknown" for each attribute did not work well, as it led to larger trees and an additional reduction of the number of instances in the leaves.

The research resulted in the development of a new decision tree learning algorithm, called Assistant (Kononenko et al., 1984), which reached the classification accuracy of 44% in the primary tumor diagnostic task.

The reason for encouraging results is that (good) ML algorithms can model the probability distributions more accurately than human experts. On the other hand, physicians use additional information about patients which cannot be straightforwardly coded in a form suitable for ML. Therefore, the comparison of prediction performance is biased, as physicians were, for the sake of comparison, constrained to use the same information as ML algorithms. Our encouraging results motivated other researchers to apply ML in various areas of medical diagnosis, see an overview in (Kononenko, 2001).

The main five contributions of Assistant with respect to ID3 were:

2.1. *An ad-hoc normalization of the Information gain* – dividing information gain of the attribute with k possible values with $\log_2 k$ in order to prevent the overestimation of multivalued attributes. Although it improved the performance, it was ad-hoc. Ross Quinlan, inspired by our research, introduced another normalization – so called Gain-ratio in his famous system C4.5 (Quinlan, 1986), while the appropriate normalization of Information gain was introduced in ML community later with the so-called Distance measure (Mantaras, 1989).

2.2. *Using (an ad-hoc) decision tree pruning*. We introduced a parameter which indicated how many training instances should be in the leaf in order to allow further subtree building. Later, inspired by our idea, many researchers proposed various pre- and post-pruning techniques, however all of them introduced one or more parameters for controlling the strength of pruning. For example, our colleague from Jožef Stefan Institute in Ljubljana, Bojan Cestnik developed a post-pruning technique based on the *m*-estimate of probabilities (Cestnik and Bratko, 1991) which uses parameter *m* for pruning control.

We were looking for a parameter-less pruning techniques, yet without success. We needed another ten years to develop a satisfactory decision tree pre-pruning method which required no parameter setting. The method is based on the MDL-principle (Li and Vitanyi, 1993), which we first used to develop the MDL attribute evaluation method (Kononenko, 1995). The basic idea is to evaluate how compressive a (discrete) attribute is. The effectiveness of that method depends on the appropriate selection of (optimal) data coding. The same idea was later extended to parameter-less decision tree pre-pruning (Kononenko, 1998). The method evaluates how compressive the subtree is in comparison to a leaf alone (without the subtree). Again, the effectiveness of the method depends on the appropriate coding of the data and the tree structure.

2.3. *Classification in combination with the Naïve Bayesian classifier (NB) in the tree leaves.* One version of this idea is to use NB in the empty (null) leaves. This allows us to classify new instances for which no support from the training set in the corresponding leaf exists. The obvious generalization is to use NB in all leaves, allowing the classification process to efficiently use the information of attributes, not tested on the path from the root to the leaf. Later, the same idea was used by researchers who developed regression trees, where in the leaves Linear regression can be used.

2.4. *Building binary decision trees.* In order to avoid over-splitting the training data set (and also to overcome the bias of Information gain to overestimate multivalued attributes) we introduced the binarization of continuous and discrete attributes in order to build binary decision trees. Binary trees proved to be smaller and more accurate, avoiding also the so called replication problem – the appearance of more identical or similar subtrees in a non-binary decision tree.

2.5. *Dealing with incomplete data*. We introduced the methodology for dealing with missing values of attributes, by introducing the instance weights which correspond to the (conditional) probabilities that the instance with missing value has a certain attribute value. The weighted instance then follows all the branches from the current node, each with an appropriate weight. This attribute weighting was generalized to the so called "don't care" values, where any attribute value is allowed. For such an instance the weight is multiplied with the number of possible values of the attribute with "don't care" value. The methodology was later adopted as a standard way for dealing with incomplete data in decision tree learning.

Later, a reimplementation of Assistant was developed, called Assistant 86 (Cestnik et al., 1987) which was followed by a commercial system Assistant Professional.

3 Naïve Bayesian classifier

During the development of the Assistant learning algorithm, I intuitively developed a »simple statistical method«, as I called it at that time and compared its results with decision trees. The surprisingly simple method performed on the primary tumor problem equally well as Assistant did. At that time, however, we claimed that decision trees are preferable due to their "transparency", which does not hold for »statistical methods«. I knew, that my »statistical method« was ad-hoc but I was not able to formally interpret it. With the help of Prof. Bratko we realized that my ad-hoc statistical method was almost the same as the Naïve Bayesian classifier (NB), however lacking the prior probability of the class in the NB formula. (At that time we called it Simple Bayes and only at the ISSEK Workshop in Bled, Slovenia in 1984, where I for the first time presented Assistant for building decision trees, Prof. Donald Michie tossed the name "*Naïve Bayesian classifier*" – and later this name was accepted by ML community).

It turned out that the corrected NB ("statistical method" upgraded with the prior class probability) was able to significantly outperform Assistant in the primary tumor domain (reaching 50% of classification accuracy) as well as on two other medical diagnostic problems (lymphography diagnosis and the breast cancer recurrence prediction).

We became motivated to further research NB in relation to decision trees (Kononenko, 1989a), and we developed the explanation method for NB where for each attribute the amount of information for or against the class is provided in the sum of information contributions during the classification process (Kononenko, 1989b). The explanation is obtained by changing probabilities P in the NB formula into information contributions (using $-\log_2 P$). Surprisingly, this explanation turned out to be more intuitive and more transparent to physicians, who claimed that they also sum up the evidence for or against the diagnosis.

In 1988 I was listening to an inspiring talk by Prof. Igor Grabec in Ljubljana about artificial neural networks and I decided to do more research in this area. We generalized the Hopfield's (1982) discrete model into Bayesian neural networks, where each neuron in the model uses NB (Kononenko, 1989c), and later in my PhD I generalized it into continuous model. Our generalization of NB to Semi-naïve Bayes (Kononenko, 1991) motivated several researchers to try different approaches to avoid the naivety of NB.

At the same time, in his PhD, Bojan Cestnik developed the *m*-estimate of probabilities, which proved to improve the performance of NB (Cestnik, 1990).

4 ReliefF and RReliefF

In 1992 I attended the ICML conference in Aberdeen in Scotland. The audience was highly impressed by the talk of Prof. Larry Rendell, who described the algorithm RELIEF, developed by his PhD student Kira (Kira and Rendell, 1992). RELIEF is a non-myopic attribute evaluator, i.e. it is able to efficiently evaluate the quality of attributes even if there are strong interactions between attributes. This breakthrough in the field of attribute evaluation lead to the development of ReliefF algorithm (Kononenko, 1994) which was later adopted by the ML community as a standard for evaluating the attributes in classification and many improvements and adaptations of RelieF were developed. ReliefF improved RELIEF in three major directions:

1. *Dealing with noisy data.* RELIEF was sensitive to noise in the data. Instead of searching for each instance one nearest hit (nearest instance from the same class) and one nearest miss (nearest instance form the opposite class), ReliefF searches for *k* nearest hits and *k* nearest misses where *k* is a parameter, set by the user (in the same sense as *k*-NN algorithms deal with noise).
2. *Dealing with multiclass problems.* RELIEF was designed for two-class problems only. ReliefF generalizes to more than two classes by searching for *k* nearest misses from each "opposite" class and appropriately weights the contributions of nearest misses with the prior probabilities of corresponding classes.
3. *Dealing with incomplete data.* RELIEF was designed for complete data, without any missing values. While calculating the distances between instances, ReliefF calculates the contributions of attributes with missing values using the conditional probabilities of values given the class. ReliefF is able to evaluate continuous and discrete attributes for classification. Together with my PhD student Marko Robnik-Šikonja, we developed a regressional version of ReliefF, called RReliefF, which enables the evaluation the quality of discrete and continuous attributes in regression (Robnik-Šikonja and Kononenko, 1997). Note that in regression there are no hits and no misses, as instances do not belong to classes, but rather have real values of regression variable. The basic idea of RReliefF is to use the difference of two instances in regression values to model the "probability that two instances do not belong to the same class".

Together with my PhD student Uroš Pompe, we developed also a variant of Relief which enables the (non-myopic) evaluation of literals in Inductive Logic Programming (ILP) (Pompe and Kononenko, 1998). The basic idea is to make a non-symmetrical evaluation measure, biased towards "positive class", as in ILP only positive examples should be covered by good literals (only a theory for the positive class is built) and negative examples should not be covered by good literals.

5 Conclusion

Our development of ML algorithms was highly motivated by medical diagnostic problems. Our applications started in oncology and later spread to other medical areas, such as prognostics of the femoral neck fracture recovery, rheumatology, diagnosis of lower urinary tract disorders, coronary artery disease, sport injuries etc. The overview of our research of ML for medical diagnosis was described in (Kononenko, 2001), which had a great impact on scientific community. Other, earlier references, with the greatest impact on the ML community, include (Kononenko et al., 1984; Cestnik et al., 1987; Kononenko, 1991; 1994).

The unattained goals of our early ML research, *a general method for explaining individual predictions* in a similar way as the NB's explanations, and *a general method for estimating the reliability of individual predictions* of arbitrary prediction models in classification and regression, were achieved by my PhD students: the former goal by Erik Štrumbelj, and the latter goal by the work of Matjaž Kukar, Zoran Bosnić and Darko Pevec (see the overview by Kononenko et al., 2013).

6 References

[1] Breiman L., Friedman J.H., Olshen R.A., Stone C.J. (1984) Classification and Regression Trees, Wadsforth International Group.

[2] Cestnik, B., Kononenko, I., Bratko, I. ASSISTANT 86: a knowledge-elicitation tool for sophisticated users. In:Bratko, I., Lavrač, N. Progress in machine learning : proc. of European Working Session on Learning EWSL 87. Sigma Press, 1987, p. 31-45.

[3] Cestnik, B. Estimating probabilities. In: Carlucci A. L. (ed.) Proc. ECAI 90. Pitman. 1990, p.147-149.

[4] Cestnik, B., Bratko, I. On estimating probabilities in tree pruning. In: Proc. EWSL-91: European working session on learning, Porto, Portugal, March 6-8, 1991, Springer. p.138-150.

[5] Good I.J., Probability and the Weighing of Evidence. London: Charles Griffin, 1950.

[6] Good I.J., The Estimation of Probabilities - An Essay on Modern Bayesian Methods, Cambridge: The MIT Press, 1964.

[7] Hopfield. J. J. Neural networks and physical systems with emergent collective computational abilities. Nat. Academy of Sc., 79:2554–2558, 1982.

[8] Kira, K. and Rendell, L. A practical approach to feature selection. In D. Sleeman and P. Edwards, eds, Proc. ICML, Aberdeen, UK, 1992, p. 249–256.
[9] Kononenko, I. ID3, Sequential Bayes, Naive Bayes and Bayesian Neural Networks. Proc. of European Working Session on Learning EWSL 1989, Montpellier: France, Dec. 4-6, 1989a, p.91-98.
[10] Kononenko, I. Interpretation of neural networks decisions, IASTED Int. Conf. Expert systems & apps, Zurich, June 26-29 1989b, pp.224-227.
[11] Kononenko, I. Bayesian Neural Networks, Biological Cybernetics Journal 61: 361-370, 1989c.
[12] Kononenko, I. Semi-naive Bayesian classifier, Proc. of European Working Session on Learning EWSL-91, Porto, March 4-6 1991, p.206-219.
[13] Kononenko, I. Estimating attributes: analysis and extensions of RELIEF. In: Proc. ECML-94, Springer, 1994, p. 171-182.
[14] Kononenko, I. On biases in estimating multi-valued attributes. In: Proc. IJCAI-95: Montréal, Canada, August 20-25, 1995. Volume 2, 1995, p. 1034-1040.
[15] Kononenko, I. The minimum description length based decision tree pruning. In Proc. PRICAI '98: Springer, 1998, p. 228-237.
[16] Kononenko, I. Machine learning for medical diagnosis: history, state of the art and perspective. Artif. intell. med., 2001, 23(1) 89-109.
[17] Kononenko, I., Bratko, I., Roškar, E.: Experiments in automatic learning of medical diagnostic rules, Proc. ISSEK workshop, Bled, august 1984, p. 1-16.
[18] Kononenko, I. Štrumbelj, E., Bosnić, Z., Pevec, D., Kukar, M., Robnik Šikonja, M. Explanation and reliability of individual predictions. Informatica (Lj.), 2013, 37(1) 41-48.
[19] Li, M. and Vitanyi, P. An Introduction to Kolmogorov Complexity and its Applications. Springer Verlag, 1993.
[20] Mantaras. R. L. ID3 revisited: A distance based criterion for attribute selection. Methodologies for Intelligent Systems, Charlotte, U.S.A, 1989.
[21] Pompe, U., Kononenko, I. Efficient induction and effective use of first-order knowledge. Appl. artif. intell., 1998, vol. 12, no. 5, p. 421-453.
[22] Quinlan J.R. Discovering rules by induction from large collections of examples. Expert systems in the Micro Electronic Age, Edinburgh University, 1979.
[23] Quinlan J.R. Induction of Decision Trees. Machine Learning, 1986, 1(1) 81-106.
[24] Robnik Šikonja, M., Kononenko, I. An adaptation of RELIEF for attribute estimation in regression. Proc. ICML'97, Nashville, July 8-12, 1997, p.296-304.

AlphaZero – What's Missing?

Ivan Bratko
University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, Ljubljana
E-mail: bratko@fri.uni-lj.si

Keywords: computer game playing, computer chess, machine learning, explainable AI

Received: March 8, 2018

In December 2017, the game playing program AlphaZero was reported to have learned in less than 24 hours to play each of the games of chess, Go and shogi better than any human, and better than any other existing specialised computer program for these games. This was achieved just by self-play, without access to any knowledge of these games other than the rules of the game. In this paper we consider some limitations to this spectacular success. The program was trained in well-defined and relatively small domains (admittedly with enormous combinatorial complexity) compared to many real world problems, and it was possible to generate large amounts of learning data through simulated games which is typically not possible in real life domains. When it comes to understanding the games played by AlphaZero, the program's inability to explain its games and the knowledge acquired in human-understandable terms is a serious limitation.

Povzetek: Decembra 2017 so poročali, da se je program AlphaZero v manj kot 24 urah naučil igrati šah, go in shogi bolje, kot katerikoli človek in katerikoli drug računalniški program specializiran za to igro. To je dosegel kar z igranjem s samim seboj, brez dostopa do kakršnegakoli znanja o teh igrah, razen samih pravil igre. Vsiljuje se vprašanje, ali obstajajo kakšne omejitve tega neverjetnega podviga. Program se je učil v dobro definiranih in razmeroma enostavnih domenah (čeprav je res, da imajo te igre ogromno kombinatorično zahtevnost) v primerjavi z mnogimi problemi realnega sveta. Za te igre je bilo mogoče s simulacijo generirati ogromne količine učnih podatkov, kar navadno ni možno v domenah iz realnega življenja. Osnovna pomanjkljivost programa AlphaZero je tudi njegova nezmožnost, da bi svoje odigrane partije razložil na človeku razumljiv način.

1 Introduction

In December 2017, an amazing achievement has been reported (Silver, Hubert et al. 2017). DeepMind's program AlphaZero was able to learn in less than 24 hours to play each of the games of chess, Go and shogi better than any human, and better than any other existing specialised computer program for these games.

This was a third event in the success story at DeepMind with game playing programs with the word Alpha in their names. It started with the famous program AlphaGo (Silver et al. 2016) which convincingly defeated one of the best human go players in a match of five games. That was the first time ever that a computer program was able to defeat a leading human player at Go. AlphaGo was specialised at Go, and learned from exemplary high quality games of Go previously played by strong human players. AlphaGo Zero (Silver, Schrittwieser et al. 2017) was able to learn to play Go even better. The impressive difference between AlphaGo and AlphaGo Zero was that the latter can learn from games just played by itself, thus without having access to examples of well-played games or any other source of game-specific knowledge of the game, except the bare rules of the game.

Finally, AlphaZero is a general game playing program not specialised to Go, so it can learn to play any game of this kind just by self-play. For example, to get to the strength level of the best human chess players, AlphaZero needed no more than one and a half hours of learning by self-play.

The basic architecture of AlphaZero is as follows. AlphaZero learns by reinforcement learning from simulated games against itself. It uses a deep neural network that learns to estimate the values of positions and the probabilities of playing possible moves in a position. To select a move to play in the current board position, AlphaZero performs Monte Carlo Tree Search (MCTS). This search consists of simulating random games from the current positions, in which the probabilities of random moves increase with the move probabilities returned by the neural network, and decrease with the moves' visit counts. The use of MCTS in chess is in contrast to search in other strong chess programs. They perform Alpha-Beta search which had been considered before AlphaZero much more appropriate for chess.

2 An interesting observation about AlphaZero training in chess

To appreciate this achievement, let us consider some illustrative quantitative facts about AlphaZero at chess. As reported by Silver, Hubert et al. (2017), in chess training AlphaZero played about 44 million games against itself in nine hours of self-play. This took 700

thousand "steps" of training. According to the plots of chess rating improvement in time of training (Silver, Hubert et al., 2017), AlphaZero attained the chess strength of best human players after about 110 thousand training steps. By that time, AlphaZero had played about 6.9 million games with itself.

Now let us consider some quantitative facts from the human history of chess. ChessBases's Mega Database is a comprehensive collection of chess games played in all history of human chess. Mega Database is very representative of about all important chess games ever played by humans, so it is well representative of all chess concepts and ideas ever found by human players. The 2018 version of Mega Database contains 7.1 million games which quite amazingly matches AlphaZero's estimated 6.9 million games needed to reach the best humans' chess strength. Of course it may be argued that this is a mere coincidence. And it can be rightfully observed that this comparison is rather crude: it is not true that the best human players derive their skill from *all* 7.1 million games. It is certainly not true that all the games in Mega Database are needed to subsume the present chess knowledge by mankind. Therefore Mega Database, viewed as a kind of codification of total human chess knowledge, contains a lot of redundancy. Nevertheless, the numbers do seem to offer a first "feasibility check" of AlphaZero's achievement.

3 Are there any limitations to AlphaZero approach?

Given that the games of chess, Go and shogi are so difficult for humans, and that AlphaZero made the same progress at chess, say, in 1.5 hours of self-play as the mankind did in over a hundred years, this looks impressive indeed. If the problem of such difficulty for humans can be mastered in one and a half hours by a machine using AI techniques, then an impression is that AI can now do everything.

But let us consider whether this impression is really so true in general. What are the limitations? Let us look at the problems dealt with by AlphaZero from a little broader perspective.

(1) These games are limited to the board worlds, which amounts to 64 squares for chess, and 381 squares for Go. True, these small worlds give rise to combinatorial complexities of astronomic proportions. For chess, an old estimate by Claude Shannon (1950) is: there are over 10^{40} possible chess positions, and over 10^{120} possible games. The magnitude of these numbers is popularly illustrated by their comparison to the number of all atoms in the observable universe, which is of the order 10^{80}. The number of possible games of chess is thus incomparably larger than the size of the universe. And, also true, both Go and shogi are in these terms much more complex than chess. On the other hand, the combinatorial complexity of these games is rather deceiving. Compared to many real world domains studied by biology, chemistry and physics, these games are small.

(2) The rules of these games are simple and known. Therefore almost unlimited experimentation with these games through simulated games is possible. This gives rise to the automatic generation of very large numbers of training instances from which AlphaZero could learn.

This is very different from many complex real-world domains in which learning data is collected through time consuming and expensive experiments, and therefore the amount of training data is much more limited. In contrast to machine learning from big data, in such domains the *scarcity* of data is often the problem. For example, Wiley et al. (2016) describe reinforcement learning by a tracked robot for which no sufficiently accurate simulation model was available. Therefore, experimentation had to be carried out with the actual physical robot, so the number of trials was severely limited due to time constraints and wear and tear of the robot. More elaborate methods of machine learning were needed to enable more effective use of available data. The situation with available data may be even more constrained, like in medicine where examples of patients with a disease under study can only be "generated by nature". For machine learning to be successful with "small data", different machine learning methods and algorithms are needed. In particular, it is desired that the learning method can make use of domain background knowledge. In this way lack of data can be compensated by prior knowledge. For example, the learning program may use the laws of physics that are already known prior to learning.

4 Does AlphaZero play chess "more like humans"?

There have been some speculations that AlphaZero is not only by far the strongest chess playing program, but that it also plays chess in a way that is more similar to the way strong human players play chess.

This conjecture is based on a particular comparison between AlphaZero and Stockfish, one of the strongest chess programs before AlphaZero. AlphaZero convincingly defeated Stockfish in a match of 100 games in which AlphaZero won 28 games and drew the rest. The particular point of comparison is the number of positions searched per second by the two programs. Stockfish searched 70 million of positions per second compared with 80,000 by AlphaZero. This was interpreted as indicative of a more human-like style by AlphaZero simply because in general computers base their strength on the brute force computational power which allows them to search deeper. By contrast, humans can only search typically of the order of a few tens of positions per move, or something like a few positions per second.

Therefore the humans, to compensate for this inferior search ability, have to rely on deeper chess knowledge and intuition. The argument then is that AlphaZero with about thousand times lower search speed than Stockfish must have better chess knowledge to still be able to win. This argument is not completely convincing. In terms of search speed, AlphaZero is still incomparably faster than humans. Another big difference between AlphaZero and

human style of play is in the search method used. AlphaZero uses Monte Carlo tree search technique which is based on random simulations of possible games from the current position, and counting favourable outcomes resulting from moves tried. This is certainly not the way that humans perform their search. On the other hand, moves played as the result of MCTS indeed seem to resemble human players' decisions more than moves played by typical chess engines. In particular, it appears that moves played by AlphaZero better reflect long-term positional judgement in chess that is attributed to strong human players' deep understanding of the game. We will return to this question in the next section when analysing a surprising positional sacrifice by AlphaZero in one of the games against Stockfish.

5 Examples of super play by AlphaZero

The world of chess was stunned by examples of play by AlphaZero from some of the published games between AlphaZero and Stockfish. Probably the most spectacular example comes from the following game in which AphaZero had White pieces. This example was discussed many times in numerous chess media, for example in (Guid 2018). After 18 moves, the position in Fig. 1 occurred in the game, with White to move. Here Black is threatening to capture White knight on h6 with the queen or the king. So it seems that White knight has to retreat to g4, which a reasonable human player would actually do. After that, if both sides played their best moves, White knight would eventually escape to safety, but Black would come out with a somewhat better position. However, in position in Fig. 1, AlphaZero played incredibly **19 Rf1-e1**, leaving his knight on h6 to be captured by Black. In the game, Stockfish indeed took the knight and appeared to be winning. AlphaZero did have some positional compensation for the knight, but that did not appear to be anything nearly enough for the material disadvantage. But AlphaZero's judgement turned out to be better in the long run. White managed to create threats virtually out of nothing, and 20 moves later managed to achieve a clear advantage. To appreciate the details of all this requires some chess knowledge, so further chess comments are given in the Appendix.

It is very hard to clearly explain that **19 Rf1-e1** was really a good move, and how AlphaZero was able to find this decision. It seems that the combination of AlphaZero's Monte Carlo Tree Search and AlphaZero's move evaluation stored in its neural network somehow resulted in such a deep positional judgement.

One possible explanation for this might be as follows. Positional evaluation in chess takes into account static features in the current position. Such features tend not to change quickly during play, so they have long-lasting effects. An example of such a positional feature is weak pawns that cannot move and are hard to defend, and can thus become targets for enemy pieces in the course of the game. Another example are chain formations of blocked pawns that create more space for one of the sides. More space gives to one side better

Figure 1: Position after Black's move **18 ... g6-g5**. AlphaZero here played the surprising **19 Rf1-e1**, leaving White knight on h6 undefended.

chances to manoeuver their pieces and thus create chances for attack in the long run on the part of the board with space advantage. However, it usually takes many moves before such positional advantages can be exploited and turned into something more tangible like material advantage. It may also happen that positional advantage cannot be exploited at all. In such cases, the positional advantage simply evaporates in the long run. It is very difficult for humans to estimate whether positional advantage can eventually be converted or not because it is hard to see so far into the future of the game. It is often far enough that this may also be a problem for a typical chess engine that uses Alpha-Beta search. Here it is that Monte Carlo Tree Search might be much more appropriate because it is more selective and can therefore go much deeper than Alpha-Beta. Of course, for Monte Carlo search to be successful, it has to be well guided by the move probability estimates, which seems to be a major strength in AlphaZero. In position of Figure 1, the positional advantage of AlphaZero's knight sacrifice was only converted into material gains after twenty moves. This is too deep for Alpha-Beta search, but possible to see by MCTS. Now although it seems that random trials of MCTS are quite absurd to be carried out by a human player, it can be imagined that something roughly similar is actually done by strong human players. When a good player tries to estimate how concrete the consequences of a positional advantage may become, he or she tries to calculate very deeply and selectively sample variations. Favourable results from these variations will increase the player's confidence into the correctness of a positional sacrifice.

6 Can humans understand and learn from Alpha Zero?

The chess moves played by AlphaZero in the example above call for an explanation. Ideally, AlphaZero would

be able of comment on its games and explain its decisions in human-understandable terms. So humans would be able to learn from AlphaZero new chess concepts and ideas, enhance their own chess knowledge and be able to use it in their own play.

In this respect, the lack of explanation facility is a serious limitation of AlphaZero paradigm, and many forms of machine learning in general. Many of the present successful ML methods that can outperform humans have this same limitation. It is hard for humans, and human experts, to understand what has actually been learned by the program. Ideally, learning programs should be able to explain what they discovered through learning, so that this new knowledge could also be used by humans.

This idea has been around in the area of machine learning almost from its beginning, at that time also known under the term "machine synthesis of expert knowledge". This phrase was coined by Donald Michie in early 1980's, some time before the idea became generally accepted. Donald also set up an international association called ISSEK (International School for the Synthesis of Expert Knowledge). The main activity of ISSEK was a series of workshops in 1980's and 1990's to enable a collaboration among research laboratories interested in developing machine learning techniques for the synthesis of new knowledge from data. As an attempt at precisely defining these aspects of machine learning, Michie (1988) defined three criteria for machine learning, and it will be useful to repeat them here. Essentially, these criteria were:

(1) Weak criterion: the learning system improves its performance through learning from experience

(2) Strong criterion: as (1), plus the system can output what it has learned in explicit symbolic form

(3) Ultra-strong criterion: as (2), plus the explicit symbolic description produced can be used by a human operationally, that is to improve the human's own performance at solving the task

By far the strongest attention in machine learning has been devoted to criterion (1), and the imbalance of attention between the three criteria has probably been increasing over time. Importance of criteria (2) and (3) with relevant examples was discussed for example in (Bratko 1997).

There has been recent renewed interest in relation to the latter two criteria within Explainable AI (XAI 2017; Miller 2017). A related issue is the question of comprehensibility of a description by humans. For a human to be able to use operationally what was learned, the human at least has to understand the result of learning. Therefore, for the ultra-strong criterion to be applicable in practice as a measure of success, a measure of comprehensibility by a human of a (machine-generated) description is required. Although such a need has been often observed in machine learning, little progress seems to have been made in this respect. (Muggleton et al. 2018) is a rare recent attempt at defining an operational measure of comprehensibility.

In terms of Donald Michie's criteria, AlphaZero has been a tremendous success in terms of the weak criterion for machine learning, but no attention seems to have been paid to the other two criteria in the development of AlphaZero. As a result, AlphaZero has miraculously acquired a lot of new game-specific knowledge, but at the moment it is hidden from humans in a black box. As described by Voosen (2017), a human interested in that knowledge can play a time consuming game of an AI detective to uncover small bits of that knowledge in the box. Fundamental progress in terms of Michie's ultra-strong criterion with AlphaZero, and other similarly influential systems that will appear in the future, will be needed to increase their impact in the important direction of improving human knowledge.

7 Conclusion

In this paper, we considered some limitations of AI techniques on which AlphaZero is based. These limitations are indicative of some directions for future research in AI. Many games played by AlphaZero are very interesting, and it seems that, at least in chess, AlphaZero has discovered new concepts that human players are not aware of. At the moment, humans can only make guesses about what these new concepts might be. Therefore, the development of explanation techniques, aiming at human-friendly conceptualisation of the automatically acquired game-playing knowledge would be very well motivated. Also, improving machine learning methods towards more data-efficient learning would be important for applicability in many real-world domains.

8 Acknowledgements

I would like to thank Matej Guid, Martin Moina and Marjan Šemrl, a former correspondence chess world champion, for discussion.

9 References

[1] I. Bratko, Machine learning: between accuracy and interpretability. In: *Machine Learning, Networks and Statistics*. Eds. G. Della Riccia, H.-J. Lenz, R. Kruse. Springer, 1997. pp. 163-178.

[2] M. Guid, AlphaZero. *Šahovska misel* (*Chess Thought Magazine*), Februar 2018 (in Slovene).

[3] D. Michie, Machine learning in the next five years. *Proc. Third European Working Session on Learning*, pages 107–122. Pitman, 1988.

[4] T. Miller, Explanation in AI: Insights from the social sciences. 2017. arXiv.org > cs > arXiv:1706.07269

[5] S.H. Muggleton, U. Schmid, C. Zeller, A. Tamaddoni-Nezhad, and T. Besold. Ultra-strong machine learning - comprehensibility of programs learned with ILP. *Machine Learning*, 2018. In Press.

[6] Claude Shannon (1950). Programming a computer for playing chess. *Philosophical Magazine*. **41** (314)

[7] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I.

Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[8] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of Go without human knowledge. *Nature*, 550:354–359, 2017.

[9] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, D. Hassabis, Mastering chess and chogi by celf-clay with a ceneral Reinforcement Learning algorithm. 2017. arXiv.org > cs > arXiv:1712.01815

[10] T. Wiley, C. Sammut, B. Hengst, I. Bratko, A multi-strategy architecture for on-line learning of robotic behaviours using qualitative reasoning. *Advances in Cognitive Systems Journal*, **4** (2016), pp. 93-111.

[11] P. Voosen, How AI detectives are cracking open the black box of deep learning. Science, July 2017.

[12] XAI 2017 (Proc. IJCAI-17 Workshop on Explainable AI), 2017. http://www.intelligentrobots.org/files/IJCAI2017/IJCAI-17_XAI_WS_Proceedings.pdf

10 Appendix: Detailed analysis of the game AlphaZero vs. Stockfish from position of Fig. 1

In position of Fig. 1, White knight is in trouble and it seems that he has to retreat from h6 to g4. This is the only safe square for the knight. The knight is now under attack of Black bishop on c8, but the knight is defended by White queen. However, White's problem is not completely over because Black can try to chase White queen away from defending the knight on g4. Thus the following continuation is logical: **19 Nh6-g4 b6-b5** (attacking White queen), **20 Qa4-e4** (the only square from which the queen can still defend the knight, but now Black has double attack on White queen and knight with the next pawn move) **f7-f5**. Fortunately for White, White can check Black king and the following variation is more or less forced: **21 Qe4-e5+ Kg7-f7 22 Qe5xd6 Be7xd6 23 Rf1-d1 Bd6-c7 24 Ng4-e3**. Now White knight has survived the trouble, but Black is a pawn up and the position is somewhat better for Black. This variation is also given as the best possibility for White by typical chess programs, and it is what every reasonable human player would do, accepting a worse position as the least possible damage. AlphaZero however very surprisingly played **19 Rf1-e1**, leaving the unfortunate knight on h6 under threat. The knight can now be immediately captured by Black king: **19 ... Kg7xh6** which Stockfish actually did in the game. A typical chess program now evaluates the position as considerably better for Black. Black is a whole piece up. True, White can play **20 h2-h4** and Black king will be feeling a little uncomfortable, so White does have some compensation for the sacrificed piece. But is this compensation sufficient? The answer appears to be a clear "no" to practically any human player, as well as any chess program other than AlphaZero. Black has big material advantage, and White seems to have no tangible compensation in return. It is too complex to calculate all the possible continuations to sufficient depth in this position because there are no forced variations clearly favourable to White or Black. So in practice this position can only be evaluated through a kind of "intuitive positional judgement" (in quotes when it refers to a computer). In this case, AlphaZero was in fact capable of such deep positional judgement, something that is extremely difficult for humans, and so far has been considered even harder for machines. In the game, after **19 Kg7xh6**, the following moves were played: **20 h2-h4 f7-f6 21 Bc1-e3 Bc8-f5 22 Ra1-d1 Qd6-a3 23 Qa4-c4 b6-b5 24 h4xg5+ f7xg5 25 Qc4-h4+ Kh6-g6 26 Qh4-h1**. The position at this point is shown in Fig. 2.

White queen now looks very passive in the corner, and thus White, still with a piece down, seems considerably worse. But the prospects of White queen on h1 are actually excellent. The idea is that the queen at h1 supports the move by White bishop **g2–e4**, and after the exchange of the light coloured bishops, White queen will

Figure 2: Position after 26 Qh4-h1.

threaten to enter the center via light squares with great force. This actually happened in the game and 15 moves later White achieved a clear advantage. So the controversial move **19 Rf1-e1** by AlphaZero in position of Fig. 1 turned out to be a brilliant positional sacrifice much admired by the chess world.

Explanation of Prediction Models with ExplainPrediction

Marko Robnik-Šikonja
University of Ljubljana, Faculty of Computer and Information Science,
Večna pot 113, 1000 Ljubljana, Slovenia
Email: marko.robnik@fri.uni-lj.si, https://fri.uni-lj.si/en/employees/marko-robnik-sikonja

Keywords: machine learning, comprehensibility of models, explanation of models, perturbation methods

Received: October 31, 2017

State-of-the-art prediction models are getting increasingly complex and incomprehensible for humans. This is problematic for many application areas, especially those where knowledge discovery is just as important as predictive performance, for example medicine or business consulting. As machine learning and artificial intelligence are playing an increasingly large role in the society through data based decision making, this is problematic also from broader perspective and worries general public as well as legislators. As a possible solution, several explanation methods have been recently proposed, which can explain predictions of otherwise opaque models. These methods can be divided into two main approaches: gradient based approaches limited to neural networks, and more general perturbation based approaches, which can be used with arbitrary prediction models. We present an overview of perturbation based approaches, and focus on a recently introduced implementation of two successful methods developed in Slovenia, EXPLAIN and IME. We first describe their working principles and visualizations of explanations, followed by the implementation in ExplainPrediction package for R environment.

Povzetek: Najboljši napovedni modeli postajajo vse bolj zapleteni in nerazumljivi za ljudi. To je problematično za številna aplikativna področja, zlasti tista, kjer je odkrivanje znanja enako pomembno kot napovedna točnost, npr. medicina ali poslovno svetovanje. Ker strojno učenje in umetna inteligenca preko na podatkih temelječega odločanja igrata vse večjo vlogo v družbi, je to problematično tudi s širšega vidika in vse bolj skrbi javnost in zakonodajalce. Kot možna rešitev se je v zadnjem času pojavilo več metod razlage za napovedne modele. Te metode lahko razdelimo na dve skupini: na gradientne metode, omejene predvsem na umetne nevronske mreže, in splošnejše pristope na osnovi perturbacij vhodov, ki jih je mogoče uporabiti pri poljubnih napovednih modelih. Predstavljamo pregled perturbacijskih pristopov in dve uspešni metodi razviti v Sloveniji, EXPLAIN in IME. Najprej opišemo njuno delovanje in vizualizacije razlag, nato pa še implementacijo v paketu ExplainPrediction za okolje R.

1 Introduction

Machine learning methods and especially prediction models are becoming an essential ingredient of many modern products and services. Through a paradigm of data-based decisions, they impact mundane everyday tasks (e.g., shopping and entertainment recommendations), as well as life-changing decisions (e.g., medical diagnostics, credit scoring, or security systems). As societies are getting more and more complex, we can expect that their reliance on automatic decisions will increase. It is natural that those affected by various decisions of prediction models want to get feedback and understand the reasoning process and biases of the underlying models. The impact and influence of automatic decisions are getting so ubiquitous that the whole area of artificial intelligence is receiving an increasing attention from lawmakers who demand that decisions of important models are made transparent. Besides public services, the areas where models' transparency is of crucial importance are for example medicine, science, policy making, strategic planning, business intelligence, finance, marketing, law, and insurance. In these areas, users of models are just as interested in comprehending the decision process, as in the classification accuracy of prediction models. Unfortunately, most of the top performing machine learning models are black boxes in a sense that they do not offer an intrinsic introspection into their decision processes or provide explanations of their predictions and biases. This is true for Artificial Neural Networks (ANN), Support Vector Machines (SVM), and all ensemble methods (for example, boosting, random forests, bagging, stacking, and multivariate adaptive regression splines). Approaches that do offer an intrinsic introspection such as decision trees or decision rules do not perform so well or are not applicable in many cases (17).

To alleviate this problem two types of model explanation techniques have been proposed. The first type, which is not discussed in this work, is based on the internal working of the particular learning algorithm. The explanation methods exploit model's representation or learning process to gain insight into the presumptions, biases, and reasoning leading to final decisions. Two well-known models where

such approach works well are neural networks and random forests. Recent explanations for neural networks classifiers of images mostly rely on layer-wise relevance propagation (3) or gradients of output neurons with respect to the input (28) to visualize parts of images significant for particular prediction. The random forest visualizations mostly exploit the fact that during bootstrap sampling, which is part of this learning algorithm, some of the instances are not selected for learning and can serve as an internal validation set. With the help of this set important features can be identified and similarity between objects can be measured.

The second type of explanation approaches are general and can be applied to any predictive model. The explanations provided by these approaches try to efficiently capture the causal relationship between inputs and outputs of the given model. To this end, they perturb the inputs in the neighborhood of given instance to observe effects of perturbations on model's output. Changes in the outputs are attributed to perturbed inputs and used to estimate their importance for a particular instance. Examples of this approach are methods EXPLAIN (24), IME (31), and LIME (21). These methods can explain models' decision process for each individual predicted instance as well as for the model as a whole. We implemented the methods, proposed by our group, EXPLAIN and IME, in R package ExplainPrediction (23).

The objectives of the paper are twofold. First, we explain how general perturbation-based explanation methods work and second, we describe the implementation details, parameters, and visualization of the ExplainPrediction package which implements them. The first aim is achieved through an explanation of their working principle and graphical explanation of models' decisions on a well-known data set. The second aim is no less important. In machine learning, open source implementations enable progress, empirical comparisons, and replicability of research. Two types of explanations are implemented in ExplainPrediction and demonstrated in the paper, individual predictions of new unlabeled cases and functioning of the model as a whole. This allows inspection, comparison, and visualization of otherwise opaque models.

The structure of the remainder of the paper is as follows. In Section 2, we present the background and related work on perturbation based explanation approaches. In Section 3, we present methods EXPLAIN and IME. Their implementation, parameters, and use with the ExplainPrediction package are covered in Section 4. In Section 5, we present conclusions and promising research directions.

2 Background and overview of perturbation approaches

We first present different modes of explanation and properties of model explanation approaches, followed by an overview of explanation approaches.

True causal relationships between dependent and independent variables are typically hidden except in artificial domains where all the relations, as well as the probability distributions, are known in advance. Therefore only explanations of prediction process for a particular model is of practical importance. The prediction accuracy and the correctness of explanation for a given model may be orthogonal: the correctness of the explanation is independent of the correctness of the prediction. However, empirical observations show that better models (with higher prediction accuracy) enable better explanations (31). We discuss two types of explanations:

- **Instance explanation** explains predictions with the given model of a single instance and provides the impact of input feature values on the prediction.
- **Model explanation** is usually an aggregation of instance explanations over many (training) instances, to provide top-level explanations of features and their values. This aggregation over many instances enables identification of different roles attributes may play in the classifications of instances.

In a typical data science problem setting, users are concerned with both prediction accuracy and the interpretability of the prediction model. Complex models have potentially higher accuracy but are more difficult to interpret. This can be alleviated either by sacrificing some prediction accuracy for a more transparent model or by using an explanation method that improves the interpretability of the model. Explaining predictions is straightforward for symbolic models such as decision trees, decision rules, and inductive logic programming, where the models give an overall transparent knowledge in a symbolic form. Therefore, to obtain the explanations of predictions, one simply has to read the rules in the corresponding model. Whether such an explanation is comprehensive in the case of large trees and rule sets is questionable. Piltaver et al. (18) developed criteria for comprehensibility of decision trees and performed a user study, which showed that the depth of the deepest leaf that is required when answering a question about a classification tree is the most important factor influencing the comprehensibility.

For non-symbolic models, there are no intrinsic explanations. A lot of efforts have been invested in increasing the interpretability of complex models. For SVM, Hamel (12) proposed an approach based on self-organizing maps that groups instances then projects the groups onto a two-dimensional plane. In this plane, the topology of the groups is hopefully preserved and support vectors can be visualized. Many approaches exploit the essential property of additive classifiers to provide more comprehensible explanations and visualizations, e.g., (14) and (19).

Visualization of decision boundaries is an important aspect of model transparency. Barbosa et al. (6) present a technique to visualize how the kernel embeds data into a high-dimensional feature space. With their Kelp method, they visualize how kernel choice affects neighborhood

structure and SVM decision boundaries. Schulz et al. (27) propose a general framework for visualization of classifiers via dimensionality reduction. Goldstein et al. (11) propose another useful visualization tool for classifiers that can produce individual conditional expectation plots, graphing the functional relationship between the predicted response and the feature for individual instance.

Some explanations methods (including the ones presented in Section 3) are general in a sense that they can be used with any type of classification model (15; 21; 24; 30). This enables their application with almost any prediction model and allows users to analyze and compare outputs of different analytical techniques. Lemaire et al. (15) applied their method to a customer relationship management system in the telecommunications industry. The method which successfully deals with high-dimensional text data is presented in (16). Its idea is based on general explanation methods EXPLAIN and IME and offers an explanation in the form of a set of words which would change the predicted class of a given document. Bosnić et al. (9) adapt the general explanation methodology to data stream scenario and show the evolution of attribute contributions through time. This is used to explain the concept drift in their incremental model. In a real-life breast cancer recurrence prediction, Štrumbelj et al. (29) illustrate the usefulness of the visualizations and the advantage of using the general explanation method. Several machine learning algorithms were evaluated. Predictions were enhanced with instance explanations using the IME method. Visual inspection and evaluation showed that oncologists found the explanations useful and agreed with the computed contributions of features. Pregeljc et al. (20) used traditional modeling approaches together with data mining to gain insight into the connections between the quality of organization in enterprises and the enterprises' performance. The best performing models were complex and difficult to interpret, especially for non-technical users. Methods EXPLAIN and IME explained the influence of input features on the predicted economic results and provided insights with a meaningful economic interpretation. The interesting economic relationships and successful predictions come mostly from complex models such as random forests and ANN. Without proper explanation and visualization, these models are often neglected in favor of weaker, but more transparent models. Experts from the economic-organizational field, which reviewed and interpreted the results of the study, agreed that such an explanation and visualization is useful and facilitates comparative analysis across different types of prediction models. Bohanec et al. (7) present an innovative application of explanation methods EXPLAIN and IME in the context of B2B sales forecasting. They demonstrate how users can validate their assumptions with the presented explanations and test their hypotheses using the explanations for a sort of what-if analysis. Bohanec et al. (8) address the problem of weak acceptance of machine learning models in business environments. The propose a framework of top-performing machine learning models coupled with general explanation methods to provide an additional information to the decision-making process. This is shown to reduce error, efficiently support business decision makers and builds a foundation for sustainable organizational learning. Demšar and Bosnić (10) use the general explanation methods EXPLAIN and IME to detect concept drift in data streams. Due to the generality of explanations, their drift detector can be combined with an arbitrary classification algorithm and features good drift detection, accuracy, robustness, and sensitivity.

Many explanation methods are related to statistical sensitivity analysis and uncertainty analysis (26). In that methodology sensitivity of models is analyzed with respect to models' input. A related approach, called inverse classification (1) tries to determine the minimum required change to a data point in order to reclassify it as a member of a different class. An SVM model-based approach is proposed by Barbella et al. (5). Another sensitivity analysis-based approach explains contributions of individual features to a particular classification by observing (partial) derivatives of the classifiers prediction function at the point of interest (4). A limitation of this approach is that the classification function has to be first-order differentiable. For classifiers not satisfying this criterion (for example, decision trees) the original classifier is first fitted with a Parzen window-based classifier that mimics the original one and then the explanation method is applied to this fitted classifier. The method is practically useful with kernel-based classification method to predict molecular features (13).

Due to recent successes of deep neural networks in image recognition and natural language processing, several explanation methods specific to these two application areas emerged, recently. Methods working on images try to visualize parts of images (i.e., groups of pixels) significant for a particular prediction. These methods mostly rely on the propagation of relevance within the network. For example, layer-wise relevance propagation (3), and computation of gradients of output neurons with respect to the input (28). In language processing Arras et al. (2) applied layer-wise relevance propagation to a convolutional neural network and a bag-of-words SVM classifier trained on a topic categorization task. The explanations indicate how much individual words contribute to the overall classification decision.

3 Methods EXPLAIN and IME

General explanation methods can be applied to any classification model which makes them a useful tool both for interpreting models (and their predictions) and comparing different types of models. By modification of feature values of interest, what-if analysis is also supported. Such methods cannot exploit any model-specific properties (e.g., gradients in ANN) and are limited to perturbing the inputs of the model and observing changes in the model's output (15; 24; 30).

The presented general explanation methods provide two types of explanations for prediction models: instance explanations and model explanations (see Section 2). Model explanations work by summarizing a representative sample of instance explanations. The presented methods estimate the impact of particular explanation feature for a given instance by perturbing similar instances.

The key idea of EXPLAIN and IME is that the contribution of a particular input value (or set of values) can be captured by "hiding" the input value (set of values) and observing how the output of the model changes. As such, the key component of general explanation methods is the expected conditional prediction – the prediction where only a subset of the input variables is known. Let Q be a subset of the set of input variables $Q \subseteq S = \{X_1, \ldots, X_a\}$. Let $p_Q(y_k|x)$ be the expected prediction for x, conditional to knowing only the input variables represented in Q:

$$p_Q(y_k|x) = \mathbb{E}(p(y_k)|X_i = x_{(i)}, \forall X_i \in Q). \quad (1)$$

Therefore, $p_S(y_k|x) = p(y_k|x)$. In practical settings, the classification function of the model is not known - one can only access its prediction for any vector of input values. Therefore, exact computation of this equation is not possible and sampling-based approximations are used.

To produce model explanations we sum instance level explanations. The evidence for and against each class is collected and visualized separately. In this way, one can, for example, see that a particular value of an attribute supports specific class but not in every context.

3.1 EXPLAIN, one-variable-at-a-time approach

EXPLAIN method computes the influence of a feature value by observing its impact on the model's output. The EXPLAIN assumes that the larger the changes in the output, the more important role the feature value plays in the model. The shortcoming of this approach is that it takes into account only a single feature at a time, therefore it cannot detect certain higher order dependencies (in particular disjunctions) and redundancies in the model. The EXPLAIN assumes that the characterization of the i-th input variable's importance for the prediction of the instance x is the difference between the model's prediction for that instance and the model's prediction if the value of the i-th variable was not known. The source of explanations is therefore:

$$p(y_k|x) - p_{S\setminus\{i\}}(y_k|x). \quad (2)$$

If this difference is large then the i-th variable is important. If it is small then the variable is less important. The sign of the difference reveals whether the value contributes towards or against class value y_k. This approach was extended in (24) to use log-odds ratios (or weight of evidence), or information gain instead of the difference in predicted class probabilities.

The lack of information about A_i in $p_{S\setminus\{i\}}(y_k|x)$ is approximated with several predictions. For nominal attributes, we replace the actual value of A_i in each prediction with each of the possible values of attribute A_i, and weight each prediction with the prior probability of the value:

$$p(y_k|x \setminus A_i) \doteq \sum_{s=1}^{m_i} p(A_i = a_s) p(y_k|x \leftarrow A_i = a_s) \quad (3)$$

Here m_i is the number of nominal values of attribute A_i and the term $p(y_k|x \leftarrow A_i = a_s)$ represents the predicted probability of y_k when in instance x we replace the actual value of A_i with a_s. For numerical attributes, we use discretization to split the values of numerical attribute A_i into intervals. The middle points of these intervals are taken as the representative replacement values in Eq. (3), for which we compute predictions $p(y_k|x \leftarrow A_i = a_s)$. Instead of prior probabilities of individual values $p(A_i = a_s)$, we use probabilities of the intervals for weighting the predictions.

To demonstrate the behavior of the method an example of an explanation is given. Let a binary domain contain three important (A_1, A_2, and A_3) and one irrelevant attribute (A_4), so the set of attributes is $S = \{1, 2, 3, 4\}$. The class variable C is expressed as the parity (xor) relation of three attributes $C = A_1 \oplus A_2 \oplus A_3$.

Let us assume that we trained a perfect model for this problem. Our correct model classifies an instance $x = (A_1 = 1, A_2 = 0, A_3 = 1, A_4 = 1)$ to class $C = 0$, and assign it the probability $p(C = 0|x) = 1$. When explaining classification for this particular instance $p(C = 0|x)$, method EXPLAIN simulates the lack of knowledge of a single attribute at a time, so it has to estimate $p_{S\setminus\{1\}}(C = 0|x)$, $p_{S\setminus\{2\}}(C = 0|x)$, $p_{S\setminus\{3\}}(C = 0|x)$, and $p_{S\setminus\{4\}}(C = 0|x)$. Without the knowledge about the values of each of the attributes A_1, A_2, and A_3, the model cannot correctly determine the class value, so the correct estimates of class probabilities are $p_{S\setminus\{1\}}(C = 0|x) = p_{S\setminus\{2\}}(C = 0|x) = p_{S\setminus\{3\}}(C = 0|x) = 0.5$ The differences of probabilities $p_S(y_k|x) - p_{S\setminus\{i\}}(y_k|x)$ therefore equal 0.5 for each of the three important attributes, which indicate that these attributes have positive impact on classification to class 0 for the particular instance x. The irrelevant attribute A_4 does not influence the classification, so the classification probability remain unchanged $p_{S\setminus\{4\}}(C = 0|x) = 1$. The difference of probabilities $p_S(C = 0|x) - p_{S\setminus\{4\}}(C = 0|x) = 0$ so the explanation of the irrelevant attribute's impact is zero.

In reality, the trained models are rarely perfect, so the obtained probabilities used in Eq. (2) contain a certain amount of error which translates to an error of explanations. The empirical evaluation (24) has shown that better models produce better explanations.

3.2 IME, all-subsets approach

The one-variable-at-a-time approach is simple and computationally less intensive but has some disadvantages. The main disadvantage is that disjunctive concepts or redundancies between input variables may result in unintuitive con-

tributions for variables (31). A solution was proposed in (30), where all subsets of values are observed. Such procedure demands 2^a steps, where a is the number of attributes and results in the exponential time complexity. However, the contribution of the variable corresponds to the Shapley value for the coalitional game of a players. This allows an efficient approximation based on sampling.

As sampling takes values for attribute A_i from the existing set of values, we don't need any approximation similar to Eq. (3) for numerical attributes in IME.

3.3 Presenting explanations

The working and practical utility of the one-variable-at-a-time contributions and their visualization are illustrated on the well-known Titanic data set. The task is to classify the survival of a passenger in the disaster of the Titanic ship. The three input variables report the passengers' status during travel (first, second, third class, or crew), age (adult or child), and gender. Note the similarity of the problem with many business decision problems, such as churn prediction, mail response, insurance fraud, etc. As an example, random forest (rf) classifier is used. This model is robust and usually provides good prediction accuracy but it is incomprehensible. The Fig. 1a shows an example of an instance explanation for the prediction of the instance with id 2 (a first class adult male passenger). The text at the top includes the class value in question, the instance id, and the type of model. At the bottom, the description contains the type of explanation technique used, the model's prediction for the selected class value, and the actual correct class value for the instance. The input variables' names are shown on the left-hand side and their values for the particular instance are on the right-hand side. The thick dark shaded bars indicate the contributions of the instance's values for each corresponding input variable towards (or against) the class value "survived=yes". The thinner and lighter bars above indicate average contributions of these values across all instances. For the given instance one can observe that "sex=male" speaks against survival and "status=first class" speaks in favor of survival while being an adult has little influence. Thinner average bars above them reveal that being male can be both favorable and dangerous while being in the first class is on average even more beneficial than in the selected case. Note that the same visualization can be used even if some other classification method is applied. A more general view of the model is provided by averaging the explanations for the training data and their visualization in a summary form, which shows the average importance of each input variable and its values. For numerical attributes, explanations for intervals of values are shown; to get sensible intervals, numerical attributes are discretized. An example of such a model explanation for Titanic data set is presented in Fig. 1b. On the left-hand side, the input variables and their values are shown. For each value, the average negative and the average positive contributions across all instances is displayed. Note that negative and positive contributions would cancel each other out if summed together, so it is important to keep them separate. The lighter bars shown are equivalent to the lighter bars in the instance explanation on Fig. 1a. For each input variable, the average positive and negative contributions for all values and instances are shown (darker bars). The visualization reveals that the sex has the strongest effect in random forest model. Traveling in the first or second class has a predominantly positive contribution towards the survival, being a child or female has greater positive than negative contribution, while traveling in the third class has a negative contribution.

4 Implementation in R package ExplainPrediction

The methods EXPLAIN and IME are implemented in the R package ExplainPrediction. The top level entry is the explainVis function which explains predictions of a given model and visualizes the explanations. In this section, we explain the parameters of explainVis and show how to call it. We also share some useful tips for using the explanations.

4.1 Controlling explanations

The function explainVis enables fine control over computation and visualization of explanations through its arguments listed in Listing 1 and explained below.

Parameters controlling input/output

model specifies the input prediction model.

trainData is the input data set which is used to compute average explanations, discretization, and other information needed for explanation of instances and model.

testData is the input data set containing instances that are going to be explained.

fileType determines the graphical format of the output visualization file: pdf, eps, emf, jpg, png, bmp, or tif. If "none" is specified, the visualization is directed to a graphical window.

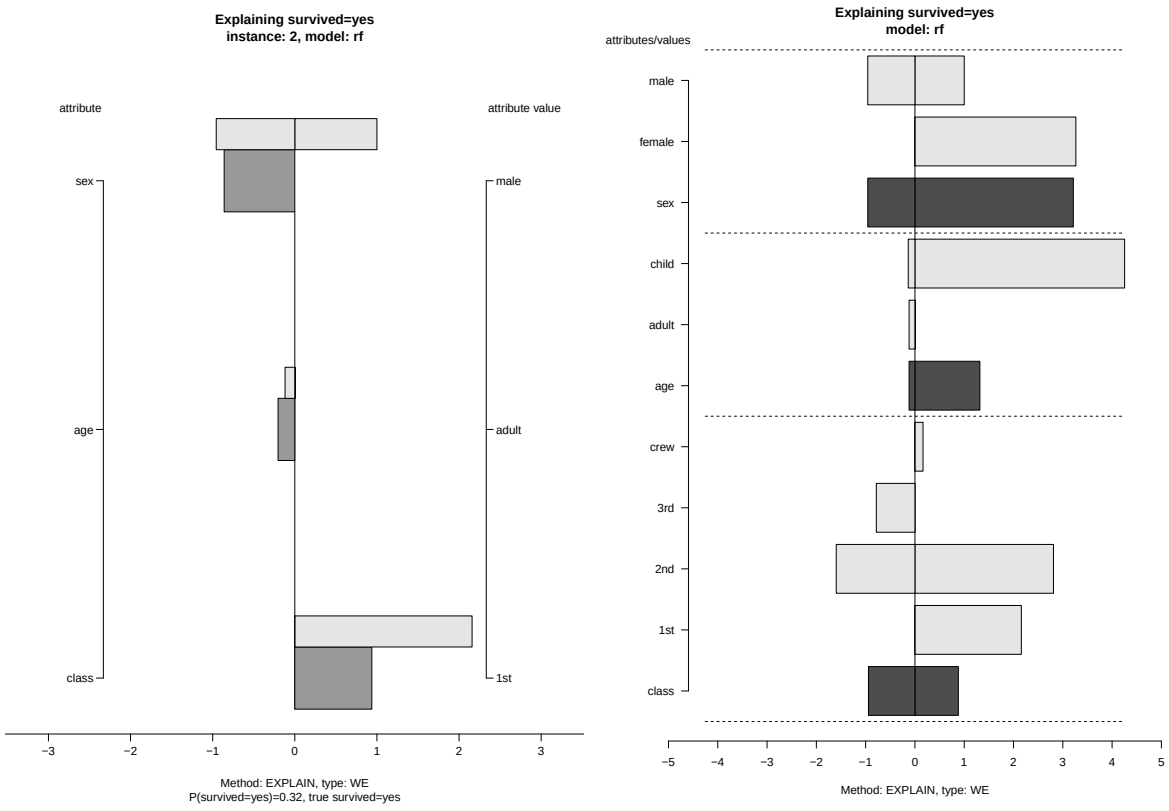
dirName specifies the output folder where resulting visualization files will be saved.

fileName contains the file name of the resulting output visualization files.

The parameters of both explanation methods

method specifies the explanation method, either EXPLAIN or IME.

classValue specifies the class value for which explanations are generated.



a) b)

Figure 1: An instance explanation a) and a model explanation b) for the random forest model classifying the Titanic data set. The tiny bars in the instance explanation represent the average positive and average negative contributions of the values and are equal to the corresponding value-bars in the model explanation (note the difference in scale).

visLevel determines the level of explanations desired, i.e. the model level or instance level explanations.

estimator specifies the feature evaluation method used to greedily discretize attributes needed when averaging explanation over intervals of numeric attributes. The default value NULL invokes discretization with attribute evaluation algorithms ReliefF (classification) or RReliefF (regression) from R package CORElearn (25).

recall can provide the list with all explanations data returned by one of the previous calls to function explainVis, which speeds-up the computations.

Parameters specific to EXPLAIN (see (24))

explainType specifies for the EXPLAIN method how the prediction with knowledge about given feature and the prediction without knowledge of this feature are combined into the final explanation. The values "WE", "infGain", and "predDiff" mean that the difference is interpreted as the weight of evidence, information gain, or plain difference of predictions, respectively. For regression problem only the difference of predictions is available.

naMode specifies for the EXPLAIN method how the impact of missing information about certain feature value is estimated. It can be estimated by the weighted average of predictions over all possible feature's values, or by inserting NA value as a feature value.

nLaplace specifies for the EXPLAIN method and classification problems the value to be used in Laplace correction of estimated probabilities.

Parameters specific to IME (see (30))

pError specifies for the IME method the estimated probability of an error in explanations. Together with the *err* parameter, this determines the number of needed samples.

batchSize specifies for the IME method the number of samples processed for each explanation in one batch. To reduce processing overhead in calls to *predict* function we process several samples at once. This strategy reduces the overhead but may process more samples than required.

maxIter sets the maximal number of iterations in IME method allowed for a single explanation.

Listing 1: Top level call to explanation methods and their visualization.

```
explainVis(model, trainData, testData, method=c("EXPLAIN", "IME"), classValue=1,
    fileType=c("none","pdf","eps","emf","jpg","png","bmp","tif","tiff"), dirName=getwd(), fileName="explainVis",
    visLevel=c("both","model","instance"), explainType=c("WE","infGain","predDiff"), naMode=c("avg", "na"),
    nLaplace=nrow(trainData), estimator=NULL, pError=0.05, err=0.05, batchSize=40, maxIter=1000,
    genType=c("rf", "rbf", "indAttr"), noAvgBins=20, displayAttributes=NULL, modelVisCompact=FALSE,
    displayThreshold=0.0, colors=c("navyblue", "darkred", "blue", "red", "lightblue", "orange"),
    normalizeTo=0, noDecimalsInValueName=2, recall=NULL
    modelTitle=ifelse(model$noClasses==0,"Explaining \%R\nmodel: \%M", "Explaining \%R=\%V\\nmodel: \%M"),
    modelSubtitle="Method: \%E, type: \%X",
    instanceTitle=ifelse(model$noClasses==0, "Explaining \%R\\ninstance: \%I, model: \%M",
                                             "Explaining \%R=\%V\\ninstance: \%I, model: \%M"),
    instanceSubtitle=ifelse(model$noClasses==0, "Method: \%E\\nf(\%I)=\%P, true \%R=\%T",
                                                "Method: \%E, type: \%X\\nP(\%R=\%V)=\%P, true \%R=\%T") )
```

genType specifies the type of data generator used to generate random part of instances in method IME. The generators from R package semiArtificial(22) are used: "rf" stands for the random forest based generator, "rbf" invokes RBF network based generator, and indAttr" assumes independent attributes and generates values for each attribute independently.

noAvgBins specifies for the IME method the number of discretization bins used to present model level explanations and average explanations.

Visualization parameters:

displayAttributes is the vector of attribute names which are visualized in model level visualization.

modelVisCompact determines if attribute values are displayed in model level visualization.

displayThreshold specifies the threshold value for absolute values of explanations below which feature contributions are not displayed in instance and model explanation graphs.

normalizeTo determines the value for instance level visualization to which the sum of the absolute values of feature contributions are normalized (e.g., 1 or 100).

colors determines colors used in visualization.

noDecimalsInValueName specifies how many decimal places will numeric feature values use in visualizations.

modelTitle, *modelSubtitle*, *instanceTitle*, and *instanceSubtitle* are string templates for various titles of different graphs. The template uses several variables, which are inserted at the appropriate place: response variable %R, the selected class value for explanation %V, type of model %M, explanation method %E, explanation type %X, instance name %I, predicted value/probability of the response %P, and the true value of the response %T.

4.2 Producing explanations

The function explainVis generates explanations and their visualizations given the trained model, its training data, and data for which we want explanations. This is the front-end explanation function which takes care of everything, internally calling other functions. The produced visualizations are output to a graphical device or saved to a file. The function returns a list with explanations, average explanations, and additional data like discretization used and data generator. An example of a call is presented in Listing 2.

The explanations support several models implemented in packages CORElearn, randomForest, nnet, and e1071. Adding support for new predictors is easy and involves preparation of class names and class values in the format expected by the package ExplainPrediction when calling the predictor. This is demonstrated in the function wrap4Explanation, which is part of the ExplainPrediction package.

4.3 Tips for using the explanations

The presented explanation techniques have many successful applications (shortly reviewed in Section 2). Here we present a few tips for successful practical use of explanations.

For many real-world problems gaining the trust of users is essential to assure successful application of machine learning models. Instance and model explanation can serve as convenient ice-breakers. If a user can check for some instances that the generated explanations match his/her understanding of the problem, this greatly increases chances of success and is more convincing than reporting high classification accuracy. This is true even for mispredicted instances as long as the model's reasoning is sensible for users.

For larger data sets with many attributes, time to produce explanations with the IME method can be substantial. However, in spite of theoretical advantages of IME over EXPLAIN, in practice, these two methods mostly produce similar explanations. This indicates that in real-world pro-

Listing 2: A code that generates explanations of model and instances.

```
require(ExplainPrediction)
require(CORElearn)
# use iris data set, split it randomly into a training and testing set
trainIdxs <- sample(x=nrow(iris), size=0.7*nrow(iris), replace=FALSE)
testIdxs <- c(1:nrow(iris))[-trainIdxs]
# build random forests model with certain parameters
modelRF <- CoreModel(Species ~ ., iris[trainIdxs,], model="rf", rfNoTrees=100, selectionEstimator="MDL",
                     minNodeWeightRF=5)
# generate model and instance explanations and visualize them in a graphical window
explainVis(modelRF, iris[trainIdxs,], iris[testIdxs,], method="EXPLAIN", fileType="none", naMode="avg",
           explainType="WE", classValue=1)
```

blems there are few redundant attributes and even less redundant attributes of exactly the same strength (if redundant attributes are not of the same strength, learning selects the stronger ones and there is no redundancy in the model). In practice, we can compare the behavior of EXPLAIN and IME on a subsample of instances and attributes. If explanations are similar, the EXPLAIN method can be used instead of IME.

To reach a desired graphical design (e.g., colors and headings) and show only the most impactful attributes requires some tweaking of visualization parameters. To avoid regeneration of explanations for each user interaction with explanations, we provide the *recall* parameter. In the first call to the explainVis function, we have to store the invisibly returned list to a variable and supply this variable as the value of parameter *recall* in subsequent calls to explainVis. In this case the function reuses already computed explanations, average explanations, discretization, etc., and only display data differently according to supplied input/output and visualization parameters (visLevel, dirName, fileType, displayAttributes, modelVisCompact, displayThreshold, normalizeTo, colors, noDecimalsInValueName, modelTitle, modelSubtitle, instanceTitle, and instanceSubtitle). Using this hint can make user interactions with explanations instantaneous even for large data sets.

5 Conclusions

We presented two general methods for explanation of prediction models and their implementation in the ExplainPrediction package. The methods allow explanation of individual decisions as well as the prediction model as a whole. The explanations provide information on how the individual input variables influence the outcome of prediction models, thus improving their transparency and comprehensibility. The general methods allow users to compare different types of models or replace their existing model without having to replace the explanation method. The explanation methods can be efficiently computed and visualized, and their implementation offers several parameters that control the speed and precision of the computed explanations, convergence rate and visualization of explanations. Several models are supported and adding support in almost any prediction model is easy.

The simplicity and elegance of the perturbation based explanations coupled with efficient implementations and visualization of instance- and model-based explanations allow application of general explanation approaches to new areas. We expect that broader practical use will spur additional research into explanation mechanisms and improvements in the visual design of explanations. There are also many possibilities for methodological improvements. An idea worth pursuing seems integration of game theory based sampling and formulation of explanations as an optimization problem. The implementation of IME in the ExplainPrediction package could be improved by rewriting it in C language and using better, context-dependent, sampling method.

Acknowledgment

We acknowledge the support of the Slovenian Research Agency, ARRS, through research programme P2-0209 (Artificial Intelligence and Intelligent Systems) and projects J6-8256 (New grammar of contemporary standard Slovene: sources and methods) and L1-7542 (Advancement of computationally intensive methods for efficient modern general-purpose statistical analysis and inference). Fruitful discussions with Erik Štrumbelj improved the implementation of the IME method.

Literature

[1] Charu C Aggarwal, Chen Chen, and Jiawei Han. The inverse classification problem. *Journal of Computer Science and Technology*, 25(3):458–468, 2010.

[2] Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. What is relevant in a text document?: An interpretable machine learning approach. *PloS ONE*, 12(8):e0181142, 2017.

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and

Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS ONE*, 10(7):e0130140, 2015.

[4] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 11 (Jun):1803–1831, 2010.

[5] David Barbella, Sami Benzaid, Janara M Christensen, Bret Jackson, X Victor Qin, and David R Musicant. Understanding support vector machine classifications via a recommender system-like approach. In R. Stahlbock, S. F. Crone, and S. Lessmann, editors, *Proceedings of International Conference on Data Mining*, pages 305–311, 2009.

[6] Adriano Barbosa, FV Paulovich, Afonso Paiva, Siome Goldenstein, Fabiano Petronetto, and LG Nonato. Visualizing and interacting with kernelized data. *IEEE transactions on visualization and computer graphics*, 22(3):1314–1325, 2016.

[7] Marko Bohanec, Mirjana Borštnar Kljajić, and Marko Robnik-Šikonja. Explaining machine learning models in sales predictions. *Expert Systems with Applications*, 71:416–428, 2017.

[8] Marko Bohanec, Marko Robnik-Šikonja, and Mirjana Kljajić Borštnar. Decision-making framework with double-loop learning through interpretable black-box machine learning models. *Industrial Management & Data Systems*, 117(7):1389–1406, 2017.

[9] Zoran Bosnić, Jaka Demšar, Grega Kešpret, Pedro Pereira Rodrigues, João Gama, and Igor Kononenko. Enhancing data stream predictions with reliability estimators and explanation. *Engineering Applications of Artificial Intelligence*, 34:178–192, 2014.

[10] Jaka Demšar and Zoran Bosnić. Detecting concept drift in data streams using model explanation. *Expert Systems with Applications*, 92:546 – 559, 2018.

[11] Alex Goldstein, Adam Kapelner, Justin Bleich, and Emil Pitkin. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65, 2015.

[12] Lutz Hamel. Visualization of support vector machines with unsupervised learning. In *Proceedings of 2006 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, 2006.

[13] Katja Hansen, David Baehrens, Timon Schroeter, Matthias Rupp, and Klaus-Robert Müller. Visual interpretation of kernel-based prediction models. *Molecular Informatics*, 30(9):817–826, 2011.

[14] Aleks Jakulin, Martin Možina, Janez Demšar, Ivan Bratko, and Blaž Zupan. Nomograms for visualizing support vector machines. In Robert Grossman, Roberto Bayardo, and Kristin P. Bennett, editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 108–117. ACM, 2005.

[15] Vincent Lemaire, Raphael Féraud, and Nicolas Voisine. Contact personalization using a score understanding method. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, 2008.

[16] David Martens and Foster Provost. Explaining documents' classifications. Technical report, Center for Digital Economy Research, New York University, Stern School of Business, 2011. Working paper CeDER-11-01.

[17] David Meyer, Friedrich Leisch, and Kurt Hornik. The support vector machine under test. *Neurocomputing*, 55:169–186, 2003.

[18] Rok Piltaver, Mitja Luštrek, Matjaž Gams, and Sanda Martinčić-Ipšić. What makes classification trees comprehensible? *Expert Systems with Applications*, 62: 333–346, 2016.

[19] Brett Poulin, Roman Eisner, Duane Szafron, Paul Lu, Russell Greiner, David S. Wishart, Alona Fyshe, Brandon Pearcy, Cam Macdonell, and John Anvik. Visual explanation of evidence with additive classifiers. In *Proceedings of AAAI'06*. AAAI Press, 2006.

[20] Marko Pregeljc, Erik Štrumbelj, Miran Mihelčič, and Igor Kononenko. Learning and explaining the impact of enterprises' organizational quality on their economic results. In R. Magdalena-Benedito, M. Martinez-Sober, J. M. Martinez-Martinez, P. Escandell-Moreno, and J. Vila-Frances, editors, *Intelligent Data Analysis for Real-Life Applications: Theory and Practice*, pages 228–248. Information Science Reference, IGI Global, 2012.

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[22] Marko Robnik-Šikonja. Data generators for learning systems based on rbf networks. *IEEE Transactions on Neural Networks and Learning Systems*, 27(5):926–938, May 2016.

[23] Marko Robnik-Šikonja. *ExplainPrediction: Explanation of Predictions for Classification and Regression*, 2017. URL `http://cran.r-project.org/package=ExplainPrediction`. R package version 1.3.0.

[24] Marko Robnik-Šikonja and Igor Kononenko. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600, 2008.

[25] Marko Robnik-Šikonja and Petr Savicky. *CORElearn - classification, regression, feature evaluation and ordinal evaluation*, 2017. URL `http://cran.r-project.org/package=CORElearn`. R package version 1.52.0.

[26] Andrea Saltelli, Karen Chan, and E. Marian Scott. *Sensitivity analysis*. Wiley, Chichester; New York, 2000.

[27] Alexander Schulz, Andrej Gisbrecht, and Barbara Hammer. Using discriminative dimensionality reduction to visualize classifiers. *Neural Processing Letters*, 42(1):27–54, 2015.

[28] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[29] Erik Štrumbelj, Zoran Bosnić, Igor Kononenko, Branko Zakotnik, and Cvetka Grašič Kuhar. Explanation and reliability of prediction models: the case of breast cancer recurrence. *Knowledge and information systems*, 24(2):305–324, 2010.

[30] Erik Štrumbelj and Igor Kononenko. An Efficient Explanation of Individual Classifications using Game Theory. *Journal of Machine Learning Research*, 11: 1–18, 2010.

[31] Erik Štrumbelj, Igor Kononenko, and Marko Robnik-Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data & Knowledge Engineering*, 68(10):886–904, 2009.

Semantic Annotation of Documents Based on Wikipedia Concepts

Janez Brank, Gregor Leban and Marko Grobelnik
Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia
E-mail: janez.brank@ijs.si, gregor.leban@ijs.si, marko.grobelnik@ijs.si

Keywords: semantic annotation, wikification, disambiguation, text mining

Received: January 30, 2017

Semantic annotation is the task of augmenting an unstructured textual document with semantic information, such as concepts from an ontology. In wikification, the Wikipedia is used as an ontology and its pages (articles) are regarded as (representations of) concepts. We describe an efficient approach for annotating a document with relevant concepts from the Wikipedia. A global disambiguation method based on constructing a mention-concept graph and computing pagerank over it is used to identify a coherent set of relevant concepts considering the input document as a whole. The presented approach is suitable for parallel processing and can support any language for which a sufficiently large Wikipedia is available. Several heuristics involved in the disambiguation of candidate annotations are discussed and an experimental evaluation of their influence is presented.

Povzetek: Semantično anotiranje je postopek, s katerim bi radi nestrukturirano besedilo dopolnili s semantičnimi informacijami, na primer s koncepti iz neke ontologije. Pri wikifikaciji se kot ontologijo uporablja Wikipedijo, pri čemer strani oz. članke v njej obravnavamo kot predstavitve konceptov. Opisujemo učinkovit pristop za anotiranje besedila z relevantnimi koncepti iz Wikipedije. Pri tem uporabljamo globalen pristop k razdvoumljanju, ki temelji na izgradnji grafa omemb in konceptov ter računanju pageranka na tem grafu, kar je nato podlaga za določitev nabora konceptov, ki se lepo skladajo med seboj in so relevantni glede na vhodno besedilo kot celoto. Opisani pristop je primeren za paralelno procesiranje in deluje za poljuben jezik, v katerem je na voljo dovolj velika Wikipedija. V članku predstavljamo in eksperimentalno ovrednotimo tudi več hevristik, ki se jih lahko uporabi pri razdvoumljanju kandidatov za anotacije.

1 Introduction

Recent years have seen a growth in the use of semantic technologies. However, in many contexts we still deal with largely unstructured textual documents that lack explicit semantic information which might be required for further processing with semantic technologies. This leads to the problem of semantic annotation or semantic enrichment as an important preparatory step before further processing of a document. Given a document and an ontology covering the domain of interest, the challenge is to identify concepts from that ontology that are relevant to the document or that are referred to by it, as well as to identify specific passages in the document where the concepts in question are mentioned.

A specific type of semantic annotation, known as *wikification*, involves using the Wikipedia as a source of possible semantic annotations [1][2]. In this setting, the Wikipedia is treated as a large and fairly general-purpose ontology: each page is thought of as representing a concept, while the relations between concepts are represented by internal hyperlinks between different Wikipedia pages, as well as by Wikipedia's category memberships and cross-language links.

The advantage of this approach is that the Wikipedia is a freely available source of information, it covers a wide range of topics, has a rich internal structure, and each concept is associated with a semi-structured textual document (i.e. the content of the corresponding Wikipedia article) which can be used to aid in the process of semantic annotation. Furthermore, the Wikipedia is available in a number of languages, with cross-language links being available to identify pages that refer to the same concept in different languages, thus making it easier to support multilingual and cross-lingual annotation.

The remainder of this paper is structured as follows. In Section 2, we present the pagerank-based approach to wikification used in our wikifier. In Section 3, we describe our implementation and some experimental evaluaton. Section 4 contains conclusions and a discussion of possible future work.

2 Pagerank-based Wikification

The task of wikifying an input document can be broken down into several closely interrelated subtasks: (1) identify phrases (or words) in the input document that refer to a Wikipedia concept; (2) determine which concept exactly a phrase refers to; (3) determine which concepts are relevant enough to the document as a whole that they should be included in the output of the system (i.e. presented to the user).

We follow the approach described by Zhang and Rettinger [1]. This approach makes use of the rich internal structure of hyperlinks between Wikipedia pages. A hyperlink can be thought of as consisting of a

source page, a target page, and the link text (also known as the anchor text). If a source page contains a link with the anchor text a and the target page *t*, this is an indication that the phrase a might be a reference to (or representation of) the concept that corresponds to page *t*. Thus, if the input document that we're trying to wikify contains the phrase *a*, it might be the case that this occurrence of *a* in the input document also constitutes a mention of the concept *t*, and the concept *t* is a candidate annotation for this particular phrase.

2.1 Disambiguation

In the Wikipedia, there may be many different links with the same anchor text *a*, and they might not all be pointing to the same target page. For example, in the English-language Wikipedia, there are links with *a* = "Tesla" that point to pages about the inventor, the car manufacturer, the unit in physics, a band, a film, and several other concepts.

Thus, when such a phrase a occurs in an input document, there are several concepts that can be regarded as candidate annotations for that particular mention, and we have to determine which of them is actually relevant. This is the problem of disambiguation, similar to that of word sense disambiguation in natural language processing.

There are broadly two approaches to disambiguation, local and global. In the local approach, each mention is disambiguated independently of the others, while the global approach aims to treat the document as a whole and disambiguate all the mentions in it as a group. The intuition behind the global approach is that the document that we're annotating is about some topic, and the concepts that we use as annotation should be about that topic as well. If the document contains many mentions that include, as some of their candidate annotations, some car-related concepts, this makes it more likely that we should treat the mention of "Tesla" as a reference to Tesla the car manufacturer as opposed to e.g. a reference to Nikola Tesla or to Tesla the rock band.

2.2 The mention-concept graph

To implement the global disambiguation approach, our Wikifier begins by constructing a *mention-concept graph* for the input document. (Some authors, e.g. [2], refer to this as a *mention-entity* graph, but we prefer to use the term "mention-concept graph" as some of the Wikipedia pages do not necessarily correspond to concepts that we usually think of as entities, and our wikifier does not by default try to exclude them.) This can be thought of as a bipartite graph in which the left set of vertices corresponds to mentions and the right set of vertices corresponds to concepts. A directed edge $a \rightarrow c$ exists if and only if the concept *c* is one of the candidate annotations for the mention *a* (i.e. if, in the Wikipedia, there exists a hyperlink with the anchor text *a* and the target *c*). A transition probability is also assigned to each such edge, $P(a \rightarrow c)$, defined as the ratio [number of hyperlinks, in the Wikipedia, having the anchor text *a* and the target *c*] / [number of hyperlinks, in the Wikipedia, having the anchor text *a*].

This graph is then augmented by edges between concepts, the idea being that an edge $c \rightarrow c'$ should be used to indicate that the concepts *c* and *c'* are "semantically related", in the sense that if one of them is relevant to a given input document, the other one is also more likely to be relevant to that document. (For example, the semantic relationship between the concepts "Electric vehicle" and "Tesla Inc." should be much stronger than between the concepts "Electric vehicle" and "Tesla (rock band)". This measure of semantic relatedness will be used subsequently to encourage the formation of a group of annotations that are semantically related in the sense that they refer to the same topic, which is hopefully also the topic of the document. This would encourage mentions of "Tesla" in a document about electric cars to be annotated with the concept "Tesla Inc." rather than "Tesla (rock band)".

Following [1], the internal link structure of the Wikipedia is used to calculate a measure of semantic relatedness. Informally, the idea is that if *c* and *c'* are closely related, then other Wikipedia pages that point to *c* are likely to also point to *c'* and vice versa. Let L_c be the set of Wikipedia pages that contain a hyperlink to *c*, and let *N* be the total number of concepts in the Wikipedia; then the semantic relatedness of *c* and *c'* can be defined as

$$SR(c, c') = 1 - \frac{\log \max\{|L_c|, |L_{c'}|\} - \log|L_c \cap L_{c'}|}{\log N - \log \min\{|L_c|, |L_{c'}|\}}.$$

Intuitively, this formula considers two concepts to be semantically related if pages that link to one of them typically also link to the other one (and vice versa). More specifically, *SR* will be higher if the overlap (i.e. the intersection) of L_c and $L_{c'}$ is large (relative to the size of L_c and $L_{c'}$), and the formula also rewards situations where the sets L_c and $L_{c'}$ are themselves large (relative to the overall number of documents *N*), as this means that the dataset includes more evidence of a semantic relationship between *c* and *c'*.

In the graph, we add an edge of the form $c \rightarrow c'$ wherever the semantic relatedness $SR(c, c')$ is > 0. The transition probability of this edge is defined as proportional to the semantic relatedness: $P(c \rightarrow c') = SR(c, c') / \Sigma_{c''} SR(c, c'')$.

This graph is then used as the basis for calculating a vector of pagerank scores [3], one for each vertex. This is done using the usual iterative approach where in each iteration, each vertex distributes its pagerank score to its immediate successors in the graph, in proportion to the transition probabilities on its outgoing edges:

$$PR_{new}(u) = \tau\, PR_0(u) + (1 - \tau)\, \Sigma_v\, PR_{old}(v)\, P(v \rightarrow u).$$

The baseline distribution of pagerank, PR_0, is used both to help the process converge and also to counterbalance the fact that in our graph there are no edges pointing into the mention vertices. In our case, $PR_0(u)$ is defined as 0 if *u* is a concept vertex; if *u* is a mention vertex, we use $PR_0(u) = z \cdot$ [number of

Wikipedia pages containing the phrase *u* as the anchor-text of a hyperlink] / [number of Wikipedia pages containing the phrase *u*], where *z* is a normalization constant to ensure that $\Sigma_u PR_0(u) = 1$. We used $\tau = 0.1$ as the stabilization parameter.

The intuition behind this approach is that in each iteration of the pagerank calculation process, the pagerank flows into a concept vertex *c* from mentions that are closely associated with the concept *c* and from other concepts that are semantically related to *c*. Thus after a few iterations, pagerank should tend to accumulate in a set of concepts that are closely semantically related to each other and that are strongly associated with words and phrases that appear in the input document, which is exactly what we want in the context of global disambiguation.

2.3 Using pagerank for disambiguation

Once the pagerank values of all the vertices in the graph have been calculated, we use the pagerank values of concepts to disambiguate the mentions. If there are edges from a mention *a* to several concepts *c*, we choose the concept with the highest pagerank as the one that is relevant to this particular mention *a*. We say that this concept is *supported* by the mention *a*. At the end of this process, concepts that are not supported by any mention are discarded as not being relevant to the input document.

The remaining concepts are then sorted in decreasing order of their pagerank. Let the *i*'th concept in this order be c_i and let its pagerank be PR_i, for $i = 1, \ldots, n$. Concepts with a very low pagerank value are less likely to be relevant, so it makes sense to apply a further filtering step at this point and discard concepts whose pagerank is below a user-specified threshold. However, where exactly this threshold should be depends on whether the user wants to prioritize precision or recall. Furthermore, the absolute values of pagerank can vary a lot from one document to another, e.g. depending on the length of the documents, the number of mentions and candidate concepts, etc. Thus we apply the user-specified threshold in the following manner: given the user-specified threshold value $\theta \in [0, 1]$, we output the concepts $c_1, \ldots, c_m$, where *m* is the least integer such that $\Sigma_{i=1..m} PR_i^2 \geq \theta \ \Sigma_{i=1..n} PR_i^2$. In other words, we report as many top-ranking concepts as are needed to cover θ of the total sum of squared pageranks of all the concepts. We use $\theta = 0.8$ as a broadly reasonable default value, though the user can require a different threshold depending on their requirements.

The motivation for using squares of pageranks instead of the pageranks themselves is to put a greater emphasis on the annotations with the highest values of pagerank, while culling the lower-scoring annotations more thoroughly. In our preliminary experiments, this led to a small improvement in performance compared to using the sums of pageranks without squaring them.

For each reported concept, we also output a list of the mentions that support it.

2.4 Treatment of highly ambiguous mentions

Our wikifier supports various minor heuristics and refinements in an effort to improve the performance of the baseline approach described in the preceding sections.

As described above, anchor text of hyperlinks in the Wikipedia is used to identify mentions in an input document (i.e. words or phrases that may support an annotation). One downside of this approach is that some words or phrases occur as the anchor text of a very large number of hyperlinks in the Wikipedia and these links point to a large number of different Wikipedia pages. In other words, such a phrase is highly ambiguous; it is not only unlikely to be disambiguated correctly, but also introduces noise into the mention-concept graph by introducing a large number of concept vertices, the vast majority of which will be completely irrelevant to the input document. This also slows down the annotation process by increasing the time to calculate the semantic relatedness between all pairs of candidate concepts. (As an example of such a highly ambiguous mention, consider the word "Country". Most of the time, when it appears as the anchor-text of a link, it's a link to the concepts "Country" or "Country music", but it also occurs in links to more than a hundred other concepts, mostly individual countries.)

We use several heuristics to deal with this problem. Suppose that a given mention *a* occurs, in the Wikipedia, as the anchor text of *n* hyperlinks pointing to *k* different target pages, and suppose that n_i of these links point to page c_i (for $i = 1, \ldots, k$). We can now define the entropy of the mention *a* as the amount of uncertainty regarding the link target given the fact that its anchor text is *a*: $H(a) = -\Sigma_{i=1..k} (n_i/n) \log(n_i/n)$. If this entropy is above a user-specified threshold (e.g. 3 bits), we completely ignore the mention as being too ambiguous to be of any use. For mentions that pass this heuristic, we sort the target pages in decreasing order of n_i and use only the top few of them (e.g. top 20) as candidates in our mention-concept graph. A third heuristic is to ignore candidates for which n_i itself is below a certain threshold (e.g. $n_i < 2$), the idea being that if such a phrase occurs only once as the anchor text of a link pointing to that candidate, this may well turn out to be noise and is best disregarded.

Optionally, the Wikifier can also be configured to ignore certain types of concepts based on their Wikidata class membership. This can be useful to exclude from consideration Wikipedia pages that do not really correspond to what is usually thought of as entities (e.g. "List of…" pages).

Another heuristic that we have found useful in reducing the noise in the output annotations is to ignore any mention that consists entirely of stopwords and/or very common words (top 200 most frequent words in the Wikipedia for that particular language). For this as well as for other purposes the text processing is done in a case-sensitive fashion, which e.g. allows us to ignore spurious links with the link text "the" while processing those that refer to the band "The The".

2.5 Miscellaneous heuristics

Semantic relatedness. As mentioned above, the definition of semantic relatedness of two concepts, $SR(c, c')$, is based on the overlap between the sets L_c, $L_{c'}$ of immediate predecessors of these two concepts in the Wikipedia link graph. Optionally, our Wikifier can compute semantic relatedness using immediate successors or immediate neighbours (i.e. both predecessors and successors) instead of immediate predecessors. However, our preliminary experiments indicated that these changes do not lead to improvements in performance, so they are disabled by default.

Extensions to disambiguation. Our Wikifier also supports some optional extensions of the disambiguation process. As described above, the default behavior when disambiguating a mention is to simply choose the candidate annotation with the highest pagerank value. Alternatively, after any heuristics from section 2.4 have been applied, the remaining candidate concepts can be re-ranked using a different scoring function that takes other criteria besides pagerank into account. This is an opportunity to combine the global disambiguation approach with some local techniques. In general, a scoring function of the following type is supported:

$$score(c|a) = w_1 f(P(c|a)) PR(c) + w_2 S(c, d) + w_3 LS(c, a) \quad (1)$$

Here, a is the mention that we're trying to disambiguate, and c is the candidate concept that we're evaluating. $P(c|a)$ is the probability that a hyperlink in the Wikipedia has c as its target conditioned on the fact that it has a as its anchor text. $f(x)$ can be either 1 (the default), x, or $\log(x)$. $PR(c)$ is the pagerank of c's vertex in the mention-concept graph. $S(c, d)$ is the cosine similarity between the text of the input document d and of the Wikipedia page for the concept c. $LS(c, a)$ is the cosine similarity between the context (e.g. previous and next 3 words) in which a appears in the input document d, and the contexts in which hyperlinks with the target c appear in the Wikipedia. Finally, w_1, w_2, w_3 are weight constants. However, our preliminary experiments haven't shown sufficient improvements from the addition of these heuristics, so they are disabled by default ($f(x) = 1$, $w_2 = w_3 = 0$) to save computational time and memory (storing the link contexts needed for the efficient computation of LS has turned out to be particularly memory intensive).

3 Implementation and evaluation

3.1 Implementation

Our implementation of the approach described in the preceding section is running as a web service and can be accessed at http://wikifier.org. The approach is suitable for parallel processing as annotating one document is independent of annotating other documents, and any shared data used by the annotation process (e.g. the Wikipedia link graph, and a trie-based data structure that indexes the anchor text of all the hyperlinks) need to be accessed only for reading and can thus easily be shared by an arbitrary number of worker threads. This allows for a highly efficient processing of a large number of documents.

The only need to modify shared data structures arises when a new dump of the Wikipedia becomes available (the Wikipedia publishes new dumps of its content twice per month). We use a separate process to periodically check the Wikipedia web site for new dumps, download them, parse them, and build indexes in a form that can be used by our wikifier. Once the wikifier web service is notified of the availability of a new index, it loads its contents into memory, temporarily stops issuing new requests to worker threads, waits for them all to finish processing their current requests, and then updates the shared data structures to include the new index and discard the old one. In this way, new indices can be brought online without shutting down the service and with a minimal interruption to its availability.

Our implementation currently processes on average more than 500,000 requests per day (the total length of input documents averages about 1.2 GB per day), including all the documents from the JSI Newsfeed service [4]. The output is used among other things as a preprocessing step by the Event Registry system [5]. The wikifier currently supports all languages in which a Wikipedia with at least 1000 pages is available, amounting to a total of 134 languages. Admittedly, 1000 pages is much too small to achieve an adequate coverage; however, about 60 languages have a Wikipedia with at least 100,000 pages, which is enough for many practical applications.

Annotations are returned in JSON format and can optionally include detailed information about support (which mentions support each annotation), alternative candidate annotations (concepts that were considered as candidates during the disambiguation process but were rejected in favour of some other higher scored concept), and WikiData/DbPedia class membership of the proposed annotations. Thus, the caller can easily implement any desired class-based post-processing.

Our wikifier also allows the user to define custom vocabularies that can be used to generate annotations in addition to the Wikipedia-based annotations described so far. A custom vocabulary is a set of concepts where each concept consists of an ID and a set of one or more words of phrases which, if they appear in the input document, trigger the inclusion of this concept among the annotations. This allows the user to extend the system with custom sets of annotations, but the downside is that such custom annotations are not part of the Wikipedia and thus cannot be included in the usual wikification process, especially not in the pagerank-based disambiguation algorithm.

As a preprocessing step, the user may specify one or more sets of "alternative labels", which are really rewriting rules of the form "$w_1\ w_2\ \ldots\ w_n \rightarrow x_1\ x_2\ \ldots\ x_m$" indicating that the sequence of the words $w_1\ w_2\ \ldots\ w_n$ may, if it occurs in the input document, be replaced by the sequence $x_1\ x_2\ \ldots\ x_m$ prior to the main part of the

wikification algorithm. (The word "may" in the preceding sentence means that the original sequence of words from the left-hand side of the rule is also kept in the document. Thus, the document is no longer a simple sequence of words, but may gradually turn into an arbitrary directed acyclic graph, the various paths through which indicate different alternatives into which the text of the document may be brought through the application of the rewriting rules.) Owing to such transformations, certain candidate mentions might appear in the document that did not appear in the original document. Several such rules may be applied one after another and may affect the same part(s) of a document. Theoretically, such rewriting rules form a Turing-complete formalism, and to keep the problem tractable our wikifier makes only three passes through the document to look for occurrences of left-side word sequences and replace them with the corresponding right-side word sequences. Currently the main use of this mechanism in our wikifier is to provide additional alternative spellings of some proper names in cases where these are not adequately covered in the Wikipedia. This has been found to be particularly useful in case of names transliterated from languages that use a different script and where several different transliteration schemes are in use.

3.2 Evaluation

One way to evaluate wikification is to compare the set of annotations with a manually annotated gold standard for the same document(s). Performance can then be measured using metrics from information retrieval, such as precision, recall, and the F_1-measure, which is defined as the harmonic mean of precision and recall. We used two manually annotated datasets:

(Dataset 1.) A set of 1393 news articles that was made available from the authors of the AIDA system and was originally used in their experiments [2]. This manually annotated dataset excludes, by design, any annotations that do not correspond to named entities. Since our wikifier does not by default distinguish between named entities and other Wikipedia concepts, we have explicitly excluded concepts that are not named entities (based on their class membership in the WikiData ontology) from the output of our Wikifier for the purposes of this experiment.

(Dataset 2.) A set of 491 news articles taken randomly from the JSI Newsfeed [4] on 29 June 2016 and annotated manually with relevant Wikipedia concepts. Unlike the first dataset, the annotations here included concepts that were not named entities.

In addition to our wikifier, we obtained annotations from the following systems: AIDA [2], Waikato Wikipedia Miner [7], Babelfy [8], Illinois [9], and DbPedia Spotlight [10]. The Waikato system is not included in experiments involving dataset 2 as their web service was no longer available at the time.

Tables 1(*a*) and 1(*b*) show the agreement not only between each of the wikifiers and the gold standard, but also between each pair of wikifiers (the lower left triangle of the matrix is left empty as it would be just a copy of the upper right triangle, since the F_1-measure is symmetric). As this experiment indicates, on the first dataset (the AIDA dataset) our wikifier ("JSI" in the table) performs slightly worse than AIDA but significantly better than the other wikifiers. On the second dataset (the JSI dataset), the best performance was achieved by the Babelnet wikifier, ours is slightly worse while AIDA is significantly worse on this dataset. Thus, overall we can conclude that our wikifier has solid performance over a pair of two considerably different dataset. Furthermore, experiments on both datasets show that there is relatively little agreement between different wikifiers, which indicates that wikification itself is in some sense a vaguely defined task where different people can have very different ideas about whether a particular Wikipedia concept is relevant to a particular input document (and should therefore be included as an annotation) or not, which types of Wikipedia concepts can be considered as annotations (e.g. only named entities or all concepts), etc. Possibly the level of agreement could be improved by fine-tuning the settings of the various wikifiers; in the experiment described above, default settings were used.

	Gold	JSI	AIDA	Waikato	Babelfy	Illinois	Spotlight
Gold	1.000	0.593	0.723	0.372	0.323	0.476	0.279
JSI		1.000	0.625	0.527	0.431	0.489	0.363
AIDA			1.000	0.372	0.352	0.434	0.356
Waikato				1.000	0.481	0.564	0.474
Babelfy					1.000	0.434	0.356
Illinois						1.000	0.376
Spotlight							1.000

Table 1(*a*): F_1 measure of agreement between the various wikifiers and the gold standard on dataset 1.

	Gold	JSI	AIDA	Babelfy	Illinois	Spotlight
Gold	1.000	0.378	0.197	0.417	0.372	0.282
JSI		1.000	0.278	0.360	0.413	0.397
AIDA			1.000	0.206	0.283	0.383
Babelfy				1.000	0.380	0.282
Illinois					1.000	0.367
Spotlight						1.000

Table 1(*b*): F_1 measure of agreement between the various wikifiers and the gold standard on dataset 2.

We also conducted a small experiment on dataset 2 to compare two forms of the thresholding criterion: one is based on the sums of squares of pageranks (as currently described in Section 2.3) and one based on the sums of the pageranks themselves. The F_1-measure between our annotations and the gold standard drops from 0.378 when using squared pageranks to 0.344 when using the pageranks directly. We used squared pageranks for thresholding in all other experiments in this section.

Evaluation of disambiguation heuristics. In the following experiment, we evaluate some of the additional disambiguation heuristics described in Section 2.5. The purpose of the experiment was to find the best-performing combination of the following heuristics and parameters from that section:

(*i*) Logarithmic link counts: in Section 2.2, we defined the transition probability $a \rightarrow c$ in the mention-concept graph as being proportional to the number of

links, in the Wikipedia, with the anchor text *a* and the target *c* (the "link count" of *c* given *a*). Alternatively, it can be defined as being proportional to the *logarithm* of this link count. The purpose of this heuristic is to provide a kind of smoothing and discourage too much of the pagerank score from flowing into just one candidate *c* for that particular mention *a*. The Wikipedia is known for having various biases in terms of how frequently certain topics are covered, so this sort of smoothing may soften the more extreme differences in the frequency of coverage while still preserving some information about which concepts *c* are associated more often with a phrase *a*.

(*ii*) Set of links used in the computation of sematic relatedness (*SR*) between two concepts in the Wikipedia link graph: this can be the in-links (the default setting), out-links, or all neighbours.

(*iii*) Threshold for re-ranking: in this scenario, the candidates *c* for a given mention *a* are first sorted by pagerank, the top few candidates are kept and are then re-ranked using the more detailed (and computationally-intensive) scoring function denoted by eq. (1). The question then is what counts as "top few candidates" to be included in the re-ranking. We define this by introducing a parameter $\vartheta \in [0, 1]$ such that a candidate *c* proceeds to re-ranking if its pagerank is $PR(c) \geq \vartheta \max_{c'} PR(c')$, where c' goes over all the candidates for the current mention *a*.

(*iv*) Linearization of pagerank in the scoring function denoted by eq. (1) into a linear rank: instead of using the pagerank directly, all the candidate concepts *c* for a given mention *a* are sorted by pagerank and a linear rank is assigned to each. If there are *k* candidates, the *i*'th of them in this order gets a linear rank of *i*/*k*. This is then used instead of *PR*(*c*) in eq. (1), as well as in the ϑ-based thresholding criterion in the previous paragraph (where ϑ then simply becomes the proportion of candidates that proceeds to the re-ranking phase). The purpose is to make sure that the range [0, 1] is covered evenly, instead of the pagerank values possibly being clustered in a small part of that range.

(*v*) Weight w_2 of the cosine similarity between the input document and the Wikipedia page of a candidate concept, in the scoring function of eq. (1). (The weight w_1 of the candidate concept's pagerank value in the scoring function was then set to $1 - w_2$. The weight w_3 of the link context similarity was kept to 0 throughout these experiments, because of the considerable additional memory and time consumption required for the link context computation and because preliminary experiments indicated that the results were not promising.)

The possible values of these five parameters that were investigated in this experiment can be summarized as follows:

(*i*) linkCounts ∈ {normal, log}
(*ii*) SR ∈ {in, out, all}
(*iii*) $\vartheta \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$
(*iv*) PR ∈ {normal, linearized}
(*v*) $w_2 \in \{0, 0.25, 0.5, 0.75, 1\}$

The default settings are: normal link counts, SR = in, $\vartheta = 1$ (no second-stage re-ranking), PR = normal, $w_2 = 0$.

Table 2 shows, for each possible value of each parameter, the best and the average performance (in terms of F_1-measure relative to the gold standard) that can be achieved by fixing that parameter to that value and allowing the other parameters to range over all the possible values indicated above.

For comparison, the last two rows show the performance with no parameters fixed (allowing us to tune the best possible combination of all parameters) and the performance with all parameters fixed at their default values.

This experiment was done on Dataset 1 and used 10-fold cross-validation. Nine folds (the training set) were used to tune any parameters that were not held fixed and the best resulting combination of parameters was then evaluated on the tenth fold (the test set). This was repeated for all ten choices of the test fold. Table 2 shows the average and the standard deviation of the F_1 performance on the test fold over all 10 choices of the test fold.

As we can see from this experiment, it is indeed possible to achieve a small improvement in performance by employing some of the heuristics described here. The best-performing combination of parameters was {normal link counts, SR = in, $\vartheta = 0.6$, PR = normal, $w_2 = 1$}, which resulted in an F_1 score of 0.6152, up from the score of 0.5917 achieved by the default parameter values. A paired *t*-test showed this difference to be significant at a *p*-value of 0.0005. However, we can also see that, in practical terms, this improvement is small and might not be noticed by the user. Furthermore, it is clear that several of the heuristics employed were in fact counterproductive: using logarithms of link counts to define the mention-concept transition probabilities; using out-links or all neighbors (instead of just in-links) in the definition of semantic relatedness; and using linearized pageranks. Shifting these parameters away from their default settings in fact led to a deterioration of F_1 (in all these cases, the deterioration is statistically significant with a *p*-value of 0.001 or less.) Improvements in performance mostly came from re-ranking the most promising candidates ($\vartheta = 0.6$) based on the cosine similarity between the input document and the Wikipedia pages of the candidate concepts.

The "Average F_1" column of Table 2 shows that no parameter by itself can ensure good performance unless the other parameters are also chosen suitably, as the average performance over all the combinations of other parameters is poor regardless of which parameter has been fixed and at what value.

4 Use in a real-life application

Semantically annotating documents can be of high importance in several real-life applications. An example of such an application is Event Registry [5]. Event Registry is a system that collects and analyzes news content generated globally and identifies the world

Parameter	Avg. F_1	Max. F_1
linkCounts = normal	0.5883 ± 0.0253	0.6152 ± 0.0239
linkCounts = log	0.5368 ± 0.0268	0.5833 ± 0.0221
SR = in	0.5800 ± 0.0232	0.6152 ± 0.0239
SR = out	0.5626 ± 0.0265	0.5945 ± 0.0253
SR = all	0.5451 ± 0.0283	0.5927 ± 0.0268
PR = normal	0.5644 ± 0.0261	0.6152 ± 0.0239
PR = linearized	0.5607 ± 0.0259	0.5978 ± 0.0223
$\vartheta = 0$	0.5604 ± 0.0256	0.5974 ± 0.0223
$\vartheta = 0.2$	0.5614 ± 0.0256	0.5982 ± 0.0221
$\vartheta = 0.4$	0.5634 ± 0.0258	0.6054 ± 0.0225
$\vartheta = 0.6$	0.5649 ± 0.0259	0.6152 ± 0.0239
$\vartheta = 0.8$	0.5642 ± 0.0260	0.6004 ± 0.0216
$\vartheta = 1$	0.5610 ± 0.0273	0.5939 ± 0.0280
$w_2 = 0$	0.5610 ± 0.0273	0.5939 ± 0.0280
$w_2 = 0.25$	0.5646 ± 0.0266	0.5979 ± 0.0209
$w_2 = 0.5$	0.5655 ± 0.0265	0.6060 ± 0.0228
$w_2 = 0.75$	0.5654 ± 0.0253	0.6128 ± 0.0248
$w_2 = 1$	0.5563 ± 0.0245	0.6152 ± 0.0239
Nothing fixed	0.5626 ± 0.0260	**0.6152 ± 0.0239**
All fixed to default values	0.5917 ± 0.0226	0.5917 ± 0.0226

Table 2**:** F_1 measure of agreement between our wikifier and the gold standard while keeping one parameter fixed and tuning the others. "Avg. F_1" shows average performance over all possible combinations of non-fixed parameters; "Max. F_1" shows the best performance achieved by tuning the non-fixed parameters on the training folds. Both columns show the F_1 performance on the test fold. Since cross-validation was used, the performances are shown in the form "average ± standard deviation" over all 10 possible splits of the data into 9 training folds and 1 test fold.

events mentioned in the news. As the application aims to extract knowledge in structured form from the unstructured text, we will now describe how the Wikifier's semantic annotations provide the critical input required by the system.

For each news article, Event Registry stores the list of identified semantic annotations. Among other things, this allows the users to search for news content using the semantic tags and not keywords, as we are used to in the search engines. The main advantages of using tags versus keywords are that one can e.g. (*a*) specifically ask for articles about apple, the fruit, versus Apple, the company, (*b*) find articles about IBM regardless of how it's mentioned in the news articles ("IBM", "I.B.M.", "International Business Machines", etc.), and (*c*) find articles about White House in any language. The last use case is available because the Wikipedia also maintains information on which Wikipedia pages in different languages represent the same concept. Consequently, the tag for "White House" in an English article will be the same, as the tag for "Casa Blanca" in a Spanish article.

When Event Registry identifies a group of news articles that represent the same event, it uses the semantic information in the news articles to determine the core event information. First, it analyzes all news articles in the event and calculates how frequently individual concepts appear in these articles. A ranked list of commonly mentioned concepts is then used as a semantic summary or a "fingerprint" of an event.

Another critical piece of information about the event is its geographical location. In order to determine the location, Event Registry again analyzes concepts mentioned in the news articles about the event, and considers as possible candidates only those that refer to a geographical location. For each candidate location, a set of learning features is extracted. The learning features that we extract are as follows:

- Mentions of the location in the articles about the event. The value is simply the ratio of the number of news articles about the event that mention the location somewhere in the text and the total number of articles about the event.
- Mentions of the location in the dateline (beginning of the article). This feature is computed as the ratio of the number of articles about the event in which the location is mentioned in the dateline and the total number of articles about the event.
- Normalized versions of the previous two features. In this case we compute a variation of the previous two features, where we don't compute a simple ratio, but weight the contribution of an individual article by the cosine similarity of the article to the centroid of the event. Articles closer to the centroid (more relevant articles) therefore contribute more to the final feature value.
- Commonality of the location — how frequently is the location generally present in the news articles. The value is computed as the ratio of the number of articles in Event Registry that mention this location and the total number of articles.

Based on these features, a logistic regression model computes a probability for each of the candidate locations to be the location of the event. If the location with the highest probability is above the predetermined threshold, the location is chosen as the location of the event. The logistic model was trained on 1239 manually labeled events and has 96.2% classification accuracy.

Semantic annotations are also of high importance when a search is performed and a large number of results need to be summarized. An example of such a summary is displayed in Figure 1, where we searched for events about hurricanes. The resulting list that contained over 23 000 events was summarized as shown in the figure to illustrate what are the top concepts mentioned in these events.

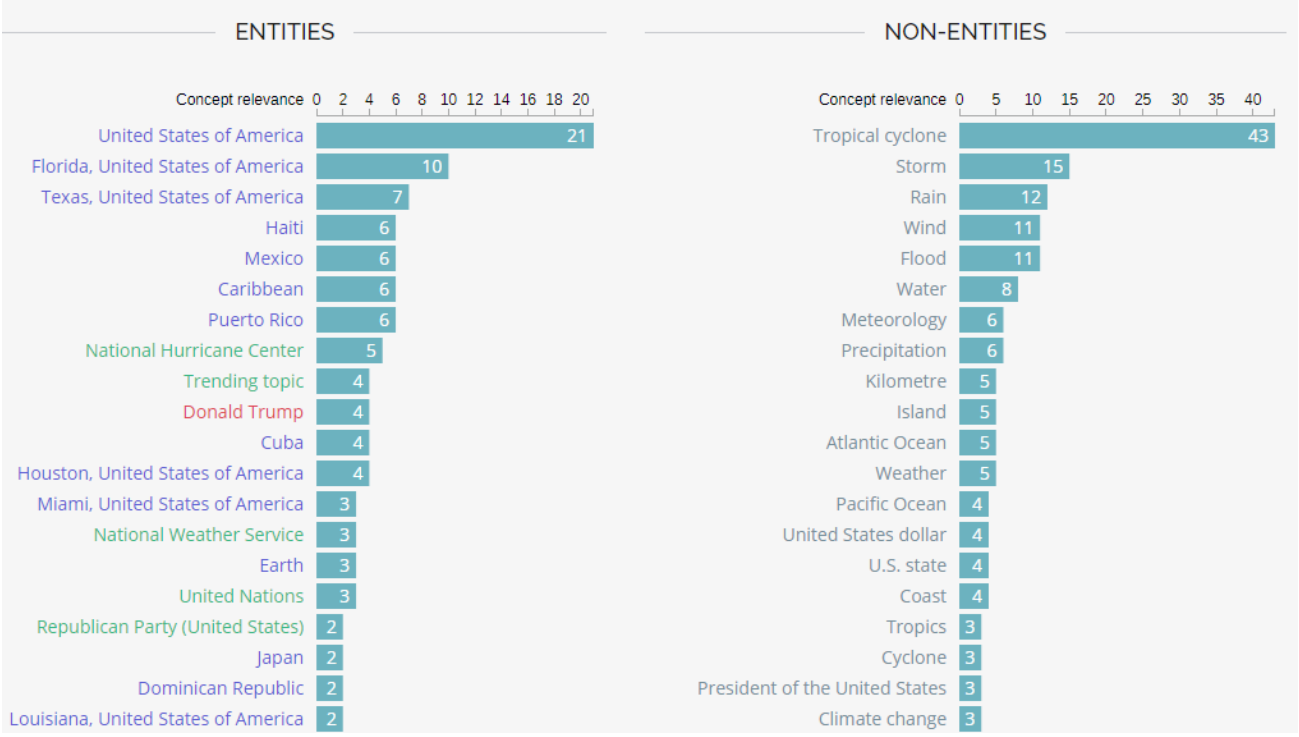


Figure 1: Summary of top concepts in events about hurricanes.

5 Conclusions and future work

We have presented a practical and efficient approach to wikification that requires no external data except the Wikipedia itself, can deal with documents in any language for which the Wikipedia is available, and is suitable for a high-performance, parallelized implementation.

The approach presented here could be improved along several directions. One significant weakness of the current approach concerns the treatment of minority languages. When dealing with a document in a certain language, we need hyperlinks whose anchor text is in the same language if we are to identify mentions in this input document. Thus, if the document is in a language for which the Wikipedia is not available, it cannot be wikified using this approach; and similarly, if the Wikipedia is available in this language but is small, with a small amount of text, low number of pages, and generally poor coverage, the performance of wikification will be low. One idea to alleviate this problem is to optionally allow a second stage of processing, in which Wikipedias in languages other than the language of the input document would also be used to identify mentions and provide candidate annotations. This might particularly improve the coverage of concepts that are referred to by the same words or phrases across multiple languages, as is the case with some types of named entities. For the purposes of pagerank-based disambiguation in this second stage, a large common link-graph would have to be constructed by merging the link-graphs of the Wikipedias for different languages. This can be done by using the cross-language links which are available in the WikiData ontology, providing information about when different pages in different languages refer to the same concept.

Another interesting direction for further work would be to incorporate local disambiguation techniques as a way to augment the current global disambiguation approach. When evaluating whether a mention a in the input document refers to a particular concept c, the local approach would focus on comparing the context of a to either the text of the Wikipedia page for c, or to the context in which hyperlinks to c occur within the Wikipedia. Preliminary steps taken in this direction in Sec. 2.5 did not lead to improvements in performance, but this subject is worth exploring further. Instead of the bag-of-words representation of contexts, other vector representations of words could be used, e.g. word2vec [6].

6 Acknowledgement

This work was supported by the Slovenian Research Agency as well as the euBusinessGraph (ICT-732003-IA), EW-Shopp (ICT-732590-IA) and RENOIR (H2020-MSCA-RISE-691152) projects.

7 References

[1] L. Zhang, A. Rettinger. *Final ontological word-sense-disambiguation prototype.* Deliverable D3.2.3, xLike Project, October 2014.

[2] J. Hoffart, M. A. Yosef, I. Bordino, *et al.* Robust disambiguation of named entities in text. *Proc. of the 2011 Conf. on Empirical Methods in Natural Language Processing*, Edinburgh, Scotland, 2011, pp. 782–792.

[3] L. Page, S. Brin, R. Motwani, T. Winograd. *The PageRank citation ranking: Bringing order to the web*. Digital Libraries Project Report SIDL-WP-1999-0120, Stanford University, 1998.
[4] M. Trampuš, B. Novak. Internals of an aggregated web news feed. *Proc. SiKDD 2012*.
[5] G. Leban, B. Fortuna, J. Brank, M. Grobelnik. Event registry: Learning about world events from news. *Proc. of the 23rd Int. Conf. on the World Wide Web* (WWW 2014), pp 107–110.
[6] T. Mikolov, K. Chen, G. Corrado, J. Dean. *Efficient estimation of word representations in vector space*. Arxiv.org, 1301.3781 [cs.CL], 2013.
[7] D. Milne, I. H. Witten. An open-source toolkit for mining Wikipedia. *Artificial Intelligence*, 194:222–239 (January 2013).
[8] A. Moro, A. Raganato, R. Navigli. Entity linking meets word sense disambiguation: A unified approach. *Trans. of the Assoc. for Comp. Linguistics*, 2:231–234 (2014).
[9] L. Ratinov, D. Roth, D. Downey, M. Anderson. Local and global algorithms for disambiguation to Wikipedia. *Proc. of the 49th Annual Meeting of the Assoc. for Comp Linguistics: Human Language Technologies* (2011), pp. 1375–84.
[10] J. Daiber, M. Jakob, C. Hokamp, P. N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. *Proc. of the 9th Int. Conf. on Semantic Systems*, 2013.

Continuous Blood Pressure Estimation from PPG Signal

Gašper Slapničar and Mitja Luštrek
Joef Stefan Institute, Jamova cesta 39, 1000 Ljubljana
E-mail: gasper.slapnicar@ijs.si, mitja.lustrek@ijs.si

Matej Marinko
Faculty of Mathematics and Physics, Jadranska cesta 19, 1000 Ljubljana
E-mail: matejmarinko123@gmail.com

Keywords: photoplethysmography, blood pressure estimation, regression analysis, m-health

Received: November 11, 2017

Given the importance of blood pressure (BP) as a direct indicator of hypertension, regular monitoring is encouraged for healthy people and mandatory for patients at risk from cardiovascular diseases. We propose a system in which photoplethysmogram (PPG) is used to continuously estimate BP. A PPG sensor can be easily embedded in a modern wearable device, which can be used in such an approach. The PPG signal is first preprocessed in order to remove major noise and movement artefacts present in the signal. A set of features describing the PPG signal on a per-cycle basis is then computed to be used in regression models. The predictive performance of the models is improved by first using the RReliefF algorithm to select a subset of relevant features. Afterwards, personalization of the models is considered to further improve the performance. The approach is validated using two distinct datasets, one from a hospital environment and the other collected during every-day activities. Using the MIMIC hospital dataset, the best achieved mean absolute errors (MAE) in a leave-one-subject-out (LOSO) experiment were 4.47 mmHg for systolic and 2.02 mmHg for diastolic BP, at maximum personalization. For everyday-life dataset, the lowest errors in the same LOSO experiment were 8.57 mmHg for systolic and 4.42 mmHg for diastolic BP, again using maximum personalization. The best performing algorithm was an ensemble of regression trees.

Povzetek: Krvni tlak je neposreden pokazatelj hipertenzije. Razvili smo sistem, ki krvni tlak ocenjuje iz fotopletizmograma (PPG), kakršen je že vgrajen v večino modernih senzorskih zapestnic. Signal PPG smo sprva predprocesirali in segmentirali na cikle. Predprocesiranje odpravi večino šuma, ki se pogosto pojavlja zaradi gibanja. Iz očiščenega signala smo nato izračunali množico značilk, ki smo jih uporabili v regresijskih modelih. Sistem smo izboljšali z uporabo algoritma RReliefF za izbor relevantnih značilk in z uporabo dela podatkov vsake osebe za učenje personaliziranih napovednih modelov. Sistem smo vrednotili na dveh podatkovnih množicah, eni iz kliničnega okolja in drugi zbrani med rutinskimi dnevnimi aktivnostmi posameznikov. V poizkusu smo model vsakič naučili na vseh osebah razen eni in ga nato testirali na izpuščeni osebi. Z uporabo klinične podatkovne množice smo v omenjenem poizkusu dosegli najnižji povprečni absolutni napaki (MAE) 4.47 mmHg za sistolični in 2.02 mmHg za diastolični krvni tlak, pri največji stopnji personalizacije. Za množico, zbrano med dnevnimi aktivnostmi, smo dosegli najnižji napaki 8.57 mmHg za sistolični in 4.42 mmHg za diastolični krvni tlak, ponovno pri največji stopnji personalizacije. Najbolje se je obnesel ansambel regresijskih dreves.

1 Introduction

World Health Organization (WHO) listed cardiovascular diseases as the most common cause of death in 2015, responsible for almost 15 million deaths combined [1]. Hypertension is one of the most common precursors of such diseases and can be easily detected with regular blood pressure (BP) monitoring, which is especially critical for patients already suffering from hypertension or related cardiovascular diseases, as it can indicate potential vital threats to their health.

While regular BP monitoring is important, it is also troublesome, as devices using inflatable cuffs are still considered the "golden standard". The cuff placement is critical, as the sensor must be located directly above the main artery in the upper arm area, at approximately heart height [4]. These requirements impose relatively strict movement restrictions on the subject and require substantial time commitment, thus causing low subject adherence to regular monitoring. Furthermore, when done by the subject him/herself in a home environment, this process can cause stress, which in turn influences the BP values, making the measurements less reliable. This problem is usually not alleviated by having the medical personnel perform the measurement, as this can again cause anxiety in the subject, commonly known as the "white coat syndrome".

Our work focuses on photoplethysmogram (PPG) analysis and the development of a robust non-obtrusive method for continuous BP estimation. It will be implemented and used in an m-health system based on a wristband with an embedded PPG sensor. This will allow the user to wear the device without any interference or limits imposed upon their daily routine, allowing for truly continuous measuring without stressing the user and thus potentially influencing the BP values.

The rest of the paper is organized as follows. Section 2 gives a brief overview of the related work. Section 3 explains the methodology we have used, focusing on signal pre-processing and machine learning features. Section 4 elaborates on the experimental setup and results, and Section 5 concludes with a summary and plans for future work.

2 Related work

Photoplethysmography is a relatively simple technique based on inexpensive technology, which is becoming increasingly popular in wearable devices for heart rate estimation. It is based on the illumination of the skin and measurement of changes in its light absorption [5]. In its basic form it only requires a light source to illuminate the skin (typically a light-emitting diode – LED light) and a photodetector (photodiode) to measure the amount of light either transmitted through, or reflected from the skin. Thus PPG can be measured in either transmission or reflectance mode. Both modes of operation are shown in Figure 1.

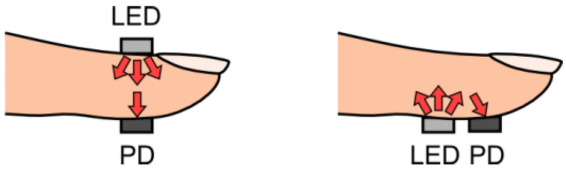


Figure 1: Transmission and reflectance mode in which the PPG signal can be obtained. LED is the light source while PD is the photodetector [6].

With each cardiac cycle, the heart pumps blood towards the periphery of the body. This produces a periodic change in the amount of light that is either absorbed or reflected from the skin to the photodetector, as the tissue changes its tone based on the amount of blood in it.

Exploring the recent applications of PPG, we can see that it is becoming more widely used in BP estimation. One of two common approaches are typically used:

1. BP estimation using two sensors (PPG + Electrocardiogram (ECG))
2. BP estimation using the PPG sensor only

The first approach requires the use of two sensors, typically an ECG and a PPG sensor, in order to measure the time it takes for a single heart pulse to travel from the heart to a peripheral point in the body. This time is commonly known as pulse transit time (PTT) or pulse arrival time (PAT), and its correlation with BP changes is well established.

The more recent approach is focused on the PPG signal only; however, the relationship between the PPG and BP is only postulated and not as well established as the relationship between the PTT and BP. This approach is, however, notably less obtrusive, especially since PPG sensors have recently become very common in most modern wristbands.

BP is commonly measured in millimeters of mercury (mmHg), which is a manometric unit routinely used in medicine and many other scientific fields. A mercury manometer is a curved tube containing mercury, which is closed at one end while pressure is applied on the other end. 1 mmHg of pressure means that the pressure is large enough to increase the height of the mercury in the tube for 1 mm. To put the values discussed in this paper into perspective, the normal healthy adult BP is considered to be around 120 mmHg (16 kPa) for systolic and 80 mmHg (11 kPa) for diastolic BP [2].

One of the earliest PPG-only attempts was conducted by Teng et. al. in 2003 [3]. The relationship between the arterial BP (ABP) and certain features of the PPG signals was analyzed. Data were obtained from 15 young healthy subjects in a highly controlled laboratory environment, ensuring constant temperature, no movement and silence. The mean differences between the linear regression estimations and the measured BP were 0.21 mmHg for systolic (SBP) and 0.02 mmHg for diastolic BP (DBP). The corresponding standard deviations were 7.32 mmHg for SBP and 4.39 mmHg. Using mean errors instead of mean absolute errors as the evaluation metric is questionable, since it does not reflect the actual performance of the derived model and the error can be extremely low, even if the actual predictions are high above and under the actual observed BP values.

A paper was published in 2013 in which the authors used data from the Multiparameter Intelligent Monitoring in Intensive Care (MIMIC) waveform database [7, 8] to extract 21 time domain features and use them as an input vector for artificial neural networks (ANNs) [9]. The results are not quite as good as with the linear regression model described earlier; however, the data was obtained from a higher number and variety of patients in a less controlled environment. Mean absolute errors of less than 5 mmHg for both SBP and DBP were reported. While the environment was less controlled compared to the previous work, the patients were still within a hospital setting and hospital equipment was used for data collection. Furthermore, only an undisclosed subset of all the available data from MIMIC was used.

Another research was conducted in 2013 in which the authors used a smartphone camera to capture the PPG signal using the camera flash as the light source and the phone camera as the photodiode [10]. PPG features were again extracted and fed to a neural network, which estimated SBP

and DBP. All the data processing and BP evaluation was done in a cloud in order to reduce the computational burden on the device. It is not clear how many subjects participated in the experiment, however, they reported the maximum error not exceeding 12 mmHg. The error metric is not explained in detail, however, based on the given results table, we can presume that MAE was used. Such a method requires some user effort, as the user must place and hold his finger over the camera and LED light. This prevents any other activities during this time.

It is clear that the PPG-only approach has potential, however, a robust unobtrusive method that works well on a general case is yet to be developed.

3 Methodology

The proposed system consists of two main modules, namely the signal pre-processing and machine learning module. The former is responsible for cleaning the PPG signal of most noise and then segmenting it into cycles, where one PPG cycle corresponds to a single heart beat. The latter extracts features describing the PPG signal on a per-cycle basis, selects a subset of relevant features using the RReliefF algorithm [12], and finally feeds the subset into regression algorithms, which build the prediction models.

3.1 Signal pre-processing

PPG sensors must be very sensitive in order to detect tiny variations in light absorption of the tissue. This also makes them highly susceptible to movement artefacts. This problem is especially obvious when dealing with PPG collected via a wristband, as the contact between the sensor and the skin can be compromised during arm movements. This is partially alleviated by using green light, which is less prone to artefacts, however, major artefacts often remain in the signal. Subsequently, substantial effort is directed towards PPG pre-processing.

3.1.1 Cleaning based on established medical criteria

In the first phase, both the BP and PPG signal are roughly cleaned based on established medical criteria [13]. A 5-second sliding window is used to detect segments with extreme BP values or extreme changes of the BP in a short time period. Thresholds for extreme values and changes are selected based on established medical criteria in related work [13] and are given in Table 1. Some thresholds were slightly modified, since the criteria given in the referenced paper seem too strict for some subjects encountered in our datasets. We have thus loosened the criteria in accordance with empirical observations in our datasets (e.g., the original criteria excludes all data with SBP > 180, while we observed some segments with SBP over 180 mmHg).

After the cleaning of the clinical dataset, 85% of data is kept on average, while 15% is discarded. This is very subject dependent, as for some subjects nearly all the data is removed (e.g., sensor anomaly which shows 0 ABP almost all the time), while for majority of subject most of the data is kept. For everyday-life dataset, which contains a lot more noise, only 40% of data is kept, while 60% is discarded. This is the result of some subjects having long noisy segments of the PPG signal. It should be noted, that these percentages are also subject of the parameters for trade-off between the required quality and the amount of signal kept, which are discussed in 3.1.3.

Criterion	Threshold
SBP	> 250 or < 80
DBP	> 150 or < 40
SBP – DBP	< 20
ΔSBP or ΔDBP in 5 sec	> 50

Table 1: Established medical criteria and thresholds for rough signal cleaning. Δ signifies a change in BP value. 5-second segments meeting any of these criteria are removed from the signal.

3.1.2 Peak and cycle detection

In order to do further cleaning and subsequent feature extraction, PPG cycle detection is mandatory. This is not trivial, as substantial noise in the PPG signal poses a significant problem, as mentioned earlier.

This problem was tackled in several steps. First, a filtering transformation, which enhances the systolic upslopes of the pulses in the PPG signal, is used. It is designed to use the derivative of the PPG signal at lower frequencies, in order to detect the abrupt upslopes of the systolic pulse compared to the diastolic or dicrotic pulse in the PPG signal. This is based on a low-pass differentiator (LPD) filter, which removes high frequency components and performs differentiation. Once the steepest points in the PPG signal are located, the following peak is chosen as the PPG systolic peak. Afterwards, a time-varying threshold for peak detection is applied, which ensures that potential double peaks or diastolic peaks close to the systolic ones are not chosen. The procedure is explained in detail in a paper by Lzaro et al. [14].

After the peaks are detected, finding the cycle start-end points is simpler, as the dominant valleys between the detected peaks must be found. An example of detected peaks and cycle locations using the described method is shown in Figure 2.

3.1.3 Cleaning based on ideal templates

After cycles are successfully detected, the second cleaning phase begins. A 30-second sliding window is used.

First, the most likely length of a cycle L in the current window is determined using autocorrelation analysis. A copy of the PPG signal in the current window is taken and

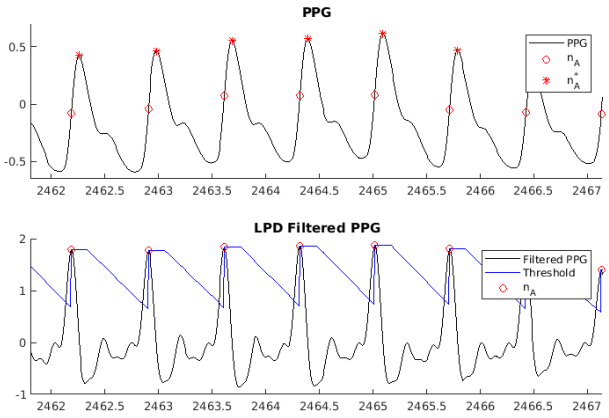


Figure 2: The upper subplot shows a PPG segment. Lower subplot shows the LPD filtering transformation of the same PPG segment. Peaks of the transformation in the lower subplot correspond to the steepest systolic upslopes of the PPG in the upper subplot, and are denoted as n_A. Actual detected PPG peaks are denoted as n_A^*.

shifted sample by sample up to a certain length that contains at least two heart beats. When the copy is shifted by the number of samples corresponding to exactly one cycle, the autocorrelation reaches its first peak, and this number of samples is chosen as L.

Presuming that the majority of cycles within a 30-second window are not morphologically altered, we can create an "ideal cycle template" for this window. Such a template is created by always taking the next L samples from each cycle starting point and then computing the mean cycle. Each individual cycle is then compared to the computed template and its quality is evaluated using three signal quality indices (SQIs), which are defined as follows [15]:

1. **SQI1**: First L samples of each cycle are taken and then each cycle is directly compared to the template using a correlation coefficient.

2. **SQI2**: Each cycle is interpolated to length L and then the correlation coefficient with the template is computed.

3. **SQI3**: The distance between each cycle and the template is computed using dynamic time warping (DTW).

Finally the thresholds for each SQI are empirically determined. Each cycles' SQIs are evaluated and if they reach the required quality threshold, that cycle is kept, otherwise it is removed. If more than half the cycles in the current 30-second window are under the thresholds, the whole window is discarded as too noisy. An example of this cleaning is shown in Figure 3.

Once the PPG signal is cleaned and only high-quality cycles with minimal morphological anomalies remain, features can be extracted from each cycle.

3.2 Machine learning

In order to derive the relationship between the PPG and BP, features describing the PPG signal were computed and then the relevant subset of these features was selected to be used in the regression algorithms.

3.2.1 Features

In accordance with the related work [3, 9, 10], several time-domain features were computed from the PPG signal, and the set of features was further expanded with some from the frequency [13] and complexity-analysis domains. Most features focus on describing the morphology of a given PPG cycle, as shown in Figure 4.

Feature	Description
Tc	Cycle duration
Ts	Time from start of cycle to systolic peak
Td	Time from systolic peak to end of cycle
Tnt	Time from systolic peak to diastolic rise
Ttn	Time from diastolic rise to end of cycle
S1	Area under the curve (AUC) from start of cycle to max upslope point
S2	AUC from max upslope point to systolic peak
S3	AUC from systolic peak to diastolic rise
S4	AUC from diastolic rise to end of cycle
AUC syst	S1 + S2
AAC syst	Area above the curve (AAC) from start of cycle to systolic peak
AUC diast	S3 + S4
AAC diast	AAC from systolic peak to end of cycle

Table 2: Elaborations of some of the used features shown in Figure 4.

In addition to the features focusing on the PPG cycle morphology, which were highlighted thus far, the following features were computed and considered:

1. *AI – Augmentation Index*: a measure of wave reflection on arteries.

$$AI = \frac{\textit{diastolic rise amplitude}}{\textit{systolic peak amplitude}}$$

2. *LASI – Large Artery Stiffness Index*: an indicator of arterial stiffness, which is denoted as *Tnt* in Figure 4 and Table 2.

3. *Complexity analysis*: signal complexity and mobility are computed for the 30-second PPG segment containing the current cycle. Mobility represents an estimate of the mean frequency and is proportional to the standard deviation of the power spectrum. Complexity gives an estimate of change in frequency by comparing the signal similarity to a pure sine wave. They are

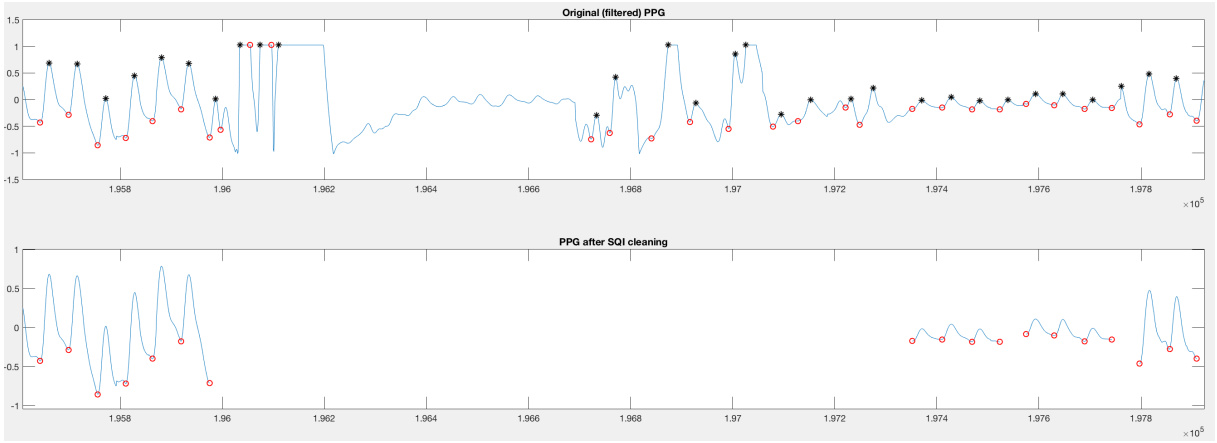


Figure 3: Example of the cleaning algorithm in the second phase of the signal pre-processing. Comparing the top (uncleaned) and bottom (cleaned) PPG signal, we see that the obvious artefact segments are removed.

given by Najarian and Splinter [11] as follows (presuming a zero-mean signal):

$$S_0 = \sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}},$$

$$S_1 = \sqrt{\frac{\sum_{j=2}^{N-1} d_j^2}{N-1}},$$

$$S_2 = \sqrt{\frac{\sum_{k=3}^{N-2} g_k^2}{N-2}},$$

where x is the PPG signal, d is the first order derivative of x and g is the second order derivative of x.

$$Mobility = \sqrt{\frac{var(d)}{var(x)}} = \frac{S_1}{S_0}$$

$$Complexity = \frac{mobility(d)}{mobility(x)} = \sqrt{\frac{S_2^2}{S_1^2} - \frac{S_1^2}{S_0^2}},$$

4. *FFT features*: amplitudes and phases of the frequency-domain representation of the 30 second PPG segment containing the current cycle. The length of the window was chosen such that it contains enough cycles (expected 1 cycle per second) for the frequencies in the segment to be reliably determined.

Considering all the time and frequency-domain features along with the complexity-analysis features, and the amount of instances (cycles) available, we are often dealing with a very large matrix of training data. The number of rows (instances) is on the order of magnitude 10^5 and the number of columns (features) is on the order of magnitude 10^2, thus dimensionality reduction through selection of a subset of relevant features is feasible, but not mandatory. More importantly, feature selection allows us to determine which features are useful for the learning process, and which are irrelevant, allowing us to obtain a smaller subset containing only the relevant features.

3.2.2 Feature selection

The RReliefF algorithm was chosen for feature selection. It is a modification of the ReliefF algorithm, suitable for regression problems with continuous target variables. The algorithm was applied to a subset of 10% of all data chosen randomly. This was repeated 10 times. All the features with non-zero relevance, as chosen by the algorithm, were considered in each iteration and their importance was saved. Looking at the final scores of the algorithm across all the iterations, we notice that quite a few features are considered irrelevant, while the same features are commonly chosen as important for both SBP and DBP, as shown in Figure 5. Noting the fact that the same features were selected in each of the 10 iterations, we can assume that the relevant features are not dependent on the selected subset of the available data.

As mentioned, all the features with non-zero importance were taken, as more than half were discarded as irrelevant by the RReliefF algorithm. Among the non-zero importance features, some features from each of the groups mentioned earlier (temporal, frequency and complexity analysis) are present. Most area-based features were marked as irrelevant, while certain times (Tc, Ts and Td), both complexity-analysis (signal complexity and signal mobility) as well as some frequency-domain (amplitudes and phases at low frequencies) features were marked as important. These non-zero importance features were then used in the regression algorithms.

The relevant features were determined using the larger and more varied dataset from the MIMIC database. The same subset of features was also used with the smaller everyday-life dataset. Both datasets are described in more detail in the following section.

Since the feature selection procedure only slightly improved the results, we have not considered experiments with other or additional feature selection methods.

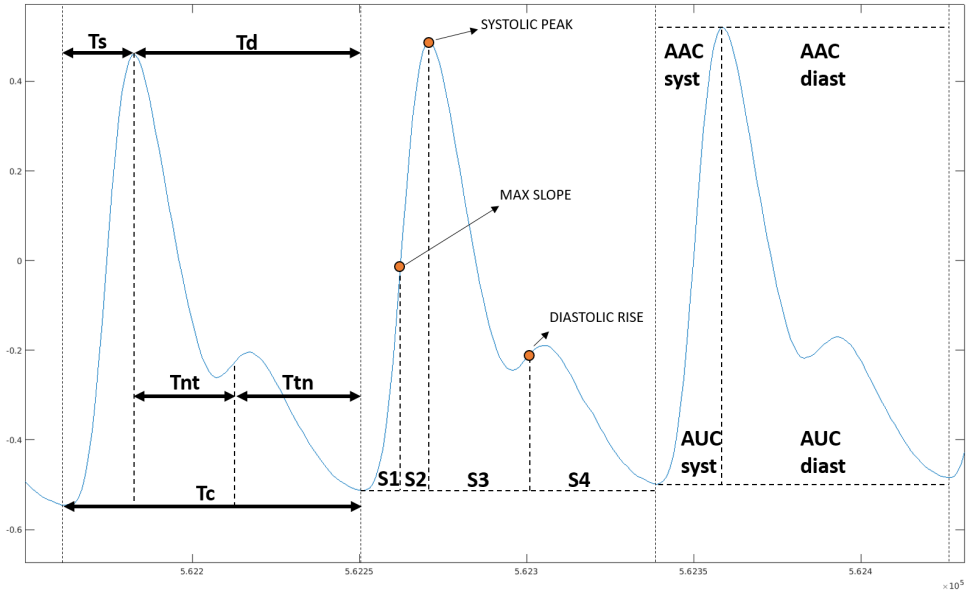


Figure 4: Time and area based features that describe the morphology of the PPG signal on a per-cycle basis. The features are listed and elaborated in Table 2.

4 Experiments and results

In an effort to make the proposed method as general as possible, two datasets were considered for the experimental evaluation. The data from all subjects, which met the requirements of having both the PPG and BP signals recorded, were always used in the experiments.

4.1 Data

The first dataset is from the publicly accessible MIMIC database, which is commonly used for experiments and competitions in the signal processing field. The original version contains data from 72 hospitalized patients. All patients with both the PPG and BP signal were originally considered, however, after the filtering and pre-processing, only 41 patients had enough high-quality data remaining to be used in the experiments. All the data was collected in a hospital environment using hospital measuring equipment, including an ABP measuring device. The ABP is measured by inserting a cathether in an artery, making it highly invasive, however, it offers the most precise BP monitoring.

The second dataset was collected at Jožef Stefan Institute (JSI) using the Empatica E4 wristband for the PPG and a digital cuff-based Omron BP monitoring device for the ground truth BP, as is common in such experimental settings in related work. This device is reported to be clinically validated according to the British Hypertension Society and the Association for the Advancement of Medical Instrumentation (AAMI) protocols [17], which means that the mean errors do not exceede 5 mmHg. The collection procedure at JSI was conducted in accordance with the standardized clinical protocol. The correct placement of the cuff on the upper arm area with the sensor above the main artery was ensured. The measurements were done in an upright sitting position, making sure the cuff was located at approximately heart height. The recommended protocol was followed as best as possible, however, in an ideal situation the ground truth BP should be measured as ABP within an artery. Due to the invasive nature of ABP measurement, this is not feasible in an everyday-life situation, so the digital cuff-based monitor was used as a good replacement. An upper-arm cuff-based monitor was chosen over a wrist-based one, as the latter is less accurate and extremely sensitive to body position.

In the first completed phase of the data collection, 8 healthy subjects were considered, 5 male and 3 female. Each subject wore the wristband PPG measuring device for several hours during their everyday activities. They measured their BP every 30 minutes or more often. Finally, only parts of the PPG signal 3 minutes before and after each BP measurement point were taken into consideration, as the measured BP value is only relevant for a short time. Ideally, the BP would be measured more often, however, this would place further stress on the subjects and was not possible during their everyday routine. Furthermore, additional physiological variations (e.g., breathing rate) could be obtained from the PPG and used for the BP estimation, however, this was not yet considered but might be a subject of future work.

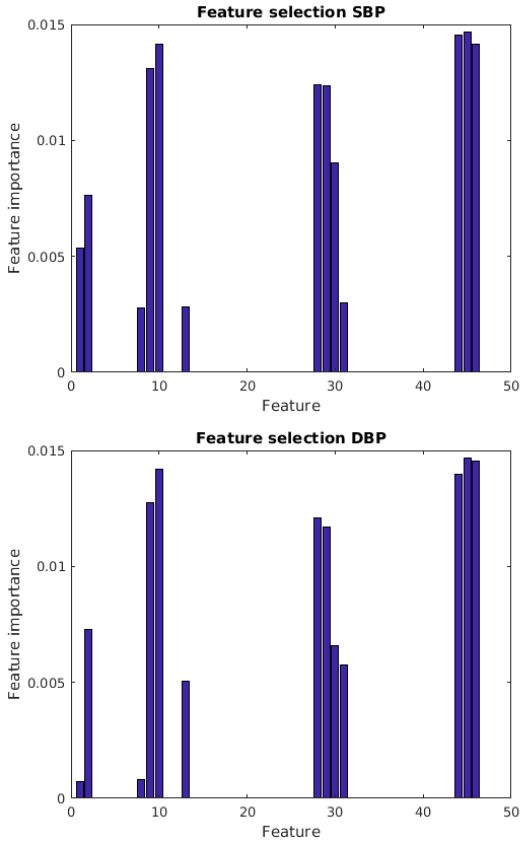


Figure 5: The output of RReliefF algorithm, which shows the feature importance for each of the considered features.

4.2 Experimental setup

Two experimental setups were considered, 5-fold cross validation and LOSO. The purpose of the first was to establish initial observations about the selected features and performance of different regression algorithms. The second experiment was conducted to evaluate the generalization performance of the algorithms and subsequently determine potential requirement for personalization.

4.2.1 K-fold cross validation

The MIMIC dataset consisted of roughly 200 000 instances post filtering, which correspond to 41 patients. The instances were obtained by uniformly taking 20 3-minute segments from the whole recording for a given patient. Each instance (cycle) in a given 3-minute segment was assigned the mean SBP and mean DBP of this segment. This simulates the patients measuring their BP periodically, but not more than once in 3 minutes.

K-fold cross validation ($k = 5$) was conducted, where instances were first shuffled randomly and then all the data was split into nearly equal folds. Then $k - 1$ folds were taken for learning and the remaining fold was used for testing. This was repeated k times. The random shuffling of instances makes it so that instances belonging to a given subject might appear in both training and testing sets. This was taken into account (a sort of implicit personalization), as this experiment was merely a starting point to determine the initial performance of the algorithms and was later complemented by a Leave-one-subject-out experiment.

Several regression algorithms were compared using the full set of features. The algorithm that performed best using all the features was additionally evaluated using only the subset of best features as selected by the RReliefF algorithm. The predictive performance of these options in 5-fold cross validation is discussed in detail in the Results section.

4.2.2 Leave-one-subject-out

Due to increased computational complexity of a leave-one-subject-out experiment compared to k-fold cross-validation, data was additionally sub-sampled, by taking 500 uniformly selected cycles from each patient's data.

During the initial attempt, a regression model was trained in each iteration on all the subjects, except the one left out. It was ensured that no instances from the testing subject appeared in the training set. This yielded poor results. Notable improvements can be made by using a small amount of each patient's data for training, most likely due to each patient having a subtly unique cardiovascular dynamic and relation between PPG and BP. This was additionally confirmed by doing cycle morphology analysis, during which it was established that similar cycle shapes do not necessarily signify similar BP values. Due to the mentioned factors, personalization of the trained models was considered in an attempt to improve the predictive performance of the models.

In the second attempt, the regression models were again trained using all the subjects except the one left out. This time, however, the models were further personalized by using some instances from the left out subject. The instances of the left out subject were grouped by their BP values. These groups were then sorted from lowest to highest BP. Afterwards, every n-th group ($n = 2, 3, 4, 5, 6$) of instances was taken from the testing data and used in training in order to personalize the model to the current patient. This ensures personalization with different BP values, as taking just a single group of instances gives little information, since the BP will be constant within this group. Given the fact that the MIMIC data consists of roughly 5x the number of patients compared to everyday-life data, the personalization data for it was multiplied 5 times, making it noticeable within the large amount of training data from the remaining patients.

During both attempts, several regression algorithms were once again considered, as given in Tables 3 and 4. The MAE was used as the evaluation metric. All models were compared with a dummy regressor, which always predicted the mean BP value of the same combination of general and personalization data as the other models used for training. Finally, the regressor with the lowest MAE was chosen as

best.

For successful personalization, the user should measure their PPG continuously and also make a few periodic measurements of their BP using a reliable commercial device. This allows the model to personalize to the user, learning from a small sample of their labeled data, thus improving its predictive performance.

4.3 Results

Using the personalization approach, notable improvements have been made over the dummy regressor in both experiments. The results are discussed in detail in the following sections.

4.3.1 K-fold cross validation results

MAE with corresponding standard deviations in the 5-fold cross validation experiment for the MIMIC data are given in Table 3, while the results for the everyday-life data are given in Table 4.

Algorithm	**MAE_{SBP} [mmHg]**
Dummy (predicts mean)	19.70 ± 16.07
Linear regression	18.47 ± 15.91
Ensemble (all feat.)	5.83 ± 7.74
Ensemble (relevant feat.)	4.90 ± 6.59
Algorithm	**MAE_{DBP} [mmHg]**
Dummy (predicts mean)	8.73 ± 6.77
Linear regression	8.14 ± 7.98
Ensemble (all feat.)	2.92 ± 4.09
Ensemble (relevant feat.)	2.21 ± 3.70

Table 3: MAE of different algorithms for SBP and DBP estimation in 5-fold cross validation using the MIMIC hospital dataset.

Algorithm	**MAE_{SBP} [mmHg]**
Dummy (predicts mean)	11.46 ± 7.51
Linear regression	11.21 ± 8.00
Ensemble (all feat.)	9.12 ± 7.90
Ensemble (relevant feat.)	7.87 ± 7.47
Algorithm	**MAE_{DBP} [mmHg]**
Dummy (predicts mean)	5.01 ± 3.99
Linear regression	5.01 ± 8.00
Ensemble (all feat.)	4.38 ± 3.74
Ensemble (relevant feat.)	3.84 ± 3.63

Table 4: Mean absolute errors of different algorithms for SBP and DBP estimation in 5-fold cross validation using the JSI-collected everyday-life dataset.

Ensemble of shallow regression trees has shown the best predictive performance in the 5-fold cross validation for both SBP and DBP using both datasets. We also notice a slightly better performance when only the relevant features, as given by RReliefF, are used in comparison to the default feature set.

As the ensemble of regression trees has shown the best performance, its hyperparameters were optimized using Bayesian optimization. All the available hyperparameters were optimized using the MATLAB built-in Bayesian Optimization Workflow [16]. It optimizes both the hyperparameters of the ensemble as well as the hyperparameters of the weak learners, which are chosen to be shallow Regression Trees. The optimization is ran for 30 iterations, trying to minimize the objective cross-validation loss function. Bootstrap aggregation was chosen as superior over gradient boosting strategy, and the optimal number of weak learners was determined to be 77. The maximum number of splits in the weak learner was determined to be 1, meaning that the regression trees are in fact regression stumps.

4.3.2 Leave-one-subject-out results

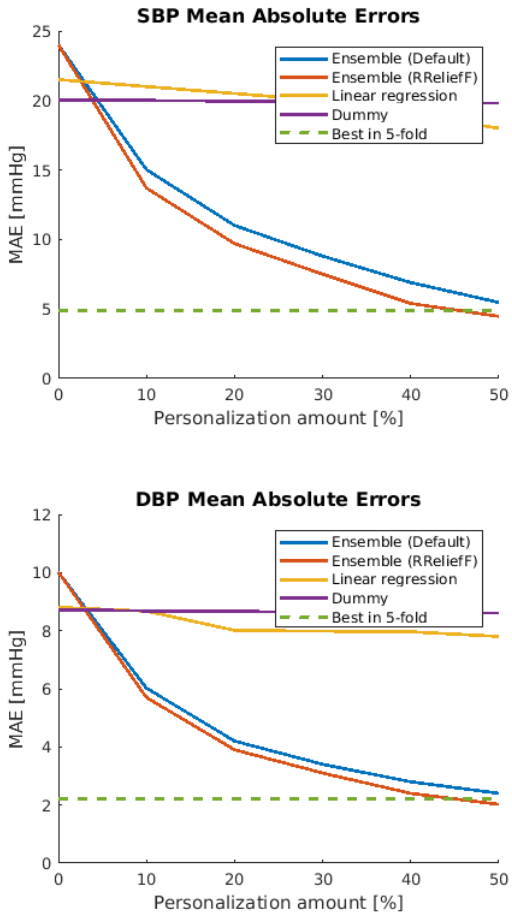


Figure 6: MAE for SBP and DBP for the MIMIC dataset, at different amounts of personalization.

The lowest error using the MIMIC data was again achieved using the hyperparameter tuned Ensemble of regression trees algorithm with RReliefF selected subset of features. The highest amount of personalization (50%) gave the best results. 50% personalization corresponds to 10 BP measurements conducted by the subject, given the fact that 20 segments with 20 different BP values were taken. Obtaining 10 BP measurements by the subject, in order to personalize the model, seems like a reasonable requirement.

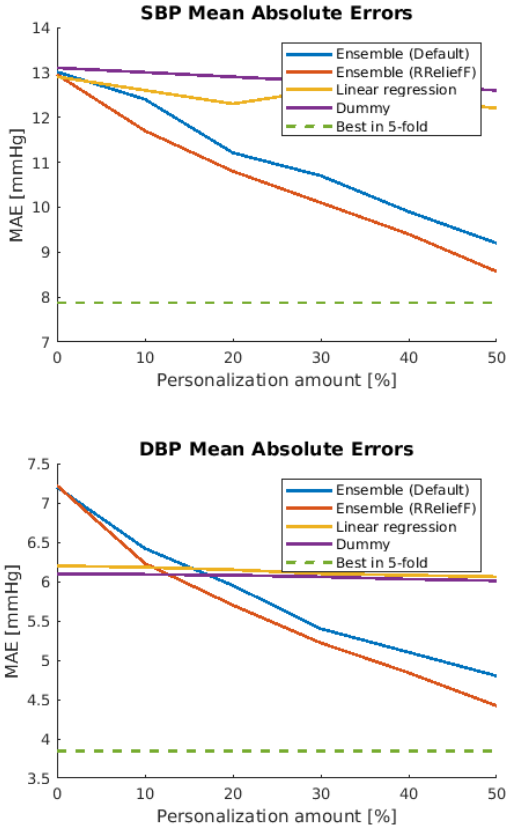


Figure 7: MAE for SBP and DBP for the everyday-life dataset, at different amounts of personalization.

The JSI-collected everyday-life data has proven to be more problematic, as there were only a few different BP values recorded in the first phase of data collection. Furthermore, due to the high amount of movement artefacts, a lot of data was removed by the cleaning algorithm, leaving a very small amount of usable data with a very low variation in BP. This further enhanced the performance of the dummy regressor, which achieved much lower MAE compared to the MIMIC dataset, however, improvements were again achieved by using personalization, as shown in Figure 7.

5 Conclusion

We have developed a system for BP estimation using only the PPG signal, and have evaluated its performance on two distinct datasets using two experimental setups.

The first module of the system deals with signal preprocessing, removing most movement artefacts and anomalies from the PPG signal. It then detects PPG cycles corresponding to heart beats and feeds them to the second module, which computes a number of features describing each cycle. This is followed by feature subset selection using the RReliefF algorithm and finally the features are fed into several regression algorithms. Predictive models were created and evaluated on a hospital MIMIC dataset as well as an everyday-life dataset collected at JSI. The lowest MAE achieved for the MIMIC hospital dataset in 5-fold cross validation were 4.90 $\pm$ 6.59 mmHg for SBP and 2.21 $\pm$ 3.70 mmHg for DBP. The best performing algorithm was an Ensemble of shallow regression trees. Its hyperparameters were optimized using Bayesian optimization. Finally, the same models were evaluated on the same dataset using the leave-one-subject-out validation, achieving the lowest MAE of 4.47 $\pm$ 5.85 mmHg for SBP and 2.02 $\pm$ 2.94 mmHg for DBP, again using the same hyperparameter-tuned Ensemble and the subset of features selected by the RReliefF algorithm. These results were achieved using the maximum, 50% personalization. Similar trends can be observed for the everyday-life JSI-collected dataset. The lowest MAE in 5-fold cross validation were 7.87 $\pm$ 7.47 mmHg for SBP and 3.84 $\pm$ 3.63 mmHg for DBP. Ensemble of shallow regression trees with optimized parameters prevailed again. In LOSO validation, the lowest MAE of of 8.57 $\pm$ 7.93 mmHg for SBP and 4.42 $\pm$ 3.61 mmHg for DBP were achieved.

5.1 Interpretation of results

Comparing the results of the 5-fold cross-validation to those of the LOSO evaluation, we first notice, that the best performing algorithm is the same. In each fold in the 5-fold cross validation, 80% of randomly shuffled instances were taken for training, which translates to 80% personalization for each subject. This is the reason behind the lower MAE in the 5-fold cross-validation, however, similar MAE was also achieved with higher amounts of personalization in the LOSO experiment. The developed system shows promising results and could be used by both regular people and hypertensive patients during their everyday routine, by wearing an unobtrusive wristband. It could inform them of their current medical condition regarding BP. Further testing with more field-collected data is required to more accurately determine its performance, however, it already achieves low MAE when personalization is considered.

5.2 Future work

We plan to expand our data collection experiment at JSI, which will give us more data and more variety within the collected BP data. Once enough data is collected, we plan to upgrade the machine learning part of our pipeline using deep-learning algorithms. These are well-suited for problems dealing with signal analysis and represent the state of the art approach in signal processing in recent years, making them a suitable candidate for our domain.

Acknowledgement

The HeartMan project has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 689660. Project partners are Jožef Stefan Institute, Sapienza University, Ghent University, National Research Council, ATOS Spain SA, SenLab, KU Leuven, MEGA Electronics Ltd and European Heart Network.

References

[1] The World Health Organization. “The top 10 causes of death”, 2015.

[2] Mayo Foundation for Medical Education and Research (MFMER). “Blood pressure chart: What your reading means”. Accessed online: 2nd March, 2018.

[3] Teng et. al. “Continuous and noninvasive estimation of arterial blood pressure using a photoplethysmographic approach”, *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, 2003.

[4] Frese et al. “Blood Pressure Measurement Guidelines for Physical Therapists”, *Cardiopulmonary Physical Therapy Journal*, 2011.

[5] Shelley et al. “Pulse Oximeter Waveform: Photoelectric Plethysmography”, *Clinical Monitoring: Practical applications for anesthesia and critical care*, 2001.

[6] Tamura et al. “Wearable Photoplethysmographic SensorsPast and Present”, *Electronics*, 2014.

[7] Goldberger et al. “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals”, *Circulation*, 2000.

[8] Moody et al. “A Database to Support Development and Evaluation of Intelligent Intensive Care Monitoring”, *Computers in Cardiology*, 1996.

[9] Lamonaca et al. “A neural network-based method for continuous blood pressure estimation from a PPG signal”, *IEEE International Congress I2MTC*, 2013.

[10] Lamonaca et al. “Application of the Artificial Neural Network for blood pressure evaluation with smartphones”, *2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)*, 2013.

[11] Najarian et al. “Biomedical Signal and Image Processing, 2nd Edition”, *CRC Press*, 2012.

[12] Robnik-Šikonja et al. “Theoretical and Empirical Analysis of ReliefF and RReliefF”, *Machine Learning*, 2003.

[13] Xing et al. “Optical Blood Pressure Estimation with Photoplethysmography and FFT-Based Neural Networks”, *Biomedical Optics Express*, 2016.

[14] Lzaro et al. “Pulse Rate Variability Analysis for Discrimination of Sleep-Apnea-Related Decreases in the Amplitude Fluctuations of Pulse Photoplethysmographic Signal in Children”, *IEEE Journal of Biomedical and Health Informatics*, 2014.

[15] Li et al. “Dynamic time warping and machine learning for signal quality assessment of pulsatile signals”, *Physiological Measurement*, 2012.

[16] The MathWorks Inc., Natick, Massachusetts, United States. “MATLAB 2017a Optimization Toolbox”, 2017.

[17] Coleman et al. “Validation of the Omron M7 (HEM-780-E) oscillometric blood pressure monitoring device according to the British Hypertension Society protocol”, *Blood Pressure Monitoring*, 2008.

Quantitative Score for Assessing the Quality of Feature Rankings

Ivica Slavkov, Matej Petković, Dragi Kocev and Sašo Džeroski
Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia
Jožef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia
E-mail: saso.dzeroski@ijs.si

Keywords: feature ranking, feature selection, evaluation methodology, high-dimensional data

Received: January 18, 2018

Feature ranking is a machine learning task that is related to estimating the relevance (importance) of individual features in a dataset. Relevance estimates can be used to induce an ordering of the features from a dataset, also called a feature ranking. In this paper, we consider the problem of the evaluation of different feature rankings. For that purpose, we propose an intuitive evaluation method, based on iterative construction of feature sets and their evaluation by learning predictive models. By plotting the obtained predictive performance of the models, we obtain error curves for each feature ranking. We then propose a scoring function to quantitatively assess the quality of the feature ranking. To evaluate the proposed method, we first define a synthetic setting in which we analyse the method and investigate its properties. By using the proposed method, we next perform an empirical comparison of several feature ranking methods on datasets from different domains. The results demonstrate that the proposed method is both appropriate and useful for comparing feature rankings of varying quality.

Povzetek: Rangiranje značilk je naloga strojnega učenja, povezana z ocenjevanjem pomembnosti značilk v podatkih. Značilke lahko nato uredimo glede na dobljene ocene in tako dobimo ureditev, ki ji prav tako pravimo rangiranje značilk. V tem delu obravnavamo problem evalvacije različnih metod za urejanje značilk. Predlagamo postopek, ki temelji na iterativni konstrukciji množic značilk ter njihovi evalvaciji s pomočjo napovednih modelov. Če dobljene ocene natančnosti modelov narišemo na graf, dobimo krivulje natančnosti za vsako rangiranje značilk. Te krivulje s predlaganim postopkom pretvorimo v kazalec, ki poda kakovost rangiranja številsko. Metodo najprej evalviramo na sintetičnih podatkih, nato pa jo preizkusimo še na realnih podatkih iz različnih domen. Rezultati pokažejo, da je predlagana metoda uporabna za razločevanje rangiranj značilk različnih kvalitet.

1 Introduction

In many application domains, such as bioinformatics and computer vision, supervised learning methods are becoming more frequently applied to high-dimensional problems. In such problems, one typically expects only a relatively small proportion of all input features to be relevant for predicting the output. Also, all relevant features are not equally important. In many practical applications, the problem of discovering the relevant features and/or qualitatively assessing their relative importance can be the main objective of the application of machine learning techniques, even taking precedence over the need to obtain the best possible predictive model. In bioinformatics, for example, the main objective of the analysis of microarray datasets is to identify genes whose expression is, individually or jointly, indicative of some biological state of interest (e.g., a disease), with the goal of improving the understanding of this biological state.

There are two machine learning tasks related to the analysis of feature relevance, namely *feature selection* (FS) and *feature ranking* (FR) [9]. The purpose of feature selection is typically to solve the so-called minimal-optimal problem [15], i.e., to find a minimal subset of features that best explain the output [8]. Feature ranking, on the other hand, solves the so-called all-relevant problem [15], i.e., providing an ordered list of the features from the most to the least important according to a given notion of relevance. Feature ranking methods range from univariate techniques, that assess the relevance of each feature independently of the others (e.g., using mutual information or p-values derived from some statistical test), to multivariate techniques, that derive more complex feature importance scores taking into account potential interactions among the features (e.g., ReliefF [18] or Random forests [2]). These two problems of feature selection and feature ranking are linked: A solution to the feature selection problem can be found by setting a cut-off point on a feature ranking.

The present paper focuses on feature ranking, and more specifically addresses the challenging problem of the evaluation of the output of feature ranking algorithms. Feature selection as stated above is a well-defined optimization problem and as a consequence, the output of two different feature selection methods can be directly compared according to the predictive performance of a model trained from the selected features and/or according to to the size

of the selected subsets. The problem of feature ranking, on the other hand, can not be as easily formulated as an optimisation problem, mainly because there is no commonly accepted notion of feature relevance. Actually, feature ranking methods typically correspond to different definitions of relevance or assumptions of dependence (e.g., univariate versus multivariate, linear versus non-linear). As a consequence, when run on the same dataset, different methods will typically provide different rankings of the features. Determining the best ranking among several ones for a problem at hand is thus a practically very relevant question. For specific problems, this question can be addressed by using domain specific knowledge. In the general case, however, this is an unsolved problem that we would like to address in this paper.

The remainder of this paper is organized as follows. We start by discussing related work in Section 2. We then propose a new algorithmic procedure for evaluating feature rankings that does not require any prior knowledge and can thus be applied on real problems. Following previous works, this method is based on the evaluation of the predictive performance of models trained from nested feature subsets derived from the rankings (described in Section 3). More precisely, two error curves are constructed: the forward feature addition curve (FFA) and the reverse feature addition curve (RFA). They depict the performance of models built on nested feature subsets obtained by taking features from either the top or the bottom of the ranking. Next, we propose a score based on the differences between the FFA and RFA curves as a way to compare different feature ranking methods. We investigate the performance of the proposed method on a wide range of datasets. We start by experiments on the synthetic datasets (Section 4) and proceed with a description of its use on real-world benchmark datasets (Section 5). Section 6 concludes with a summary of our contributions and an outline of possible directions for further work.

2 Related work

The evaluation of feature ranking methods has been typically performed on artificial problems, where the relevant and irrelevant features are known by construction. In such a setting, feature ranking algorithms can be evaluated based on their capability to delineate relevant from irrelevant features. This capability can be measured, for example, through a ROC curve showing the trade-off achieved by the algorithm between assigning high ranks to relevant features and low ranks to irrelevant ones [11]. In the context of the ReliefF algorithm [18] the concepts of *separability* and *usability* are defined to evaluate feature rankings. Separability measures how well the algorithm separates the relevant from the irrelevant features by the difference between the lowest estimated relevance of the relevant features and the highest relevance of the irrelevant features. Usability, on the other hand is defined as the difference between the highest estimated relevance of the relevant features and the highest estimated relevance of the irrelevant features. When a ground truth ranking of the features is known (and not only which features are relevant/irrelevant), finer measures can be used to compare a learnt feature ranking to the ground truth, such as the Spearman's rank correlation.

Evaluating feature ranking methods on artificial problems is very useful to assess a newly proposed ranking algorithm or to highlight overall differences between methods. In practice however, the best method is expected to be problem dependent. This stresses the need for methods to quantitatively assess feature ranking methods in real-world scenarios, where it is not known a-priory which features are relevant and which are irrelevant. In such settings, the performance of feature ranking algorithms has been mostly evaluated from the point of view of their predictive performance associated to feature subsets derived from the rankings.

A way to assess feature rankings is to estimate the predictive performance obtained by using subsets of feature derived from these rankings. For example, for a given number of features k, a ranking A could be considered better than a ranking B if a model trained from the top-k features of ranking A is more accurate than a model trained from the top-k features of ranking B. Variations of this evaluation procedure are given in [9, 7, 16, 21] where the predictive models are compared for different numbers of top-k features.

3 Evaluation method for feature rankings

In general, the purpose of feature ranking algorithms is to solve the all-relevant feature selection problem [15]. However, besides delineating relevant from irrelevant features, a feature ranking algorithm should also provide a proper ordering of features according to their relevance to some target concept. An ideal feature ranking algorithm should produce the ground truth ranking. In reality however, the ranking methods provide only an approximation of it.

A good feature ranking method would produce a ranking that is well ordered. This means that the more relevant features would have a higher rank, i.e., all of them are concentrated at the beginning of the feature ranking. In contrast, a bad feature ranking method is not necessarily the one that produces an inverse ground truth ranking. Namely, we consider as a worst-case scenario if the feature ranking produces a random ranking. In this case the relevant features are uniformly distributed in the ranked list. Estimating and comparing this distribution of relevant features across a ranking is the intuition on which we base our evaluation approach.

3.1 Evaluation method definition

Formally, we would like to evaluate a feature ranking algorithm $r(\cdot)$. The input to the algorithm is a dataset $\mathcal{D}$, consisting of a set of n input features $\mathcal{F}$ and a target F_t, while the output is a feature ranking $\mathbf{R} = r(\mathcal{D})$, i.e., a list whose i-th component gives us the rank of i-th feature.

For an arbitrary feature subset $\boldsymbol{S} \subseteq \mathcal{F}$, we can assess if it contains relevant features by constructing and evaluating predictive models $\mathcal{M}(\boldsymbol{S}, F_t)$. We evaluate them, obtain the value of error measure $\mathrm{err}(\mathcal{M}(\boldsymbol{S}, F_t))$, and decide whether the set $\boldsymbol{S}$ contains important features or not.

The error estimates should provide insight into the correctness of the feature ranking and constitute an evaluation thereof, thus we construct the feature subsets of two types. The sets of the first type, denoted by $\boldsymbol{S}^i$, contain the top i ranked features, $1 \leq i \leq n$. The sets of the second type, denoted by $\boldsymbol{S}_i$, contain the bottom i features. Note that in the special case where $i = n$, i.e., the number of features, we have $\boldsymbol{S}^n = \boldsymbol{S}_n$

For each constructed feature subset $\boldsymbol{S}$, $\boldsymbol{S} = \boldsymbol{S}^i$ or $\boldsymbol{S} = \boldsymbol{S}_i$, we build predictive models $\mathcal{M}(\boldsymbol{S}, F_t)$, and evaluate their prediction errors. In that way, we obtain two curves: the *forward feature addition* (FFA) curve consists of points $(i, \mathrm{FFA}(i)) = (i, \mathcal{M}(\boldsymbol{S}^i, F_t))$ (see Fig. 1a), while the *reverse feature addition* (RFA) curve consists of points $(i, \mathrm{RFA}(i)) = (i, \mathcal{M}(\boldsymbol{S}_i, F_t))$ (see Fig. 1b).

In practical scenarios, if the number of features n is high, running the algorithm might be costly. One simple way for a speed up would be to avoid forming all the feature subsets, and instead add $\Delta i > 1$ features to the set $\boldsymbol{S}^i$ to obtain $\boldsymbol{S}^{i+\Delta i}$. The rationale behind this is that in real-world scenarios involving high-dimensional data, only a small portion of the features are relevant. Therefore, the values of $\mathrm{FFA}(i)$ would not change much when adding more features to a relatively large number of features i. Also, the number of features added can be dependent on i. In the same manner, we can form the set $\boldsymbol{S}_{i+\Delta i}$ from the set $\boldsymbol{S}_i$.

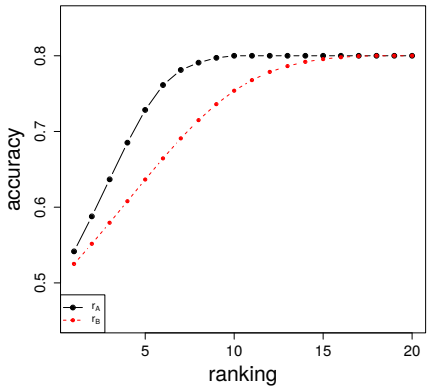
3.2 Quantitative comparison of two rankings

A visual inspection of the curves can only provide a qualitative intuition about which ranking method is better. For quantification purposes, it would be necessary to have a function which provides a cumulative assessment of the differences between the error estimates at different points of the curves. In the most general case, this would be an aggregation function $\mathrm{agg} : \mathbb{R}^n \to \mathbb{R}$, which would take a sequence of the weighted point-wise differences between two curves for its argument. For the FFA curve, we would have

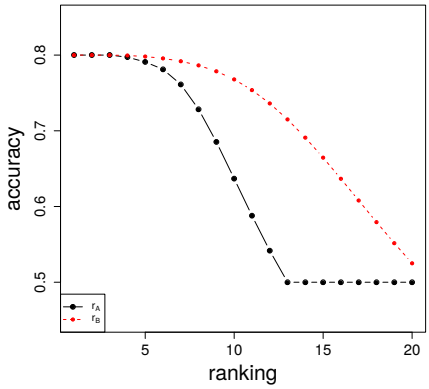
$$\mathrm{FFA}_\delta(r_A, r_B) = \underset{i}{\mathrm{agg}}\ w_i(\mathrm{FFA}_{r_A}(i) - \mathrm{FFA}_{r_B}(i)), \quad (1)$$

while for the RFA curve we would have

$$\mathrm{RFA}_\delta(r_A, r_B) = \underset{i}{\mathrm{agg}}\, w_i(\mathrm{RFA}_{r_A}(i) - \mathrm{RFA}_{r_B}(i)). \quad (2)$$



(a) Comparison of FFA curves



(b) Comparison of RFA curves

Figure 1: Comparison of different ranking methods r_A and r_B

There are several sensible choices for instantiations of the aggregation function agg. The choice depends on the specific task at hand. Considering that we are comparing feature rankings, two aspects are important. The first is the position of most of the relevant features in the ranking. The second relates to the position of the "most" relevant features. In a comparative sense, the first aspect relates to the position of the FFA/RFA curves differences, while the second relates to the magnitude of these differences.

Differences between the FFA/RFA curves of two ranking methods at the beginning of the curves are more important than differences at the end of the curves. Namely, if two FFA curves are different at the beginning, this means that one of the ranking methods is not putting the most relevant features at the top of the ranking. Correspondingly, for the RFA curves, differences at the beginning of the curve (at the bottom of the ranking), mean that one of the feature ranking methods is giving low ranks to features which are relevant. The second aspect is related more to the magnitude of the differences between the FFA/RFA curves than to their position. The intuition is that if "more" relevant features are misranked, then this is worse than "fewer" relevant features being misranked.

From a technical perspective, in order to emphasise the importance of position, the weighting function from

Eqs. 1 and 2, should be a function of the position, i, namely $w_i = f(i)$. In the same manner, in order to emphasise the importance of magnitude, the weighting function should depend on the size of the difference, namely $w_i = f(\delta_i)$ with δ_i the difference between the two compared curves at i. In addition, it is also possible to construct a weighting function that takes into account both position and magnitude, $w_i = f(i, \delta_i)$. To this end, we define four instantiations of Eq. 1 and Eq. 2, which we use to calculate the difference between the FFA/RFA curves from Fig. 1. We consider the following weighting functions:

- $w_i = 1$, equal weight for all differences;
- $w_i = f(i) = 1/|\boldsymbol{S}_i|$, weight inverse to feature subset size;
- $w_i = f(\delta_i) = |\delta_i|$, weight proportional to the difference magnitude;
- $w_i = f(i, \delta_i) = |\delta_i|/|\boldsymbol{S}_i|$, weight which includes both position and magnitude.

The aggregation function used for summarising the differences (in all of the four instantiations) is the weighted average:

$$\underset{i}{\text{agg}}\, w_i\delta_i = \frac{\sum_{i=1}^{n} w_i\delta_i}{\sum_{i=1}^{n} w_i}. \tag{3}$$

The obtained values are given in Table 1. They are calculated for the FFA/RFA examples in Fig. 1a and Fig. 1b. The difference is calculated for r_A with respect to r_B. As seen in Table 1, the values for the FFA curves are positive, which can be interpreted as "r_A is better than r_B". While the values for the RFA curves are negative, the interpretation is the same: "ranking method r_A is better than ranking method r_B".

In order to obtain a single number that quantifies the difference between two feature ranking algorithms, we can combine both values into a single value by calculating the so-called error curve average (ECA)

$$\text{ECA}_\delta(r_A, r_B) = \frac{\text{FFA}_\delta(r_A, r_B) - \text{RFA}_\delta(r_A, r_B)}{2}. \tag{4}$$

Note that the minus sign in the equation is due to the inverse interpretation of negative values for the RFA curve. Namely, if r_A is better than r_B, then the differences of the RFA curves should be negative. This places the overall interpretation of the ECA_δ value on the positive scale. Namely, if r_A is better than r_B, then the overall score should be positive.

w_i	1	$1/\lvert\boldsymbol{S}_i\rvert$	$\lvert\delta_i\rvert$	$\lvert\delta_i\rvert/\lvert\boldsymbol{S}_i\rvert$
FFA_δ	0.018	0.019	0.032	0.03
RFA_δ	-0.042	-0.054	-0.08	-0.077

Table 1: Different quantitative comparisons of error curves

3.3 Quantitative score for a single ranking

In real-world scenarios, the ground truth ranking is not known. Therefore, when evaluating just a single ranking algorithm, the FFA/RFA curve of the algorithm can not be compared to the one of the ground truth ranking. However, the opposite to the ground truth ranking is the uniformly random ranking, for it is the least informative. The motivation for introducing the random ranking FFA/RFA curves is the following: If we can not say how good a single ranking $\mathbf{R}$ is, maybe we can say how close to random it is.

At the point i, the expected value of the FFA/RFA curve, which belong to the uniformly random ranking $\mathbf{R}_{\text{RND}}$, produced by the algorithm $r_{\mathcal{R}}$, is dependent solely on the i and properties of the dataset under consideration. Moreover, it is the same for both the FFA and the RFA curve. For simplicity reasons, we refer to these curves as *expected curves*.

The expected value of the error measure err, is the average of the error estimations of all possible subsets $\boldsymbol{S} \subseteq \mathcal{F}$, whose cardinality equals i, i.e.,

$$E[\text{err}(\mathcal{M}(\boldsymbol{S}, F_t))] = \frac{1}{\binom{n}{i}} \sum_{\substack{\boldsymbol{S}\subseteq\mathcal{F} \\ |\boldsymbol{S}|=i}} \text{err}(\mathcal{M}(\boldsymbol{S}, F_t)) \tag{5}$$

Calculating the expected curves by following Eq. 5 to the letter is intractable, especially for high-dimensional spaces, as we have to consider an exponentially high number of feature subsets. However, for practical purposes, this problem can be circumvented by sampling the space of possible feature subsets for each i.

Suppose we have somehow calculated or approximated the expected FFA/RFA curve. If we have a ranking algorithm r that produces a good (mostly correct) ranking, its FFA curve would be above the expected FFA curve. For the RFA curve, the opposite would apply and the algorithm's curve would be below the expected RFA curve. The score $\text{ECA}_\delta(r, r_{\mathcal{R}})$ between the FFA/RFA curves of this ranking versus the expected curves can thus be used as an absolute quantitative measure of the quality of this ranking. It should be noted that when calculating $\text{ECA}_\delta(r, r_{\mathcal{R}})$ by using $w_i = 1$, it is not necessary to compute the expected curve in order to calculate this ECA_δ score. Indeed, ECA_δ can be simply computed as the sum over all positions of the difference between the FFA and RFA curves we want to evaluate:

$$\begin{aligned} \text{ECA}_\delta(r, r_{\mathcal{R}}) &= \frac{(\text{FFA}_\delta(r, r_{\mathcal{R}}) - \text{RFA}_\delta(r, r_{\mathcal{R}}))}{2} \\ &= \frac{1}{2}\left(\sum_{i=1}^{n} \frac{\text{FFA}_r(i) - \text{RFA}_r(i)}{n}\right), \end{aligned}$$

since $\text{FFA}_{r_{\mathcal{R}}}(i) = \text{RFA}_{r_{\mathcal{R}}}(i)$.

4 Evaluation on synthetic data

The goal of the experiments presented in this section is to demonstrate the usefulness of our feature ranking evaluation method. As previously mentioned, feature ranking

methods provide an approximation of the ground truth ranking that can be viewed as a noisy ground truth. A noisier ranking is more distant from the ground truth ranking and therefore of worse quality.

An evaluation method should be sensitive to the amount of noise and should provide a corresponding quality estimate of the feature ranking. For that purpose, we design experiments to demonstrate that our evaluation method is sensitive to the addition of noise to the ground truth ranking. We first generate noisy feature rankings and then construct FFA/RFA curves from them.

4.1 Generating synthetic data

We first perform an empirical evaluation of the proposed notion of FFA/RFA curves in a controlled setting by using synthetic datasets. The main advantage of using synthetic data is the possibility of defining a good baseline ranking that allows us to assess our proposed feature ranking evaluation method.

The complete statistics of the generated datasets and their feature interaction sets are summarized in Table 2. All of the datasets consisted of 1000 instances and 100 features in total. Among the 100 features, the "single" dataset has 9 relevant features, the "pair" dataset contains 18 relevant features and the "combined" dataset contains 27 relevant features. In all three datasets, every set $\mathcal{F}_{\text{int}}$ of relevant features has two additional redundant copies. Irrelevant features are realized independently of each other. More details on the generation of the datasets are available in [20].

For each dataset, we would like to define a good baseline ranking against which to compare feature ranking methods. We define this ranking from feature relevance scores $\text{rel}(F_i, F_t)$ for each feature F_i, calculated directly from the specified feature interaction structure, by using the following equation:

$$\text{rel}(\text{F}_\text{i}, \text{F}_\text{t}) = \frac{\text{I}(\mathcal{F}_{\text{int}}; F_t)}{|\mathcal{F}_{\text{int}}|},$$

where $\mathcal{F}_{\text{int}}$ is the (unique) interaction set that contains F_i and $I(\mathcal{F}_{\text{int}}; F_t)$ is the mutual information between features in $\mathcal{F}_{\text{int}}$ and the target F_t. By dividing the mutual information by the number of features, we distribute the information equally between all features in an interaction set. As a consequence, features that brings information about the target F_t individually are considered more relevant than features that bring the same amount of information about the target only in conjunction with other features.

Note that this baseline ranking is not guaranteed to be optimal in terms of the FFA and RFA curves for a given learning algorithm, but is nevertheless expected to be close to optimal. In our experiments, we will consider this ranking as a ground truth ranking, denoted R_{GT}, against which we will compare other rankings.

n	$\lvert\mathcal{F}_{\text{int}}\rvert$	$f(\mathcal{F}_{\text{int}})$	P
3	1	F_i	0.8
3	1	F_i	0.7
3	1	F_i	0.6
91	1	F_i	0.5

(a) "single" dataset

n	$\lvert\mathcal{F}_{\text{int}}\rvert$	$f(\mathcal{F}_{\text{int}})$	P
3	2	$\text{XOR}(F_i, F_j)$	0.8
3	2	$\text{XOR}(F_i, F_j)$	0.7
3	2	$\text{XOR}(F_i, F_j)$	0.6
82	1	F_i	0.5

(b) "pair" dataset

n	$\lvert\mathcal{F}_{\text{int}}\rvert$	$f(\mathcal{F}_{\text{int}})$	P
3	2	$\text{XOR}(F_i, F_j)$	0.8
3	2	$\text{XOR}(F_i, F_j)$	0.7
3	2	$\text{XOR}(F_i, F_j)$	0.6
3	1	F_i	0.8
3	1	F_i	0.7
3	1	F_i	0.6
73	1	F_i	0.5

(c) "combined" dataset

Table 2: Synthetic datasets statistics: The feature interaction sets ($\mathcal{F}_{\text{int}}$) contained in each dataset; The interaction function for the feature sets ($f(\mathcal{F}_{\text{int}})$); The values $P(f(\mathcal{F}_{\text{int}}) = F_t)$ are denoted by P. The value of n in the last row of each table corresponds to the number of irrelevant features in a dataset. In the other rows, n denotes the number of copies of each interaction set, which are identically defined but independently realized (and differ in the random component).

4.2 Adding noise to the ground truth ranking

The noise is introduced into the ranking by selecting a proportion, θ, of the features, which are randomly selected. For these features, random relevance values are assigned while the remaining features preserve their ground truth relevance. By considering these partially changed relevance values, a new noisy feature ranking, $\mathbf{R}_\theta$, is defined.

As the random relevance values can be distributed differently throughout the ranking, different FFA/RFA curves can be constructed for the same amount of noise.

We estimate the expected error values by sampling the space of possible FFA/RFA curves for a given θ. First, we generate N different noisy feature rankings and then construct N FFA/RFA curves based on them. The expected values of FFA/RFA curve are estimated by averaging the

N individual curves, namely

$$E[\text{FFA}]_\theta = \frac{1}{N}\sum_{i=1}^{N}\text{FFA}_{\theta,i}$$

$$E[\text{RFA}]_\theta = \frac{1}{N}\sum_{i=1}^{N}\text{RFA}_{\theta,i}$$

for a specified N and θ.

For estimating the error values of the FFA/RFA curves, SVMs with a polynomial (quadratic) kernel were used and a 10-fold cross validation was performed on the dataset under consideration. The epsilon parameter of the SVMs was set to 1.0E-12, while the complexity parameter was set to 0.1.

For our experiments, we consider several different amounts of noise θ, namely: 5%, 10%, 15%, 20%, 30% and 50%, as well as the completely random ranking (100% of noise). Each noisy error curve was produced by summarizing the errors of 100 noisy rankings produced for a given θ. We additionally constructed error curves based on the ground truth ranking.

The experiments were performed on the three synthetic datasets described in Section 4.1, each with its corresponding ground truth ranking, $\mathbf{R}_{GT}$.

4.3 Results on synthetic data

The results of our experiments are first plotted as graphs containing error curves. In Fig. 2, we only show the curves obtained on the "combined" dataset. These curves are representative of the curves obtained on the other datasets.

The FFA/RFA curves plotted on each graph are for rankings with different noise levels θ, as well as for the ground truth $\mathbf{R}_{GT}$ and random rankings. In both Figs. 2a and 2b, the FFA and the RFA curves seem to be sensitive to the addition of noise. To begin with, the FFA/RFA curves of all the noisy rankings are located between the ground truth ranking FFA/RFA curve and the random ranking FFA/RFA curve. As noise is added to the ground truth ranking, the FFA/RFA estimates are slowly moving away from the ground truth FFA/RFA curve and towards the random ranking FFA/RFA curve.

Next, for performing quantitative analysis of the feature rankings, we begin by summarising the differences of the noisy rankings error curves w.r.t. the ground truth error curve. Additionally, some kind of baseline is required for comparison. As the ground truth ranking is known, the distance between the ground truth ranking and the noisy rankings can serve as a baseline.

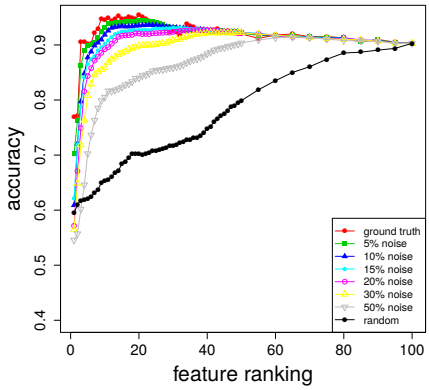
For summarising the differences between the noisy rankings FFA/RFA curves we use the ECA difference, calculated by using Eq. 4 from Section 3.2. For comparative purposes,when calculating the ECA differences, we use the different weighting functions as discussed in Section 3.2.

For calculating the baseline values, i.e., the distance between the ground truth ranking $\mathbf{R}_{GT}$ and the noisy rankings $\mathbf{R}_{\theta,i}$, we use the average Spearman rank correlation coefficient ρ between the vectors $\mathbf{R}_{GT}$ and $\mathbf{R}_{\theta,i}$. The i-th component of such a vector gives the rank of the i-th feature in dataset. The distance between rankings is then computed as

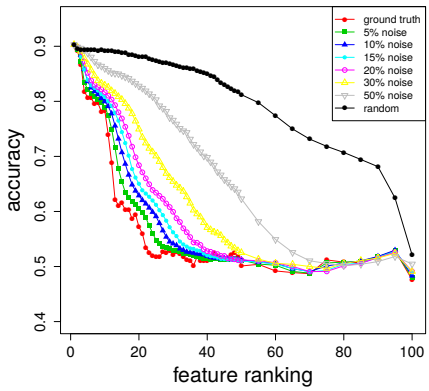
$$\text{dist}(\mathbf{R}_{GT},\mathbf{R}_\theta) = 1 - \bar{\rho}_{GT,\theta} = 1 - \frac{1}{N}\sum_{i=1}^{N}\rho(\mathbf{R}_{GT},\mathbf{R}_{\theta,i})$$

where N is the number of different noisy rankings considered for a given θ.

We obtain the results for all of the three synthetic datasets. Since there are no major differences among them, we show summarised results only for the "combined" dataset. Table 3 contains values calculated with respect to the ground truth ranking. The first row of the table refers to the distance $\text{dist}(\mathbf{R}_{GT},\mathbf{R}_\theta)$. The other rows are the ECA differences between the FFA/RFA curves of the GT ranking and the FFA/RFA curves of the noisy rankings. Each row containing the ECA differences refers to different weighting functions. All columns, except the last one, refer to different levels of noise, θ. The final column gives the correlation between $\text{dist}(\mathbf{R}_{GT},\mathbf{R}_\theta)$ (row one) and the FFA/RFA curve distances (rows 2 to 5), across different



(a) FFA curves for the "combined" dataset



(b) RFA curves for the "combined" dataset

Figure 2: Plots comparing the FFA (on the left) and RFA (on the right) curves for the "combined" dataset. Each figure contains error curves for the ground truth ranking, rankings with different noise levels θ and the random ranking.

	$\theta = 0.05$	$\theta = 0.1$	$\theta = 0.15$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.5$	$\theta = 1$	
dist	0.1	0.171	0.252	0.32	0.432	0.652	1.048	**corr.**
$w = 1$	0.009	0.02	0.027	0.037	0.061	0.117	0.223	0.992
$w = 1/r$	0.018	0.042	0.047	0.064	0.084	0.132	0.178	0.991
$w = \|\delta\|$	0.029	0.061	0.070	0.09	0.115	0.174	0.263	0.998
$w = \|\delta\|$ /r	0.044	0.091	0.095	0.121	0.142	0.199	0.254	0.982

Table 3: Comparison of different ECA values obtained by different weighting functions w. The ECA values are compared with the distance between the noisy rankings R_θ and the GT ranking R_{GT}. The final column of each table "corr." is the value of the correlation coefficient calculated between the ranking distance (first row) and each of the ECA difference rows.

noise levels θ.

The final column gives an indication of how well the ECA differences relate to the distance between the ground truth ranking and the noisy rankings. As it can be seen, the curve distances correlate very well to the rank distances, regardless of which weighting function is used.

From this quantitative analysis, it can be concluded that the ECA difference derived from the error curves has the same sensitivity to noise as the actual distance between the ground truth and the noisy rankings. This implies that our method can be used in practical scenarios not just to qualitatively distinguish between different rankings, but also to quantify the difference between them. As for the specific weights used for calculating the ECA differences, it can be concluded that any of the considered weighting schemes can be used to properly compare the error curves.

5 Evaluation on real data

Thus far, our analysis only involved artificially generated problems. In this section, we want to illustrate the use of our feature ranking evaluation method on datasets originating from various real-life domains. The purpose of the experiments is to examine the quality of the feature rankings produced by several feature ranking methods on data with different characteristics.

The analysis is primarily a comparative one, performed solely by calculating the numeric scores derived from the FFA and RFA error curves. The datasets we consider are quite diverse, with unknown interaction structure and therefore unknown ground truth ranking. However, for each dataset, it is possible to generate the expected error curves of random rankings. These expected curves are used as a baseline for comparing the different feature ranking methods.

5.1 Datasets description

For our experiments, 28 diverse classification datasets with a single target class were selected. Most of them originate from the UCI data repository [14] and are from various domains. Of the remaining 3 datasets, one is from a medical study of acute abdominal pain in children (aapc) [4], while the remaining two ("water" and "diversity") are from an ecological study of river water quality [5].

Besides covering different domains (including biology, medicine, ecology etc.) these datasets have a wide range of different properties, including number/type of features and number of instances.

The main characteristics of each dataset are summarised in Table 4.

Dataset	#Inst.	#Feat.	#Cl.
aapc	335	84	3
amlPrognosis	54	12625	2
arrhythmia	452	279	16
australian	690	14	2
bladderCancer	40	5724	3
breast-cancer	286	9	2
breast-w	699	9	2
breastCancer	24	12625	2
car	1728	6	4
chess	3196	36	2
childhoodAll	110	8280	2
cmlTreatment	28	12625	2
colon	62	2000	2
diversity	292	86	5
dlbcl	77	7070	2
german	1000	20	2
heart	270	13	2
heart-c	303	13	2
heart-h	294	13	2
ionosphere	351	34	2
leukemia	72	5147	2
mll	72	12533	3
prostate	102	12533	2
sonar	208	60	2
srbct	83	2308	4
tic-tac-toe	958	9	2
water	292	80	5
waveform	5000	21	3

Table 4: Statistics for the benchmark datasets

5.2 Experimental setup

Four feature ranking methods were applied to each dataset:

- **Information gain**, calculating the information gain of each feature F_i as $IG(F_t, F_i) = H(F_t) - H(F_t|F_i)$. This does not require any specific parameter setting.
- **SVM-RFE** is the recursive feature elimination (RFE) procedure that employs an SVM to evaluate the feature weights at each iteration. A linear SVM was employed [9] with the epsilon parameter set to 1.0E-12, while the complexity parameter was set to 0.1.
- **ReliefF** algorithm as proposed in [18]. The number of neighbours was set to 10 and all of the instances were used for estimating the relevance values.

– **Random forests**, which can be used for estimating feature relevance as described in [2]. A forest of 100 trees was used, constructed by randomly choosing a $\log_2$ of the number of features.

To generate the error curves, SVMs with polynomial (quadratic) kernel, were employed as classifiers. The epsilon parameter was set to 1.0E-12, while the complexity parameter was set to 0.1. This classifier was showed to be appropriate in our previous experiments [20].

As a baseline of the comparison, *expected* FFA and RFA *curves* were used. They were produced by generating 100 random rankings for each dataset under consideration. This was done in a similar manner as described in Section 3.3.

5.3 Results on real data

The results summarizing the error curves average (ECA) differences are given in Table 5. The ECA differences are calculated by using Eq. 4 and the weighting function $w_i = 1$, i.e., as a standard mean value. Each row of Table 5 refers to a single dataset, while each column corresponds to a single feature ranking method. The ECA values in the table are calculated w.r.t. the baseline error curve, namely, the expected error curve. This gives an indication of how much each feature ranking method is better than a random ranking generator, but also allows for comparison between the quality of the feature rankings of the different methods.

A positive value of an ECA difference indicates that a feature ranking method performs better than the random ranking generator. The negative values, however, do not necessarily indicate that it performs worse than random, but that it provides a non-random ranking that is inverse to the correct one. A value close to zero means the feature ranking method provides rankings that are more (or less) random.

An intial inspection of the results in Table 5 reveals that random forests often have negative ECA values. The FFA and RFA curves of random forests, for these particular datasets, are below/over the expected FFA/RFA curves of random rankings. Upon closer inspection of their feature rankings (results not shown here due to space limitations) we find that they are inverse to those of the other feature ranking methods.

In order to summarise the results from Table 5 and to draw meaningful conclusions about the performance of the different ranking methods across the different datasets, we use statistical tests. We adopt the recommendations of Demšar [3] and use the Friedman [6] test for statistical significance with the correction by Iman [10]. If the null hypothesis H_0 that all ranking methods perform equally well, can be rejected, we use the Nemenyi post-hoc test [13] and additionally check between which feature ranking methods the statistically significant differences appear. The level of significance $p = 0.05$ was used.

When comparing the four feature ranking methods, statistically significant differences occur. We present the results with a critical distance diagram [3] in Fig. 3. In the diagram, the feature ranking methods are ordered according to which one is better on average (across all datasets). A method is better if it is positioned closer to the value one on the axis. It can be observed that ReliefF, Info Gain and SVM-RFE significantly outperform Random Forests, while not differing significantly among each other.

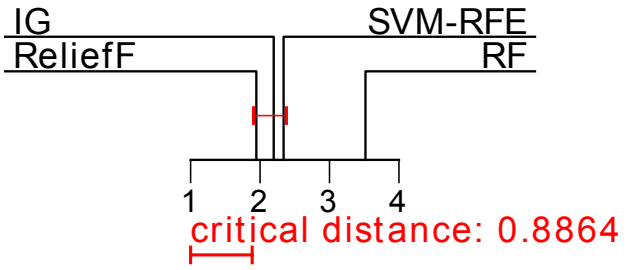


Figure 3: Critical distance diagrams representing the statistical comparison of the ECA differences of three ranking methods on the 28 datasets. The critical distance is calculated for a p value of 0.05 and is represented by a horizontal line. If the feature ranking methods are connected by a line, then their performance is not statistically significantly different.

dataset	IG	RF	ReliefF	SVM-RFE
aapc	0.269	0.299	**0.316**	0.297
amlPrognosis	**0.056**	0.007	0.027	0.043
arrhythmia	0.041	0.041	**0.057**	0.053
australian	**0.277**	0.260	0.266	0.209
bladderCancer	0.125	0.059	**0.167**	0.161
breast-cancer	**0.025**	0.013	0.012	−0.003
breast-w	**0.246**	0.206	0.190	0.194
breastCancer	0.050	0.037	**0.128**	0.110
car	**0.085**	−0.081	0.079	0.066
chess	0.279	−0.056	**0.283**	0.248
childhoodAll	0.083	0.040	0.033	**0.154**
cmlTreatment	**0.028**	−0.009	−0.026	0.004
colon	0.099	0.049	**0.163**	0.116
diversity	0.167	0.192	**0.215**	0.149
dlbcl	0.032	0.008	0.067	**0.086**
german	**0.023**	−0.002	0.013	0.022
heart	**0.159**	0.039	0.150	0.130
heart-c	**0.178**	0.057	0.163	0.163
heart-h	0.146	0.058	0.110	**0.147**
ionosphere	0.116	0.088	0.041	**0.136**
leukemia	0.140	0.056	**0.175**	0.164
mll	0.118	0.045	**0.355**	0.281
prostate	0.212	0.067	**0.236**	0.232
sonar	0.066	0.060	**0.096**	0.070
srbct	0.142	0.084	**0.292**	0.261
tic-tac-toe	0.072	−0.052	**0.082**	0.069
water	0.193	0.181	**0.217**	0.144
waveform	0.180	−0.190	0.188	**0.210**

Table 5: ECA differences between the FFA/RFA curves of four feature ranking methods w.r.t. the curves of a random ranking. The missing values are due to SVM-RFE's inability to handle multi-valued discrete/nominal attributes. Boldfaced values are the largest ECA differences in each row.

6 Conclusions

In this paper, we focus on the problem of evaluating the output of feature ranking algorithms. We define and formalize an intuitive evaluation method for quantitative comparison of feature rankings. The method is based on iterative construction and evaluation of predictive models, resulting in so-called error curves: forward feature addition curve (FFA), starting from the top of a feature ranking, and the reverse feature addition curve (RFA), starting from the bottom of a ranking. From these two curves, we calculate the error curves average (ECA) difference that we propose as a numerical indicator for comparing different feature rankings.

We first test our method in a controlled environment on synthetic data. We compare feature rankings with different amount of added noise, starting from the known ground truth ranking and ending with completely random rankings. By comparing the different ECA values obtained for the different noise levels, we show that our method is sensitive to changes in the quality of the feature ranking.

In order to demonstrate the practical application of our evaluation method, we consider a collection of classification datasets from various domains with different properties. We compare the performance of four feature ranking methods across these different datasets and evaluate their outputs by using our proposed method. The analysis of the comparative evaluation shows that the best algorithm is often domain dependent and often simple approaches such as info gain can be used to produce a proper feature ranking.

Several directions of work can be taken to further develop the proposed evaluation methodology. The first is to directly use the feature relevance values produced by the ranking algorithm when inducing predictive models. This can be easily done in feature-weighted classifiers, such as weighted kNN. The second concerns feature ranking stability, another important aspect of the feature ranking process. Although we have not considered it explicitly in this work, we would like to include it in the feature ranking evaluation process, in a manner similar to that of [19]. Also, as structured data [1] are becoming increasingly common, we would like to adapt and investigate our method for different types of structured targets. To this end, we need to use a feature ranking method for structured targets and couple it with a predictive model for structured outputs [12, 17].

Acknowledgements

IS would like to gratefully acknowledge the financial support of The Ad Futura Slovene Human Resources Development and Scholarship Fund. SD, DK, and MP have been supported by the Slovenian Research Agency through the program P2-0103, the project L2-7509, and a young researcher grant, respectively. The work has also been supported by the European Commission through the H2020 grant number 720270 (HBP SGA1).

References

[1] Gökhan H. Bakır, Thomas Hofmann, Bernhard Schölkopf, Alexander J. Smola, Ben Taskar, and S. V. N. Vishwanathan, editors. *Predicting Structured Data*. The MIT Press, Cambridge, Massachusetts, 2007.

[2] Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[3] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[4] Saso Džeroski, George Potamias, Vassilis Moustakis, and Giorgos Charissis. Automated revision of expert rules for treating acute abdominal pain in children. In *Artificial intelligence in medecine - AIME, LNCS 1211*, pages 98–109, 1997.

[5] Sašo Džeroski, Damjan Demšar, and Jasna Grbović. Predicting chemical parameters of river water quality from bioindicator data. *Applied Intelligence*, 13:7–17, 2000.

[6] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics*, 11:86–92, 1940.

[7] Cesare Furlanello, Maria Serafini, Stefano Merler, and Giuseppe Jurman. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. *BMC Bioinformatics*, 4:54, 2003.

[8] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, March 2003.

[9] Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, March 2002.

[10] Ronald Iman and James Davenport. Approximations of the critical region of the Friedman statistic. *Communications in Statistics - Theory and Methods*, 9:571–595, 1980.

[11] Kees Jong, Jérémie Mary, Antoine Cornuéjols, Elena Marchiori, and Michèle Sebag. Ensemble feature ranking. In *PKDD - LNCS 2302*, pages 267–278, 2004.

[12] Dragi Kocev, Ivica Slavkov, and Sašo Džeroski. More is better: ranking with multiple targets for biomarker discovery. In *Proc. Second International Workshop on Machine Learning in Systems Biology*, page 133, University of Liege, Belgium, 2008.

[13] Peter Bjorn Nemenyi. *Distribution-free Multiple Comparisons*. PhD thesis, Princeton University, Princeton, NY, USA, 1963.

[14] C.L. Blake D.J. Newman and C.J. Merz. UCI repository of machine learning databases, 1998. https://archive.ics.uci.edu/ml/datasets.html. Accessed on: 2015-12-13.

[15] Roland Nilsson, José M. Peña, Johan Björkegren, and Jesper Tegnér. Consistent feature selection for pattern recognition in polynomial time. *Journal of Machine Learning Research*, 8:589–612, December 2007.

[16] Silvano Paoli, Giuseppe Jurman, Davide Albanese, Stefano Merler, and Cesare Furlanello. Semisupervised profiling of gene expressions and clinical data. In *Proc. Sixth International Conference on Fuzzy Logic and Applications*, pages 284–289, 2005.

[17] Matej Petković, Sašo Džeroski, and Dragi Kocev. Feature ranking for multi-target regression with tree ensemble methods. In *Discovery Science*, pages 171–185, 2017.

[18] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53:23–69, 2003.

[19] Yvan Saeys, Thomas Abeel, and Yves Van de Peer. Robust feature selection using ensemble feature selection techniques. In *ECML/PKDD, LNCS 5212*, pages 313–325, 2008.

[20] Ivica Slavkov. *An Evaluation Method for Feature Rankings*. PhD thesis, IPS Jožef Stefan, Ljubljana, Slovenia, 2012.

[21] Antanas Verikas, Adas Gelzinis, and Marija Bacauskiene. Mining data with random forests: A survey and results of new tests. *Pattern Recognition*, 44:330–349, 2011.

Arguments in Interactive Machine Learning

Martin Možina
Faculty of Computer and Information Science, University of Ljubljana, Slovenia
E-mail: martin.mozina@fri.uni-lj.si

Keywords: argumentation, interactive machine learning, argument-based machine learning

Received: November 7, 2017

In most applications of machine learning, domain experts provide domain specific knowledge. From previous experience it is known that domain experts are unable to provide all relevant knowledge in advance, but need to see some results of machine learning first. Interactive machine learning, where experts and machine learning algorithm improve the model in turns, seems to solve this problem. In this position paper, we propose to use arguments in interaction between machine learning and experts. Since using and understanding arguments is a practical skill that humans learn in everyday life, we believe that arguments will help experts to better understand the models, facilitate easier elicitation of new knowledge from experts, and can be intuitively integrated in machine learning. We describe an argument-based dialogue, which is based on a series of steps such as questions and arguments, that can help obtain from a domain expert exactly that knowledge which is missing in the current model.

Povzetek: V strojnem učenju je pridobivanje domenskega znanja pogosto prvi korak, ključen za definicijo učnih primerov, njihovih opisov in cilja učenja. Težava je, da eksperti večinoma niso sposobni dobro izraziti svojega znanja. Lažje je, če jim najprej pokažemo preliminarne, čeprav napačne rezultate strojnega učenja, saj eksperti tako lažje uvidijo, katero domensko znanje strojno učenje potrebuje. Postopek, kjer strojno učenje in ekspert izmenjaje izboljšujeta naučeni model, se imenuje interaktivno strojno učenje. V tem članku predlagamo uporabo argumentov v komunikaciji med računalnikom in ekspertom. Ljudje se argumentiranja naučimo zgodaj in ga veliko uporabljamo. Če bi računalniki znali svoje znanje predstaviti s pomočjo argumentov ter znali upoštevati človeške argumente pri svojem učenju, bi to vodilo do lažje komunikacije in posledično do bolj točnih in bolj razumljivih računalniških modelov. V članku pokažemo, kako vključiti argumentacijo v strojno učenje in opišemo ključna vprašanja ter odgovore v dialogu med strojnim učenjem in ekspertom, ki vodijo do tistega domenskega znanja, ki naučenemu modelu manjka.

1 Introduction

Domain experts are often involved in the development of a machine learning application. They help define the machine learning problem, provide learning examples, labels, and attributes of these examples. In some cases, they are even able to provide prior knowledge that is then incorporated into machine learning algorithms, which often results in more accurate and comprehensible models.

Acquiring domain knowledge is therefore one of the key tasks in machine learning, unfortunately a very difficult one, known also as the Feigenbaum knowledge acquisition bottleneck [4]. Domingos [2] identified several reasons why combining machine learning and expert knowledge often fails and how it should be approached. One of the reasons is that the results of machine learning are rarely optimal on the first attempt. An iterative improvement, where experts and computer improve the model in turns is needed. Furthermore, some knowledge is hard to make explicit. It turns out that humans are much better at explaining particular cases than eliciting general knowledge.

There are more and more machine learning studies using iterative improvements. Fails et al. [3] used the term *interactive machine learning* to describe an iterative system for correcting errors of an image segmentation system. Since then, researchers have presented many advantages of systems that allow users to interact with machine learning. Beside having better final performance, such as accuracy score, these works report that users also gain trust and understanding of their systems. A particularly interesting one was introduced by Stumpf [18], where a user can comment on automatically generated explanations provided by a learned model. These comments are then used as constraints in the system when relearning the model. Kulesza [9] called such an interaction *explanatory debugging*, because users identify "bugs" in a system by inspecting explanations and then explain necessary corrections back to the system.

We propose a similar approach that targets domain experts instead of end users. Explanatory debugging aims at building flexible applications, which can easily conform to the preferences of a user. In a spam filtering application, for example, an explanation might include the words that contributed to the prediction. When a user disagrees with the prediction, she can select some of these words and mark them as not being indicative of spam. The system must then

reduce the influence of these words in the future.

In our case, we focus on enabling the domain experts to elicit their knowledge in the development of a machine learning application. Our approach is less constrained, because experts can use general arguments to explain and to provide feedback back to machine learning. It seems that argument is the right tool for this problem, as humans have a lot of experience with arguments. We are using them every day to convince, negotiate, express and explain our opinions.

An argument in its simplest form is expressed as a set of premises that support a conclusion. In most cases, the link between premises and conclusion is not deductive, but presumptive. Consider, for example, the following argument:

Premise 1: Raising taxes increases government revenues.

Premise 2: Government needs money.

Conclusion: Government should raise taxes.

This argument is plausible, because raising taxes can increase revenues. It is also possible that it does not, if the taxes are already too high. However, when such an argument is put forward, involved parties understand that the conclusion might not be correct. If domain experts used arguments to express their knowledge, it would not be absolutely correct, however they would be able to express their domain knowledge more easily and in a natural way.

In this position paper we do not present any machine learning algorithms, experiments or results. Instead, we motivate the use of arguments in interactions between domain experts and machine learning. The motivation is based on the following two reasons: a) humans are already well practiced in argumentation and b) with some changes machine learning algorithms can communicate using arguments. In this paper, we focus on the second reason, since human argumentation is already well covered in the literature [19]. We identify what modifications of machine learning algorithms are needed to enable the use of arguments and which questions should we ask the domain experts to receive the most relevant information.

The main contribution of this work are instructions how to enable a general machine learning algorithm to use arguments. This includes presenting explanations in terms of arguments and the definition of the constraint that arguments impose on learning. In the previous work [14] we presented an actual implementation of learning rules from arguments. In this paper, this idea is generalized. Another contribution is a description of the refinement loop (a list of steps) for obtaining the most relevant knowledge from the domain expert. We have already presented several versions of this loop in our previous publications (see [14, 8]). Here, we unify these versions and provide more detailed explanations of the steps with practical examples. Finally, this paper motivates the use of arguments in interactive machine learning and supports this motivation with arguments.

2 Explaining classifications and arguments

Explaining decision or actions of intelligent systems to end users has many benefits [11]. It can positively affect the systems use, enable better understanding of the system and result in making people trust it.

Some machine learning models have the inherent capability of generating explanations, such as decision trees or classification rules [5]. Similarly, additive models, such as naïve Bayes or logistic regression, can use weights given to features to provide explanations of their decisions [15].

However, most of the contemporary machine learning research focuses on optimizing some abstract evaluation measure, such as classification accuracy or root mean squared error, and does not consider explanations at all. There have been some attempts to explain the decisions of of such methods. For example, Štrumbelj and Kononenko [17] suggested an algorithm for generating explanations of incomprehensible methods. They evaluate prediction importance of each feature by computing the difference between the classifier's prediction of an example and the prediction of the same example when this feature is omitted. This difference is then used in the explanation of the classifier's decision for this particular example.

We shall now define the relation between explanations in machine learning and arguments. We mentioned in the introduction that an argument contains a set of premises to support a conclusion. Since classification is the conclusion of the machine learning system, and the explanation contains the main reasons for this conclusion, it seems that an explained classification is already an argument.

Explanations of classifications rarely contain one argument only. Usually, an explanation provides reasons for and against the predicted class. For example, in the case of classification rules, we can present all rules covering the classifying example. Similarly, in nomograms [15] or in the general feature-based explanation framework [17], influences of features can be either positive or negative. Showing reasons for and against predicted class is beneficial to a domain expert, since it shows all relevant information that the underlying system used to infer a decision, which increases expert's understanding of the system [9].

An explanation thus contains arguments for and arguments against the predicted class value, without explaining the actual details of the algorithm for inferring the final decision. Knowing positive and negative factors is often sufficient for human understanding, as it is similar to how arguments are used in a human conversation. In a dialogue between two persons, arguments supporting one side are often challenged with the opposing arguments. It is not rare that the same set of arguments will lead to different conclusions of the participants in the dialogue, because they have different viewpoints and employ different internal reasoning mechanisms. Yet, knowing the opposing arguments is still beneficial, because they increase our understanding of the opposing viewpoints and therefore deepen our under-

standing of the issue. By analogy, it is more important for experts to understand which factors, both positive and negative, influenced the machine learning decision, and less how it was derived.

3 Argument-based machine learning

Argument-based machine learning (ABML) is a special case of learning from data and prior knowledge, where prior knowledge is represented with arguments [14]. A specific property of arguments is that they relate to a single example only and are not general as prior knowledge usually is. Several reviews and comparisons of different applications of prior knowledge are available, see for example [6, 12, 20].

The problem with domain knowledge occurs when experts are asked to provide general knowledge. Consider, for example, asking a physician to write down general rules for diagnosing pneumonia. A very difficult task. On the other hand, this physician can easily diagnose a certain patient and explain why he has pneumonia. For this reason we suggest using arguments to elicit and represent background knowledge. While asking experts to provide general background knowledge can be a difficult task, asking them to articulate their knowledge through arguments has proved to be much more efficient [2, 8].

In ABML, arguments are used to enhance learning examples. Each argument is attached to a single learning example, while one example can have several arguments. There are two types of arguments: positive arguments are used to explain (or argue) why a certain learning example is in the class as given, and negative arguments are used to explain why it should not be in the class as given. Examples with attached arguments are called *argumented examples*.

An ABML method needs to induce a model that will explain the classification of an example using the arguments provided by the expert. An ABML method, therefore, needs to be able to explain its decisions with arguments for and against, as we described in the previous section. Moreover, it needs to be able to accept input arguments and use these arguments in explanation, which allows the expert to immediately see the impact. The reasons from the positive arguments should become a part of the arguments for the class value in the explanation, and the reasons from the negative arguments should be mentioned within the against arguments in the explanation. Such explanation is therefore more comprehensible from the expert's perspective, since it uses the same terms as the expert [8].

For example, a diagnostic machine learning system might argue that a patient probably has pneumonia, because he is a male and he is coughing. A medical expert could then counter argue that this person has pneumonia, because he has high temperature. Then, ABML should induce a new model for automatic diagnosis, which would state high temperature (among others) as the reason for this particular patient with pneumonia.

Such instance-based constraints are different from how constraints are usually implemented in machine learning, because they relate to one example only. The system does not need to mention temperature in explanations of other examples, in fact, it could even mention low temperature in explanations of other patients with pneumonia, and that would still not violate the constraint.

Arguments are presumptive by nature and that is the main reason why arguments can not be applied generally, but to specific examples only. When a medical doctor explains a diagnosis of a patient, his or her argument contains many unstated premises that seemed unimportant at a time or were simply forgotten. Maybe fever is typical only for a certain type of pneumonia or only for a certain part of the population.

To implement an argument-based variant of a machine learning algorithm, one needs to take care that the arguments from experts are mentioned in explanations. This is easier achieved with models that are a composition of several parts. For example, an argument-based random forest could simply select only trees that are consistent with arguments. In our research group we implemented the ABCN2[1] algorithm [14], an extension of the CN2 algortihm [1], which learns classification rules from argumented examples. The main difference between the original CN2 and ABCN2 algorithms is in the definition of the covering relation. In the standard definition, a rule covers an example if the condition part is true for this example. In ABCN2, a rule covers an argumented example if the condition part is true and rule is consistent with positive arguments and not consistent with negative arguments.

4 A dialogue between a domain expert and a knowledge engineer

Although interactive machine learning assumes that end users (domain experts) directly interact with machine learning algorithms, we shall assume that a knowledge engineer acts as an intermediate between a domain expert and the algorithm, since some of the suggested steps in this section would be difficult to implement automatically.

Having a machine learning algorithm that can generate arguments and can accept expert's arguments, we will now define how a domain expert and a knowledge engineer can interact with arguments. We propose a series of moves that defines a dialogue between a domain expert and a knowledge engineer. A dialogue is a goal-directed conversation between two parties, in which parties are taking turns. In each turn, a participant makes a move that responds to the previous move. In this information-seeking dialogue, a knowledge engineer is trying to elicit relevant information from a domain expert by selecting relevant examples, using explanations of these examples, and asking the right questions. In our previous previous applications of ABML, we

[1]The latest version of ABCN2 can be found at `https://github.com/martinmozina/orange3-abml`.

called this the ABML refinement loop [8]. In this paper, we unify several versions of this loop and present it in the context of the ideas from the previous two sections. Furthermore, the descriptions of the steps cover many different situations where the dialogue might lead us to.

Given that arguments always relate to a single example, an engineer and an expert talk about one example at a time. As it is unlikely that experts will have time to discuss all learning examples, selecting relevant examples is important. We call these examples *critical examples*. A discussion about a single critical example has the following seven steps.

Step 1: Selecting a critical example

Critical examples are those learning examples that would have a considerable positive influence on the quality of the model if some arguments were provided. Initially, we took misclassified examples with the highest predictive error as critical examples [8]. However, in our recent experiments, we discovered that it is better to select prototypical misclassified examples, as examples with the highest error are more likely to be outliers and are therefore hard to explain. There are various algorithms available for obtaining prototypical examples, one option is to use clustering and take centers of these clusters [16].

It should be noted that this procedure misses a whole group of potentially critical examples: examples that are correctly classified, however the model produces incorrect or unacceptable explanations. Until now we have not yet found a good criteria for selecting such examples.

Step 2: Presenting the critical example to the expert

In this step, a critical example with explanation from the machine learning model is presented to the domain expert. As critical examples are misclassified by the current model, the current explanation is likely wrong. Then, the domain expert is asked the following question: "Why is this example in the class as given?" The answer to this question should not contain the reasons that are mentioned in the current machine's exlanation.

Example. In one of the first applications of ABML, the goal was to distinguish between a good and a bad bishop in a chess position [13]. The learning data contained chess positions with one bishop only. Instances were described with attributes that are typically used in chess evaluation functions and each instance was classified as *bad bishop* or *good bishop*. One of the descriptive attributes was *mobility*, which counted the number of possible moves for the bishop. The algorithm initially learned that good bishops have high mobility. The first critical example was a position with a good bishop, which was blocked by a knight and was therefore not able to move (had low mobility). The expert was thus asked: "Why is the black bishop in this position good if it has low mobility?"

Step 3: The expert provides arguments for the critical example

The domain expert needs to provide at least one argument (a set of reasons) why the example's class value is as given. The argument must contain at least one reason, which was not in the original explanation provided by the machine learning method, otherwise this argument will not influence learning. If the expert can not give such an argument, we have to return to step 1 and provide another critical example.

In our previous experiments, a domain expert was unable to provide an argument due to the following two reasons. In the first case, an expert might find the example an outlier, because he or she cannot explain why the example is in this class. We can then remove the example from the data set, or, if not, prevent this example to become a critical example again. In the other case, which is also quite common, the expert discovers an error in the data. For example, it might turn out that the label of the example is wrong or that there is an error in the value of one of the descriptive attributes. Then we have to correct the error and start with another critical example.

Example. We used ABML to learn a diagnostic model for distinguishing between different types of tremor in patients with a neurological disease [7]. The patients were classified as *essential tremor* or *parkinsonian tremor* or *mixed tremor* (having both). In most cases, the expert (a physician) was able to explain critical examples. In one of the critical cases, however, the expert realized that some strong symptoms were overlooked at the time of diagnosis and, after a careful deliberation, decided to change the class value. In another critical case, the expert could not provide an argument, because the value of the attribute containing qualitative assessment of a physician had a incorrect value. After the value was corrected, the example was not critical anymore.

Step 4: Adding arguments to the critical example

A domain expert usually expresses arguments in natural language without considering domain description of the learning data set. The knowledge engineer then needs to rephrase provided arguments using domain description language (attributes).

However, expert's reasons in arguments are sometimes not covered with the current set of attributes. A domain expert often implicitly refers to an attribute missing from the current set of attributes. A knowledge engineer then needs to implement the new attribute, or change the definition of an old one. When the expert refers to unavailable attributes, which can not be added into the domain, we need to continue with another critical example.

Explaining examples with arguments has shown to be an effective tool for suggesting new attributes, since domain experts do not need to explicitly propose a set of relevant attributes, but can implicitly suggest new attributes in explanations.

Example. In the case with the bishop, the expert responded that the bishop's mobility is not limited, because the blocking knight can easily move to another square. We therefore had to redesign the mobility attribute by considering only pawns as blocking pieces. In the tremor application, the expert also suggested several new attributes. When a patient with essential tremor was selected as critical, the expert mentioned the presence of harmonics (a certain pattern in drawings of patients) as a clear signal of essential tremor. There was no attribute in the domain that would explicitly define the presence of harmonics. However we could compute a new boolean attribute (from four existing attributes) representing whether the harmonics were present or not.

Step 5: Discovering counter examples

After arguments are added to the critical example, ABML relearns the model. Arguments often apply to many other examples, and not just to the critical example, therefore these arguments will be mentioned in explanations of other examples. When these examples come from the same class as the critical example, such behavior is not problematic, it is in fact favorable, since more examples are now explained using the expert terms.

On the other hand, if the model uses positive arguments of the critical example to explain examples from the opposite class, we should check the validity of these explanation with the expert. A *counter example* is an example from the opposite class that is consistent with the positive argument provided by the expert, the induced model mentions this positive argument in the explanation of this counter example, and the inclusion of this argument in the data resulted in a higher prediction error for this example (e.g. the example is now misclassified or has a higher probabilistic error).

Example. After attaching the above argument to the bishop critical example, it turned out that high mobility is not enough, as a position with a bad and highly mobile bishop turned out as the counter example. The provided argument was consistent with the counter example (the mobility of the bishop was high), however it was from the opposite class (the bishop was bad).

Step 6: Refining arguments using counter examples

When a counter example was found, the expert needs to revise the initial arguments with respect to the counter example. The expert is now asked "Why is critical example in one class and why counter example in the other?" The expert may now revise the original argument and explain the difference between these two examples. The procedure then returns to the previous step and seeks for more counter examples.

Example. Comparing critical and counter positions in the chess example, the expert decided that the counter example had a noticeably worse pawn structure. This reason was added to the original argument of the critical example. Therefore, the initial argument (high mobility) was extended with an argument specifying good pawn structure. Afterwards, no counter examples were found.

Step 7: Pruning arguments with similar examples

In argumentation, to make an argument stronger and less susceptible to counter-arguments, humans often provide more reasons that are actually needed. In ABML, however, too many reasons will result in poor generalization.

As the last step in the discussion of the particular critical example, we should evaluate reasons in the provided argument whether they are necessary. A reason is unnecessary when its removal a) does not negatively affect the prediction accuracy of the critical example, b) does not introduce new counter-examples, and c) generalizes the argument to *similar examples*. Given a reason and an argument, a critical example is similar to another example, when they are from the same class and the argument would also apply to the similar example if it was removed. A single similar example is then shown to the expert, who needs to decide whether the same argument could be used for both examples.

Example. Although we encountered too specific arguments in almost every application of ABML, we have not yet used pruning. For example, when we tried to classify student Prolog programs as correct or incorrect [10] and the expert was asked to provide arguments for a correct critical program, he would often mention many syntactical patterns that are indicative of a correct program. After evaluating the rules that were learned from these arguments, we found out that many of the mentioned reasons were redundant. Therefore, in that application pruning would lead to a simpler and less fragmented model.

The above seven steps are repeated until the system can not find any new critical examples or some goal (such as accuracy or comprehensibility of the model) is achieved.

5 Conclusion

When a knowledge engineer is faced with a machine learning problem, she usually needs to first sit down with a domain expert and try to define the problem. As mentioned in the paper, this process is not trivial, since experts usually can not give us all the answers in advance, but an interactive process is preferred. Even with an interactive process, the communication can still be difficult, when domain experts do not understand machine learning, and knowledge engineers do not understand the domain.

In this paper, we proposed to use arguments as a communication method for bridging the gap between domain experts and machine learning. Argumentation is a skill used in everyday communication that everyone learns to a certain extent. Therefore, if machine learning results and domain experts' knowledge were represented as arguments, it would facilitate smoother communication.

We first showed how machine learning can interact with domain experts by explaining its decisions using arguments

for and against. Such explanations resemble argumentative reasoning and should thus be good enough for experts. Afterwards, we demonstrated how experts can express their knowledge by explaining particular learning examples with positive and negative arguments. The learning algorithm then uses these arguments to guide learning towards a model that is consistent with data and provided arguments. Finally, to close the loop, we described a dialogue between a domain expert and a knowledge engineer designed to drive the expert to provide useful knowledge.

When we first presented the ABML idea [14], the arguments were only used to explain learning examples. In one of the following experiments [13], we defined the ABML refinement loop, where arguments turned out to be an effective tool for elicitation of new attributes. This refinement loop was then further revised through many applications [8]. In this paper, we presented an extended version of the ABML refinement loop, where communication between a domain expert and a domain engineer comprises of several questions and arguments. This involves machine generated arguments, asking expert to give counter-arguments to machine learning arguments, and refining arguments given counter examples and similar examples.

Acknowledgement

This work was partly supported by the Slovene Agency for Research and Development (ARRS). We would also like to thank the two anonymous reviewers for valuable suggestions on this paper and colleagues from the Artificial Intelligence Laboratory, who contributed a lot in the past in the development of the ABML idea.

References

[1] Peter Clark and Robin Boswell. Rule induction with CN2: Some recent improvements. In *Machine Learning - Proceeding of the Fifth Europen Conference (EWSL-91)*, pages 151–163, Berlin, 1991.

[2] Pedro Domingos. Toward knowledge-rich data mining. *Journal of Data Mining and Knowledge Discovery*, 15:21–28, 2007.

[3] Jerry Alan Fails and Dan R. Olsen, Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 39–45, 2003.

[4] Edward A. Feigenbaum. Knowledge engineering: the applied side of artificial intelligence. In *Proc. of a symposium on Computer culture: the scientific, intellectual, and social impact of the computer*, pages 91–107, New York, NY, USA, 1984. New York Academy of Sciences.

[5] Alex A. Freitas. Comprehensible classification models - a position paper. *SIGKDD Explorations Newsletter*, 15(1):1–10, 2014.

[6] Valerio Grossi, Andrea Romei, and Franco Turini. Survey on using constraints in data mining. *Data Mining and Knowledge Discovery*, 31(2):424–464, 2017.

[7] Vida Groznik, Matej Guid, Aleksander Sadikov, Martin Možina, Dejan Georgiev, Veronika Kragelj, Samo Ribari, Zvezdan Pirtoek, and Ivan Bratko. Elicitation of neurological knowledge with argument-based machine learning. *Artificial intelligence in medicine*, 57(2):133–144, 2013.

[8] Matej Guid, Martin Možina, Vida Groznik, Aleksander Sadikov, Dejan Georgijev, Zvezdan Pirtoek, and Ivan Bratko. Abml knowledge refinement loop: A case study. In *Proceedings of the 2012 IEEE 20th International Symposium (ISMIS 2012)*, pages 41–50, 2012.

[9] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, pages 126–137, 2015.

[10] Timotej Lazar, Martin Možina, and Ivan Bratko. Automatic extraction of ast patterns for debugging student programs. In *International Conference on Artificial Intelligence in Education*, pages 162–174. Springer, 2017.

[11] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 2119–2128, 2009.

[12] Violeta Mirchevska, Mitja Lustrek, and Matjaz Gams. Combining domain knowledge and machine learning for robust fall detection. *Expert Systems*, 31:163–175, 2014.

[13] Martin Možina, Matej Guid, Jana Krivec, Aleksander Sadikov, and Ivan Bratko. Fighting knowledge acquisition bottleneck with argument based machine learning. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 234–238, 2008.

[14] Martin Možina, Jure Žabkar, and Ivan Bratko. Argument-based machine learning. *Artificial Intelligence*, 171(10/15):922–937, 2007.

[15] Martin Možina, Janez Demšar, Michael Kattan, and Blaž Zupan. Nomograms for visualization of naive bayesian classifier. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *Knowledge Discovery in Databases: PKDD*

2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004. Proceedings, pages 337–348, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[16] J. Arturo Olvera-López, J. Ariel Carrasco-Ochoa, J. Francisco Martínez-Trinidad, and Josef Kittler. A review of instance selection methods. *Artif. Intell. Rev.*, 34(2):133–143, 2010.

[17] Erik Štrumbelj and Igor Kononenko. An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.*, 11:1–18, March 2010.

[18] Simone Stumpf, Vidya Rajaram, Lida Li, Weng-Keen Wong, Margaret Burnett, Thomas Dietterich, Erin Sullivan, and Jonathan Herlocker. Interacting meaningfully with machine learning systems: Three experiments. *Int. J. Hum.-Comput. Stud.*, 67(8):639–662, 2009.

[19] Douglas Walton. *Foundamentals of Critical Argumentation; 1st edition*. Cambridge University Press, 2005.

[20] Ting Yu. *Incorporating prior domain knowledge into inductive machine learning: its implementation in contemporary capital markets.* PhD thesis, University of Technology, Sydney. Faculty of Information Technology., 2007.

An Inter-domain Study for Arousal Recognition from Physiological Signals

Martin Gjoreski[1,2], Mitja Luštrek[1] and Matjaž Gams[1,2]
[1]Department of Intelligent Systems, Jožef Stefan Institute
[2]Jožef Stefan International Postgraduate School
Ljubljana, Slovenia
E-mail: martin.gjoreski@ijs.si

Blagoj Mitrevski
Faculty of Computer Science and Engineering
Skopje, R. Macedonia

Keywords: arousal recognition, GSR, R-R, machine learning, emotion recognition, health

Received: October 27, 2017

Arousal recognition from physiological signals is a task with many challenge remaining, especially when performed in several different domains. However, the need for emotional intelligent machines increases day by day, starting with timely detection and improved management of mental disorders in mobile health, all the way to enhancing user experience in human-computer interaction (HCI). One of the open research questions, which we analyze in this paper, is which machine-learning (ML) methods and which input is most suitable for arousal recognition. We present an inter-domain study for arousal recognition on six different datasets. The datasets are processed and translated into a common spectro-temporal space of R-R intervals and Galvanic Skin Response (GSR) data, from which features are extracted and fed into ML algorithms. We present a comparison between dataset-specific models, "flat" models build on the overall data, and a novel stacking scheme, developed to utilize knowledge from all six datasets. When one model is built for each dataset, it turns out that whether the R-R, GSR, or merged features yield the best results is domain (dataset) dependent. When all datasets are merged into one and used to train and evaluate the models, the stacking scheme improved upon the results of the "flat" models.

Povzetek: Zaznavanje psihološkega vzburjenja iz fizioloških signalov je težka naloga, posebej če se je želimo lotiti na enoten način za več različnih domen. Vendar je potreba po inteligentnih strojih, ki so zmožni razumeti tudi čustva, vedno večja: uporabljajo se za različne probleme, od obvladovanja duševnih motenj z rešitvami mobilnega zdravstva do izboljševanja uporabniške izkušnje pri interakciji človeka z računalnikom. Odprto raziskovalno vprašanje, s katerim se ukvarja ta članek, je, katere metode strojnega učenja in kakšni vhodni podatki so primerni za zaznavanje vzburjenja. Članek opisuje več-domensko študijo zaznavanja vzburjenja na šestih različnih zbirkah podatkov. Zbirke so pretvorjene v enoten spektralno-časovni prostor intervalov R-R in galvanskega odziva kože, iz katerih izluščimo značilke in jih uporabimo kot vhod v algoritme strojnega učenja. Primerjamo modele, prilagojene posamičnim zbirkam podatkov, modele, zgrajene iz združenih podatkov vse zbirk, in inovativen ansambel modelov, ki takisto uporablja vseh šest zbirk. Izkaže se, da če zgradimo po en model za vsako zbirko podatkov, je od zbirke odvisno, ali se najbolje obnesejo značilke, izluščene iz intervalov R-R, galvanskega odziva kože ali obojega. Če zbirke podatkov združimo, pa se ansambel obnese bolje od navadnega modela.

1 Introduction

In 1897, Wundt [1] set the basis for modeling affective states by identifying the two emotional dimensions of calm-excitement and relaxation-tension. Almost a century later, in 1997, the field of affective computing [2] has been introduced, which aims for computational modeling of the affective states. Besides the maturity of the field of affective computing, modeling affective states has still remained a challenging task. Its importance is mainly reflected in the domain of human-computer interaction (HCI) and mobile health. In the HCI, it enables a natural and emotionally intelligent interaction. In the mobile health, it is used for timely detection and management of emotional and mental disorders such as depression, bipolar disorders and posttraumatic stress disorder. For example, the cost of work-related depression in Europe was estimated to €617 billion annually in 2013. The total was made up of costs resulting from absenteeism and presenteeism (€272 billion), loss of productivity (€242 billion), health care

costs of €63 billion and social welfare costs in the form of disability benefit payments (€39 billion) [3].

Affective states are complex states that results in psychological and physiological changes that influence behaving and thinking [5]. These psycho-physiological changes can be captured by a wearable device equipped with galvanic skin response (GSR – measures sweating rate), Electrocardiography (ECG – measures heart electrical activity) or blood volume pulse (BVP – measures cardiovascular dynamics) sensors. For example, the affective state of excitement usually initiates changes in heartbeat, breathing, sweating, and muscle tension, which can be captured using wearable sensors.

There are several approaches for modeling emotions, including discrete, continuous, and appraisal-driven approach. For the apprise-driven approach, context information is needed to model people's relationship to the environment that elicits their emotional response [4]. However, in computer science studies, the required context information is usually not available. In the discrete approach, the affect (emotion) is represented as discrete and distinct state, i.e., anger, fear, sadness, happiness, boredom, disgust and neutral. In the continuous approach, the emotions are represented in 2D (see Figure 1) or 3D space of activeness, valance and dominance [5]. Unlike the discrete approach, this model does not suffer from vague definitions and fuzzy boundaries, and has been widely used in affective studies [6] [7] [8]. The use of the same annotating model allows for an inter-study analysis.

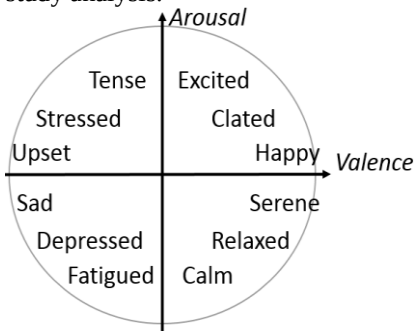


Figure 1: Circumplex model of affect. The model maps affective states in a 2D space of Arousal and Valence [5].

In this study we examine arousal recognition from GSR and heart–related physiological data, captured via: chest-worn ECG and GSR sensors, finger-worn BVP sensor, and wrist-worn GSR sensor BVP sensor. The data belongs to six publicly available datasets for affect recognition, in which there are 191 different subjects (70 females) and nearly 150 hours of arousal-labelled data. All of this introduces the problem of inter-domain learning, to which ML techniques are sensitive. To overcome this problem, we propose a preprocessing technique and a novel ML stacking scheme. The preprocessing technique translates the datasets into a common spectro-temporal space of R-R and GSR data. After the preprocessing, R-R and GSR features are extracted, which can be fed into ML algorithms to build models for arousal recognition. The novel ML stacking scheme builds dataset-specific ML models and uses a meta-learner to build general models.

The novelties of this study are:

(1) First study in affect recognition that analyzes data from six different datasets (see Section 3 Data).
(2) Methodology for translating physiological data into a common spectro-temporal space of R-R and GSR data (see Section 4.1 Pre-processing and feature extraction).
(3) Novel ML stacking scheme that generalizes from dataset-specific to general ML model for arousal recognition (see Section 4.2 Machine learning).

2 Related work

Affect recognition is an established computer-science field, but one with many remaining challenges. Many studies confirmed that affect recognition can be performed using speech analysis [10], video analysis [11], or physiological sensors in combination with ML [12]. The majority of the methods that use physiological signals use data from ECG, electroencephalogram (EEG), functional magnetic resonance imaging (fMRI), GSR, electrooculography (EOG) and/or BVP sensors.

In general, the methods based on EEG data outperform the methods based on other data [6] [7], probably due to the fact the EEG provides a more direct channel to one's mind. However, even though EEG achieves the best results, it is not applicable in normal everyday life. In contrast, affect recognition from R-R intervals or GSR data, is much more unobtrusive since this data can be extracted from ECG sensors, BVP sensors, or GSR sensors, most of which can be found in a wrist device (e.g., Empatica [13] and Microsoft Band [14]). Our methodology is tailored towards this type of data.

Regarding the typical ML approaches for affect recognition, Iacoviello et al. have combined discrete wavelet transformation, principal component analysis and support vector machine (SVM) to build a hybrid classification framework using EEG [15]. Khezri et al. used EEG combined with GSR to recognize six basic emotions via K-nearest neighbors (KNN) classifiers [16]. Mehmood and Lee used independent component analysis to extract emotional indicators from EEG, EMG, GSR, ECG and effective refractory period (ERP) [17]. Mikuckas et al. [18] presented a HCI system for emotional state recognition that uses spectro-temporal analysis only on R-R signals. More specifically, they focused on recognizing stressful states by means of the heart rate variability (HRV) analysis.

Table 1: Experimental data summary [39].

					Duration per		
Dataset	**Subjects**	**Females**	**Mean age**	**Trials**	**trial [s]**	**subject [min]**	**dataset [h]**
Ascertain	58	21	31	36	80	48.0	46.4
DEAP	32	16	26.9	40	60	40.0	21.3
Driving	10	3	35.6	1	1800	30.0	5.0
Cognitive	21	0	28	2	2400	80.0	28.0
Mahnob	30	17	26	40	80	53.3	26.7
Amigos	40	13	28	16	86	22.9	15.3
Overall	191	70	29.25	135	884.0	251.3	142.7

Regarding the more advanced ML approaches, Yin et al. [20] used an ensemble of deep classifiers for recognizing affective states using EEG, electromyography (EMG), ECG, GSR, and EOG. Using the same data, Verma et al. [19] developed an ensemble of shallow classifiers. Similarly, Kuncheva et al. [21] introduced AMBER - Advanced Multi-modal Biometric Emotion Recognition approach which uses data from EEG, EDA and HR sensor.

In contrast with the related work, which analyzes only one dataset, we perform experiments with six different datasets (domains), we analyze which ML algorithms in combination with which data type (either R-R intervals or GSR) yields best performance across all six different dataset for arousal recognition, and we propose a novel stacking method for learning from all six different domains. Finally, the work presented here is related to our previous conference paper [39]. Here we present more details regarding the data pre-processing and feature extraction, we present the novel stacking scheme and new experimental results.

3 Data

The data belongs to six publicly available datasets for affect recognition: Ascertain [6], Deap [7], Driving workload dataset [26], Cognitive load dataset [27], Mahnob [29], and Amigos [30]. Overall, nearly 150 hours of arousal-labelled data that belong to 191 subjects. Table 1 presents the data summary, which contains: number of subjects per dataset, the mean age, number of trials per subject, mean duration of each trial, duration of data per subject − in seconds, and overall duration.

Our goal was to recognize the arouse. Four datasets, Ascertain, Deap, Mahnob and Amigos, were already labelled with the subjective arousal level. One difference between these datasets was the arousal scale used for annotating. For example, the Ascertain dataset used a 7-point arousal scale, whereas the Deap dataset used a 9-point arousal scale (1 is very low, and 9 is very high, and the mean value is 5). Since the problem of arousal recognition is difficult, we decided to formulate it as a binary classification problem. From both scales, we thus split the labels in two classes using the mean value with respect the original scales. This is the same split used in the original studies. A similar step was performed for the Mahnob dataset.

Two datasets, Driving workload and Cognitive load, did not contain labels for subjective arousal level. The Driving workload dataset was labelled with subjective ratings for a workload during a driving session. For this dataset, we presume that increased workload corresponds to increased arousal. Thus, we used the workload ratings as arousal ratings. The threshold for high arousal was put on 50%. Similarly, the Cognitive load dataset was labelled for subjective stress level during stress inducing cognitive load tasks (mathematical equations). The subjective scale was from 0 to 4 (no stress, low, medium and high stress). We put the threshold for high arousal on 2 (medium stress).

4 Methods

4.1 Pre-processing and feature extraction

4.1.1 R-R data

The preprocessing is essential, since it allows merging of the six different datasets. For the heart-related data, it translates the physiological signals (ECG or BVP) to R-R intervals and performs temporal and spectral analysis. First, a peak detection algorithm is applied to detect the R-R peaks. Figure 2 presents an example for ECG signal and the detected R-R peaks. On the x-axis is the sample of the data window, on the y-axis is the output of the ECG sensor (voltage) and the detected peaks are marked with red.

Next, is temporal analysis, i.e., calculating the time distance between the detected peaks. Once the R-R intervals are detected they can be analyzed as a time series. Figure 3 is an example of an R-R time series. On the y-axis is the duration of the R-R interval, and on the x-axis is the time (in seconds) in which the R-R interval has occurred.

After the detection of R-R intervals, the R-R signal is processed. First, each R-R signal is filtered using a median filter which removes the R-R intervals that are outside of the interval [0.7*median, 1.3*median]. These parameters were determined experimentally.

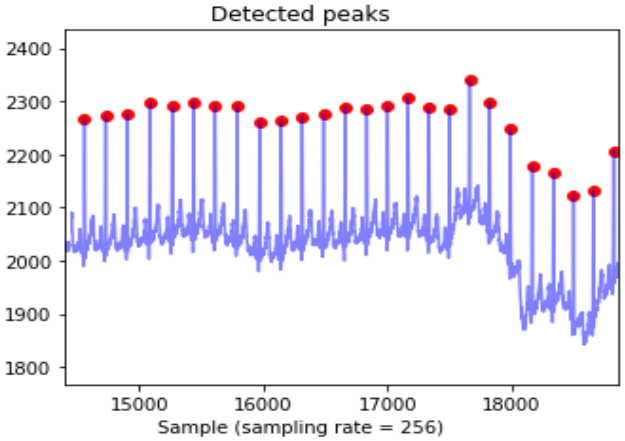


Figure 2: ECG signal and detected R-R peaks (red color). ASCERTAIN dataset t, Subject 1, Video 29 [6].

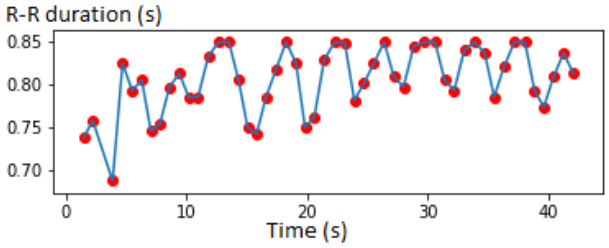


Figure 3: Example R-R signal as a time-series. ASCERTAIN dataset, Subject 1, Video 29 [6].

After the median filter, person specific winsorization is performed with the threshold parameter of 3 to remove outlier R-R intervals. From the filtered R-R signals, periodogram is calculated using the Lomb-Scargle algorithm [9]. The Lomb-Scargle algorithm allows efficient computation of a Fourier-like power spectrum estimator from unequally spaced data (as are the R-R intervals). Figure 4 presents an example Lomb-Scargle periodogram. The red color represent the low frequencies and the yellow color represents the high frequencies.

Finally, based on the related work [36], the following HRV features were calculated from the time and spectral representation of the R-R signals: the mean heart rate (meanHR), the mean of the R-R intervals (meanRR), the standard deviation of the R-R intervals (sdnn), the standard deviation of the differences between adjacent R-R intervals (sdsd), the square root of the mean of the squares of the successive differences between adjacent R-R intervals (rmssd), the percentage of the differences between adjacent R-R intervals that are greater than 20 ms, the percentage of the differences between adjacent R-R intervals that are greater than 50 ms, Poincaré plot indicies (SD1 and SD2), total spectral power of all R-R samples in power between 0.003 and 0.04 Hz (lf - low frequencies), between 0.15 and 0.4 Hz (hf − high frequencies), and the ratio of low to high frequency power.

4.1.2 GSR data

To merge the GSR data from the six datasets, several problems were addressed. Each dataset is recorded with

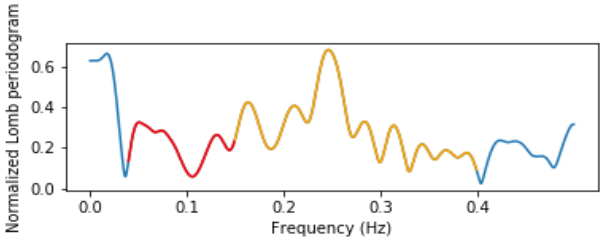


Figure 4: Normalized Lomb-Scargle periodogram calculated from R-R signal. ASCERTAIN dataset, Subject 1, Video 29 [6].

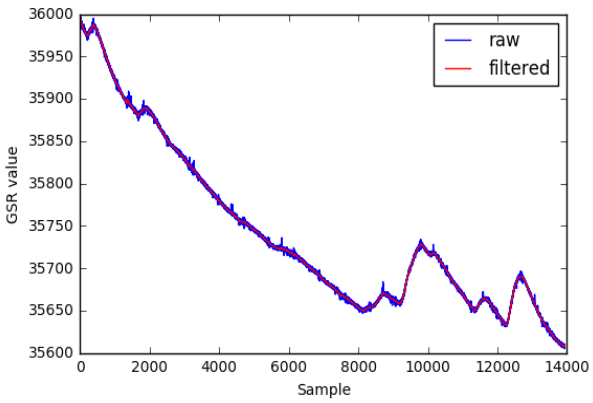


Figure 5: Filtered GSR signal. ASCERTAIN dataset, person 1, Clip 1 [6].

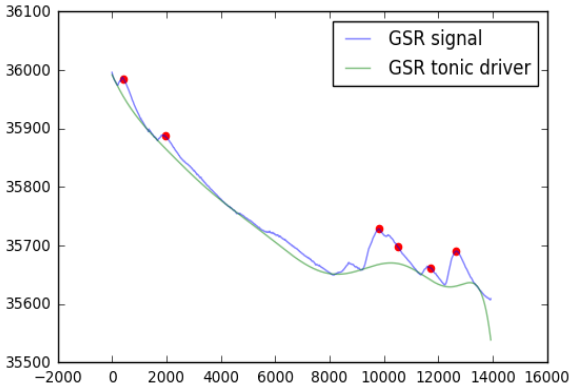


Figure 6: GSR signal decomposition (green – tonic driver, slow acting component; red – GSR responses, fast acting component). ASCERTAIN dataset, person 1, Clip 1 [6].

different GSR hardware, thus the data can be presented in different units and different scales. To address this problem, each GSR signal was converted to μS (micro Siemens). Next, the GSR signal was filtered using a lowpass filter with a cut-off frequency of 1 Hz. Figure 5 presents an example filtered GSR signal. To address the inter-participant variability of the signal, person-specific min-max normalization was performed, i.e., each signal was scaled to [0, 1] using person specific winsorized minimum and maximum values. The winsorization parameter was set to 3.

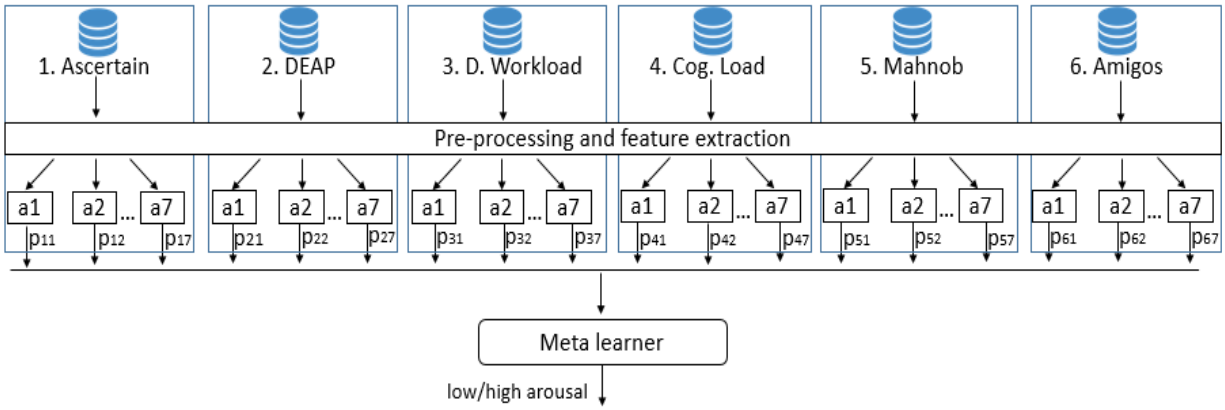


Figure 7: The novel stacking scheme for training a meta-learner that utilizes knowledge from all six datasets.

Finally, the fast acting component (GSR responses) and the slow acting component (tonic component) were determined in the signal using the "peakutils.baseline" function from the Python's PeakUtils library. The function is used with the default parameters. It iteratively performs a polynomial fitting in the data to detect its baseline. For example, in Figure 6, the GSR responses are marked with red and the tonic component (baseline) is marked with green. Based on the related work [30], the preprocessed GSR signal was used to calculate GSR features: mean, standard deviation, 1st and 3rd quartile (25th and 75th percentile), quartile deviation, derivative of the signal, sum of the signal, number of responses in the signal, rate of responses in the signal, sum of the responses, sum of positive derivative, proportion of positive derivative, derivative of the tonic component of the signal, difference between the tonic component and the overall signal.

4.2 Machine learning

4.2.1 Flat machine learning

After the feature extraction, the data is in a format which can be input for typical ML algorithms. Models were built using seven different ML algorithms: Random Forest, Support Vector Machine, Gradient Boosting Classifier, AdaBoost Classifier (with a Decision Tree as a base classifier), KNN Classifier, Gaussian Naive Bayes and Decision Tree Classifier. The algorithms were used as implemented in the Scikitlearn, the Python ML library [37]. For each algorithm, a randomized search on hyper parameters was performed on the training data using 2-fold cross-validation.

4.2.2 Stacking

The novel stacking scheme, depicted in Figure 7, was designed to train a meta-learner which would utilize the knowledge from all six datasets. In the example scenario, we used the 7 ML algorithms mentioned in the previous section. Thus, there are 42 base models (6 datasets x 7 ML algorithms). The outputs of the base models, which are probabilities for the class "high arousal", are used as input to a meta-learner. The meta-learner can be any ML algorithm previously mentioned. We experimentally chose Random Forest to be our meta-learner. The meta-learner is trained using a 10 fold-cross validation on the training data. That is, the base learners are trained on 90% of the data, then predictions are provided on the rest 10% of the data, and this procedure is repeated ten times. Finally, the meta-learner is trained on the cross-validated predictions of the base learners. In the test phase, the test instances are provided as input to all of the 42 base models, their output is summed up in a 42 dimensional vector (in Figure 7 marked as p_{11}, p_{12},…,p_{67} – six datasets and seven base models) as input to the meta-learner, which provides the final prediction for the test instance.

5 Experimental results

Two types of experiments were performed: dataset specific experiments, and experiments with merged datasets. The dataset-specific experiments were used to identify the ML algorithm and the input that would yield the best performance per dataset.

The experiments on the merged datasets were used to build general, dataset-independent ML models. This evaluation simulates a scenario where the source (dataset) is unknown, i.e., we do not know whether the subject is watching an affective video (e.g., the DEAP dataset), is driving a car (e.g., the Driving workload dataset) or he/she is working on a cognitive demanding task (e.g., the cognitive load dataset).

The evaluation was performed using trial-specific 10-fold cross-validation, i.e., the data segments that belong to one trial (e.g., one affective stimuli), can either belong only to the training set or only to the test set, thus there was no overlapping between the training and test data.

5.1 Dataset specific

The results for the dataset-specific experiments are presented in Table 2. The first column represents the ML algorithm, the second column represents the features used as input to the algorithm (R-R, GSR or Merged - M) and the rest of the columns represent the dataset which is used for training and evaluation using the trial-dependent 10-fold cross-validation. We report the mean accuracy ± the standard evaluation for the 10 folds. For each dataset, the best performing model(s) is (are) marked with green.

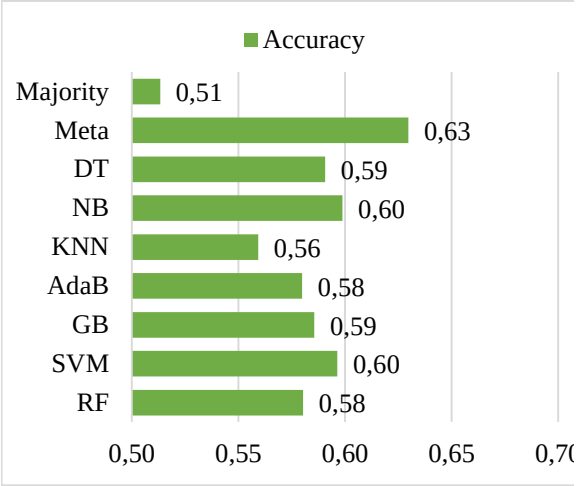


Figure 8: Accuracy of the meta-learner and the "flat" approaches for the merged-datasets experiments.

For example, on the Ascertain and the Driving workload dataset, the best performing algorithm is the SVM, on the Deap dataset, the best performing algorithm is the RF, on the Cognitive Load and the Mahnob datasets, the best performing is the NB, and on the Amigos dataset, the best performing is the AdaBoost algorithm.

When we compare which input (R-R features, GSR features or Merged-M) provide better accuracy, on two datasets, the Asceratin and the Driving workload, the results are the same, on the Deap dataset, the R-R features provide better results, on the Cognitive Load dataset, the highest accuracy is achieved both for the GSR and the Merged features, on the Mahnob dataset, the GSR features provide best accuracy and on the Amigos dataset, the Merged features.

Regarding the majority class, the biggest accuracy improvement was achieved for the Cognitive load dataset, which is an improvement of 9 percentage points. For the two datasets, the Deap and the Amigos, the improvement was 2-3 percentage points, and for the three datasets, the Ascertain, the Driving workload and the Mahnob, the best performing models were as good as the majority classifier.

5.2 Merged datasets

In the dataset-specific experiments, none of the algorithms yielded best performance (compared to the rest of the algorithms) over all datasets, thus there was no experimental hint about which algorithm would be able to generalize over all datasets. For that reason, we came up with the stacking approach, where a meta-learner learns how to combine the output of all of the algorithms trained on the different datasets. The details are presented in section 4.2. Stacking. The input to the algorithms was the merged feature set, i.e., R-R and GSR features.

We compared the meta-learning approach to a simple approach where the "flat" ML algorithms are trained on all datasets merged. The evaluation is performed using the same trial-specific 10-fold cross-validation. The results are presented in Figure 8. It can be seen that all of the "flat" algorithms achieved an accuracy bellow or equal to 60%. The meta-learning approach slightly improved the results by achieving an

Table 2: Dataset-specific experimental results. Mean accuracy ± stdDev for trial-specific 10-fold cross validation. The best performing models per dataset are marked with green [39].

Algorithm	Features	Dataset					
		Ascertain	Deap	D. Workload	Cog. Load	Mahnob	Amigos
RF	R-R	0.655 ± 0.07	0.556 ± 0.03	0.785 ± 0.24	0.739 ± 0.13	0.580 ± 0.11	0.536 ± 0.06
	GSR	0.638 ± 0.06	0.503 ± 0.04	0.780 ± 0.24	0.763 ± 0.12	0.611 ± 0.07	0.473 ± 0.11
	M	0.653 ± 0.05	0.540 ± 0.04	0.785 ± 0.25	0.755 ± 0.13	0.611 ± 0.10	0.559 ± 0.10
SVM	R-R	0.664 ± 0.07	0.536 ± 0.05	0.795 ± 0.26	0.717 ± 0.21	0.623 ± 0.15	0.521 ± 0.24
	GSR	0.664 ± 0.07	0.525 ± 0.05	0.795 ± 0.26	0.712 ± 0.20	0.588 ± 0.10	0.470 ± 0.12
	M	0.664 ± 0.07	0.513 ± 0.03	0.795 ± 0.26	0.691 ± 0.18	0.623 ± 0.15	0.506 ± 0.13
GB	R-R	0.649 ± 0.07	0.554 ± 0.03	0.785 ± 0.20	0.736 ± 0.15	0.578 ± 0.11	0.543 ± 0.06
	GSR	0.642 ± 0.05	0.500 ± 0.04	0.800 ± 0.21	0.743 ± 0.12	0.609 ± 0.08	0.527 ± 0.09
	M	0.644 ± 0.05	0.533 ± 0.03	0.755 ± 0.23	0.761 ± 0.15	0.609 ± 0.11	0.542 ± 0.09
AdaB	R-R	0.658 ± 0.06	0.532 ± 0.02	0.750 ± 0.23	0.718 ± 0.13	0.580 ± 0.09	0.531 ± 0.07
	GSR	0.633 ± 0.05	0.485 ± 0.03	0.750 ± 0.22	0.740 ± 0.13	0.589 ± 0.08	0.514 ± 0.09
	M	0.623 ± 0.05	0.526 ± 0.03	0.755 ± 0.22	0.766 ± 0.16	0.610 ± 0.08	0.560 ± 0.08
KNN	R-R	0.625 ± 0.05	0.509 ± 0.02	0.710 ± 0.19	0.715 ± 0.13	0.582 ± 0.07	0.509 ± 0.05
	GSR	0.590 ± 0.06	0.496 ± 0.04	0.795 ± 0.26	0.772 ± 0.09	0.605 ± 0.06	0.533 ± 0.08
	M	0.600 ± 0.05	0.490 ± 0.02	0.750 ± 0.23	0.770 ± 0.13	0.601 ± 0.09	0.533 ± 0.06
NB	R-R	0.654 ± 0.07	0.537 ± 0.04	0.735 ± 0.15	0.748 ± 0.15	0.574 ± 0.06	0.485 ± 0.09
	GSR	0.602 ± 0.04	0.537 ± 0.05	0.540 ± 0.22	0.803 ± 0.09	0.624 ± 0.07	0.454 ± 0.10
	M	0.591 ± 0.04	0.535 ± 0.06	0.665 ± 0.17	0.804 ± 0.12	0.592 ± 0.06	0.486 ± 0.09
DT	R-R	0.664 ± 0.07	0.519 ± 0.05	0.685 ± 0.17	0.736 ± 0.15	0.597 ± 0.09	0.505 ± 0.06
	GSR	0.640 ± 0.05	0.542 ± 0.05	0.765 ± 0.22	0.734 ± 0.08	0.583 ± 0.09	0.483 ± 0.11
	M	0.650 ± 0.05	0.524 ± 0.04	0.615 ± 0.22	0.704 ± 0.09	0.581 ± 0.13	0.551 ± 0.09
Majority		0.664	0.536	0.795	0.717	0.623	0.521

accuracy of 63%.

6 Conclusion and discussion

We presented an inter-domain study for arousal recognition on six different datasets, recorded with twelve different hardware sensors. We experimented with dataset-specific models, general models built on the overall (merged) data and general models build using the novel stacking scheme. For the dataset-specific models, we compared the results of seven different ML algorithms, using three different feature inputs (R-R, GSR or Merged – M features). For the models built on the overall (merged) data, we compared the results of the novel stacking scheme and "flat" ML models. The results on the dataset-specific setup showed that, out of the seven ML algorithms tested, none yields the best performance on all datasets. In addition to that, a clear conclusion cannot be made whether the R-R, GSR or the Merged features yield the best results – this is domain (dataset) dependent.

On the merged-datasets experiments, the novel stacking scheme slightly outperformed the "flat" models. This was expected since the stacking scheme utilizes seven different ML models built on the six different datasets, thus 42 different models (views).

However, the experimental results show that there is room for improvement regarding the accuracy achieved in both types of experiments. In the future, we plan to investigate more advanced techniques such as deep neural networks and transfer learning, which might be able to learn more accurate models that will be able to generalize across different domains. Finally, once we find the best performing scenario, we will generalize the method for arousal recognition to a method for valence recognition and method for discrete emotion recognition.

7 References

[1] W. Wundt. Outlines of psychology (C. H. Judd, Trans.). Oxford, UK: Engelman, 1897.

[2] R. Picard. Affective Computing. Cambridge, MA: MIT Press, 1997.

[3] Depression cost: http://ec.europa.eu/health/sites/health/files/mental_health/docs/matrix_economic_analysis_mh_promotion_en.pdf, [Accessed 27.03.2017].

[4] S. Marsella, J. Gratch. Computationally modeling human emotion. Commun. ACM 57, 12 (November 2014), pp. 56-67. 2014.

[5] J. A. Russell. A circumplex model of affect. Journal of Personality and Social Psychology, 1980.

[6] R. Subramanian, J. Wache, M. Abadi, R. Vieriu, S. Winkler, N Sebe. ASCERTAIN: Emotion and Personality Recognition using Commercial Sensors. IEEE Transactions on Affective Computing. 2016.

[7] S. Koelstra, C. Muehl, M. Soleymani, J.-S. Lee, A. Yazdani, T. Ebrahimi, T. Pun, A. Nijholt, I. Patras. DEAP: A Database for Emotion Analysis using Physiological Signals (PDF). IEEE Transaction on Affective Computing, 2012.

[8] M.K. Abadi, R. Subramanian, S. M. Kia, P. Avesani, I. Patras. Nicu Sebe. DECAF: MEG-Based Multimodal Database for Decoding Affective Physiological Responses. IEEE Transactions on Affective Computing, 2015.

[9] N.R. Lomb. Least-squares frequency analysis of unequally spaced data. Astrophysics and Space Science, vol 39, pp. 447-462, 1976

[10] Z. Zeng, M. Pantic, G.I. Roisman, T.S. Huang.A survey of affect recognition methods: audio, visual, and spontaneous expressions. Transactions on Pattern Analysis and Machine Intelligence, 2009.

[11] I. Abdic, L. Fridman, D. McDuff, E. Marchi, B. Reimer, Schuller, B. Driver Frustration Detection From Audio and Video. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'16), 2016.

[12] S. Jerritta, M. Murugappan, R, Nagarajan, K. Khairunizam. Physiological Signals Based Human Emotion Recognition: A Review. International Colloquium on Signal Processing and its Applications. 2011.

[13] M. Garbarino, M. Lai, D. Bender, R. W. Picard, S. Tognett. Empatica E3 - A wearable wireless multi-sensor device for real-time computerized biofeedback and data acquisition. 4th International Conference on Wireless Mobile Communication and Healthcare, pp. 3-6, 2014.

[14] Microsoft band. https://www.microsoft.com/microsoft-band/en-us

[15] D. Iacovielloa, A. Petraccab, M. Spezialettib, G. Placidib. A real-time classification algorithm for EEG-based BCI driven by self-induced emotions. Computer Methods and Programs in Biomedicine, 2015.

[16] M. Khezria, M.Firoozabadib, A. R. Sharafata. Reliable emotion recognition system based on dynamic adaptive fusion of forehead biopotentials and physiological signals.

[17] R. M. Mehmooda, H. J. Leea. A novel feature extraction method based on late positive potential for emotion recognition in human brain signal patterns. Computers & Electrical Engineering, 2016.

[18] A. Mikuckas, I. Mikuckiene, A. Venckauskas, E. Kazanavicius2, R. Lukas2, I. Plauska. Emotion Recognition in Human Computer Interaction Systems. Elektronika Ir Elektrotechnika, Reserarch Journal, Kaunas University of Technology, 2014.

[19] G. K. Verma, U. S. Tiwary. Multimodal fusion framework: A multiresolution approach for emotion classification and recognition from physiological signals. NeuroImage, 2014.

[20] Z. Yin, M. Zhao, Y. Wang, J. Yang, J Zhang. Recognition of emotions using multimodal physiological signals and an ensemble deep learning model. Computer Methods and Programs in Biomedicine, pp. 93-110, 2017.

[21] L. I. Kuncheva, T. Christy, I. Pierce, Sa'ad P. Mansoor. Multi-modal Biometric Emotion Recognition Using Classifier Ensembles.

Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, 2011.
[22] Wei Liu, Wei-Long Zheng, Bao-Liang Lu. Multimodal Emotion Recognition Using Multimodal Deep Learning. Online. Available at: https://arxiv.org/abs/1602.08225, 2016.
[23] W-L. Zheng, B-L Lu. A multimodal approach to estimating vigilance using EEG and forehead EOG. Journal of Neural Engineering, 2017.
[24] Z. Yin, M. Zhao, Y. Wang, J. Yang, J. Zhang. Recognition of emotions using multimodal physiological signals and an ensemble deep learning model. Comput Methods Programs Biomed. 2017.
[25] K.Weiss, T. M. Khoshgoftaar, D. Wang. A survey of transfer learning. Journal of Big Data, 2016.
[26] S. Schneegass, B. Pfleging, N. Broy, A. Schmidt, Frederik Heinrich. A Data Set of Real World Driving to Assess Driver Workload. 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, 2013.
[27] M. Gjoreski, M. Luštrek, M. Gams, H. Gjoreski. Monitoring stress with a wrist device using context. Journal of Biomedical Informatics, 2017, in press.
[28] M. Gjoreski, H. Gjoreski, M. Luštrek, M. Gams. Continuous stress detection using a wrist device: in laboratory and real life. ACM Conf. on Ubiquitous Computing, Workshop on mentalhealth, pp. 1185-1193, 2016.
[29] M. Soleymani, T.Pun. A Multimodal Database for Affect Recognition and Implicit Tagging, IEEE Transactions On Affective Computing, 2012.
[30] J. A. Miranda-Correa, M. Khomami Abadi, N. Sebe, I. Patras. AMIGOS: A Dataset for Affect, Personality and Mood Research on Individuals and Groups. Transactions On Affective Computing, 2017.
[31] L. H. Negri. Peak detection algorithm. Python Implementation. Online. Available at: http://pythonhosted.org/PeakUtils/.
[32] M. Wu, PhD thesis. Michigan State University; 2006. Trimmed and Winsorized Eestimators.
[33] J.D. Scargle. Studies in astronomical time series analysis. II - Statistical aspects of spectral analysis of unevenly spaced data. The Astrophysical Journal, vol 263, pp. 835-853, 1982.
[34] D. P. Kingma, J. Ba. Adam: A Method for Stochastic Optimization, http://arxiv.org/abs/1412.6980, 2014.
[35] Tensorflow. Online. Available at: https://www.tensorflow.org/
[36] R. Castaldoa, P. Melillob, U. Bracalec, M. Casertaa,c, M. Triassic, L. Pecchiaa. Acute mental stress assessment via short term HRV analysis in healthy adults: A systematic review with meta-analysis. Biomedical Signal Processing and Control. 2015.
[37] Scikit-learn, Python machine-learning library http://scikit-learn.org/dev/_downloads/scikit-learn-docs.pdf
[38] L.J.P, van der Maaten., G.E. Hinton. Visualizing High-Dimensional Data Using t-SNE. Journal of Machine Learning Research. 9: 2579–2605, 2008.
[39] M. Gjoreski, B. Mitrevski, Mitja Luštrek, Matjaž Gams. R-R vs GSR – An inter-domain study for arousal recognition, Multiconference Information Society, Ljubljana, 2017.
[40] Python library for signal analysis: http://pythonhosted.org/PeakUtils/

Computational Creativity in Slovenia

Senja Pollak[1], Geraint A. Wiggins[2,3], Martin Žnidaršič and Nada Lavrač[1,4]
[1] Department of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, Slovenia
[2] Computational Creativity Lab, Queen Mary University of London, London E1 4NS, UK
[3] AI Lab, Free University of Brussels, Brussels 1050, Belgium
[4] University of Nova Gorica, Nova Gorica, Slovenia
E-mail: {senja.pollak,martin.znidarsic,nada.lavrac}@ijs.si; geraint.wiggins@qmul.ac.uk

Keywords: computational creativity, bisociative reasoning, computational creativity platform

Received: November 6, 2017

Computational Creativity is a field of Artificial Intelligence that addresses processes that would be deemed creative if performed by a human. The field has been very active since 1999, and is now an established research field with its own International Conference on Computational Creativity (ICCC) conference series founded in 2010. This paper briefly surveys the field of Computational Creativity (CC) that is based on the analysis of ICCC conference papers, followed by a more detailed presentation of projects and selected contributions of Slovenian researchers to the field.

Povzetek: Računalniška ustvarjalnost je področje umetne inteligence, ki obravnava procese, ki bi jih ocenili kot kreativne, če bi jih izvajal človek. Področje računalniške ustvarjalnosti se je razmahnilo po letu 1999, kot veja znanosti pa se je uveljavilo leta 2010 z ustanovitvijo serije letnih konferenc z imenom International Conference on Computational Creativity (ICCC). V članku podamo kratek pregled področja računalniške ustvarjalnosti, ki temelji na analizi ICCC konferenčnih člankov, posebno pozornost pa namenimo predstavitvi projektov in izbranih dosežkov slovenskih raziskovalcev.

1 Introduction

As a sub-field of Artificial Intelligence (AI) research, *Computational Creativity*[1] (CC) is concerned with machines that exhibit behaviours that might reasonably be deemed creative [49; 11]. Slovenian researchers have made important contributions to CC. This paper aims to provide an objective snapshot of the field of computational creativity as a whole, and to give a brief summary of the particular contribution of Slovenian researchers to it.

In the next section, we summarise an analysis of the research field, that we conducted in 2016, using it to structure a brief introduction to the field for unfamiliar readers. We then summarise the contributions of Slovenian researchers to CC.

2 Domain understanding

We here summarise the results of a study of the research field of Computational Creativity [36], which was based on the analysis of papers published in the Proceedings of the International Conference on Computational Creativity (ICCC)[2]. The aim of the study was to objectively identify areas of interest in this research field. Here, we use its conclusions to motivate our subsequent outline of CC research.

In the previous study, Pollak et al. [36] used semi-automatic topic ontology generation tool OntoGen [16] to explore the texts of the complete conference proceedings of the International Conference on Computational Creativity to date. This allowed them to make an objective, explainable bottom-up analysis of the field.

The input to the OntoGen tool are documents, which are texts of individual articles from the proceedings. After manual text cleaning and removal of the papers' reference sections, OntoGen performs stemming and stop word removal, followed by the construction of Bag-of-Words (BoW) feature vector representations of documents, where the features are weighted by the TF·IDF heuristic [41] and used for clustering. The user may explore the results, and identify hierarchies of significant terms and clusters of documents. The keywords are identified by OntoGen in two ways: *descriptive* keywords are extracted from document centroid vectors, while *distinctive* keywords are extracted from the SVM classification model distinguishing the documents in the given topic (document cluster) from the documents neighbouring clusters [16]. Other functionalities used were expert's manual moving of documents between clusters to reduce inappropriate paper categorisation and active learning of selected concepts/categories.

Several outputs were presented by Pollak et al. [36], including understanding of the field of Computational Creativity based on its topics, which is also of interest to this study.

A final corpus-based categorisation of the field of com-

[1] http://computationalcreativity.net
[2] http://computationalcreativity.net/home/conferences/

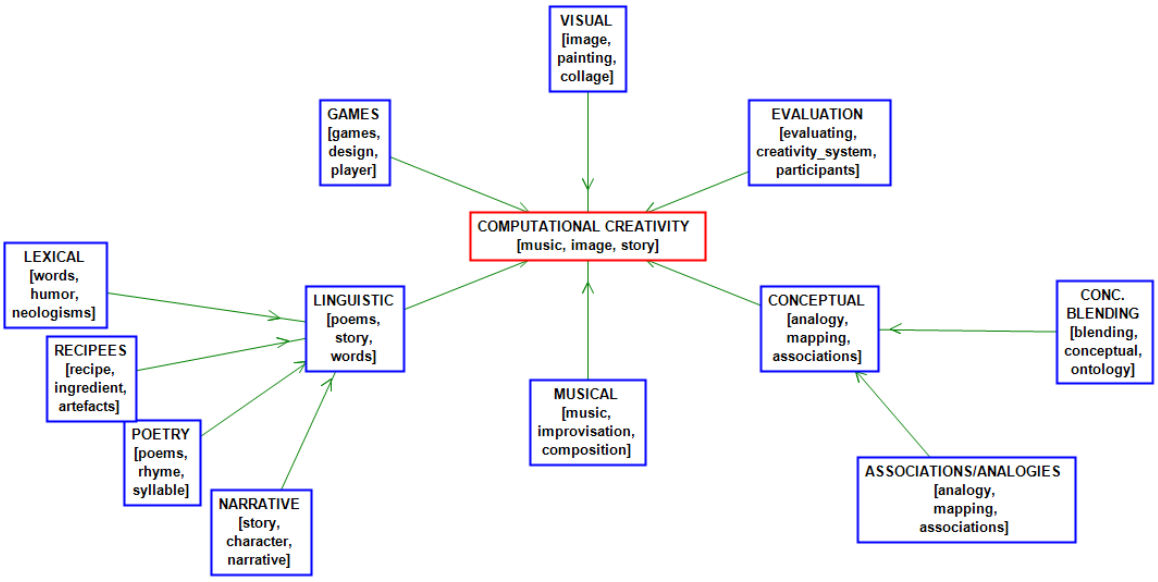


Figure 1: Semi-automatically generated conceptualization of the CC domain, with CC concept naming and sub-concept creation.

putational creativity is presented in Figure 1. The main sub-domains of computational creativity identified by our method were: Musical, Visual, Linguistic creativity, Games and creativity, Conceptual creativity as well as newly created category of Evaluation. For several domains, subcategories were detected also at lower levels, including Narratives, Poetry, Recipes and Lexical creativity as subdomains of Linguistic creativity. Each category can be further characterised through descriptive keywords listed in Table 1, as extracted from cluster centroid vectors.

3 Brief review of computational creativity

We now present a brief overview of Computational Creativity, as represented by the domains identified by Pollak et al. [36]. We have added in an additional category, *scientific creativity*, on the grounds that important work in this area was performed prior to the inception of the ICCC conference, and was therefore not represented in the conferences.

In this position paper we do not present a detailed review of the field but explain the key issues and cite some successful exemplars of CC research. A recurring general theme of ICCC is the attempt to better understand what is meant by term "creativity." Early on, it was recognised that we must move away from Romantic notions of "great" creativity, if we are to make progress. So ICCC is interested in *creative process* more than *creative output*, and there is no acceptance of the notion of "inspiration", understood as mystical intervention by some agency extrinsic to the creator. Of course, in a paper such as this, one cannot discuss process without reference to outputs, without being interminably dull. For this reason, we include examples where possible.

Boden [1, 2] first formally raised the question of creativity in AI, but there have been significant precursors of CC field in several domains that are also mentioned here.

3.1 Visual creativity

Most work on visual creativity is conceptualised in terms of painting or drawing. In this domain, there tends to be a focus on painting technique and on the objects produced.

The clear forerunner of CC in this domain was Harold Cohen, a successful artist in his own right, who built a robot painter, AARON[3], programmed in a rule-based style. Its development began in the 1970s, with developments right up to the artist's death in 2016 [26; 2]. Cohen viewed AARON as a part of his art, and therefore did not always disclose the methods used to make it work, though he did write several papers on some aspects of the system [e.g., 7; 6; 5]. Figure 2(a) shows a well-known painting by AARON.

Simon Colton's *The Painting Fool*[4] deconstructs painting from subject composition (for example, collage based on stories from The Guardian newspaper) right down to brush stroke [9]. Figure 2(b) shows an example.

DARCI[5] [27], unusually, is multi-modal and can explain itself: it combines image processing with language comprehension, so as to focus the system on the extraction and generation of meaning. DARCI produced the image in Fi-

[3] www.aaronshome.come
[4] http://www.thepaintingfool.com
[5] http://darci.cs.byu.edu

Table 1: Categories and keywords of the first layer of the semi-automatically constructed CC ontology.

Category	Automatically extracted keywords
Musical	music, chord, improvisation, melodies, harmonize, composition, accompaniment, pitch, emotions, beat
Visual	image, painting, darci, artifacts, collage, adjectives, associations, rendered, colored, artists
Linguistic	story, poems, actions, character, words, agents, narrative, artefacts, poetry, evaluating
Games	games, design, player, games_design, angelina, agents, code, jam, filter, gameplay
Conceptual	analogy, blending, mapping, conceptual, objective, associations, team, graphs, concepts, domain
Evaluation	music, poems, improvisation, evaluating, interactive, poetry, creativity system, musician, participants, behavioural
Comp. creativity	music, image, story, games, agents, words, actions, poems, character, blending

gure 2(c), explaining it as follows (there is not space here for the intermediate images): "I looked at this picture, [an elephant walking across a verdant African plain] and it reminded me of this image that I've seen before, [a standing stone] which is a picture of a stone. The picture also seemed gloomy and brooding. So I created this initial sketch, [black and white graphic drawing] and then rendered it in a style related to stone, gloomy, and brooding, which resulted in this image. [intermediate image] It turned out more like a bucket or a cauldron, and it seems creepy, but I'm happy with it."

3.2 Creative game design

Computational creativity has many applications in games, perhaps most obviously in the area of game level generation, where the landscape and structure of a game are created live. However, probably the most unexpected and interesting example of CC in games is *Yavalath*[6] [3], ranked in the top 100 board games ever invented by the BoardGameGeek website. It is highly novel in that the board is hexagonal.

Another success has been *Angelina*[7] [12], a long term project aiming for completely computational creativity of digital games.

3.3 Linguistic creativity

Creativity in language covers a broad area, including poetry and story-telling. Two systems that demonstrate different approaches are MEXICA [31] and Propper [17]. MEXICA uses a general creative method, the Engagement-Reflection model, to model a two-phase, cyclic approach to creativity. Propper takes a contrasting approach, using heuristics from literary theory [38] to guide exploratory creativity. A third successful approach is that of Tony Veale [46]. Veale's lab specialises in the development of elegant methods of extracting data from lingustic corpora, and then using that data for creative text generation, often in TwitterBots—see `@MetaphorMagnet` [e.g., 47].

[6] `http://www.cameronius.com/games/yavalath/`
[7] `http://www.gamesbyangelina.org`

3.4 Musical creativity

Musical creativity had important precursors too. David Cope's EMI [13; 50] produced many compositions, but none of the reports on the work made it clear what the system actually did, and how much was due to its author. A clearer early contribution, with full scientific reporting, was by [14], which produced musical harmony in the style of J. S. Bach. This is a remarkable contribution, and still stands today as an excellent piece of work; its fault is that its harmonisations sound *too much* like Bach—the system does not reflect on its overall balance, but applies Bachian compositional tricks everywhere.

Perhaps the first attempt at automated composition really to situate itself in CC was the work of [28]. Melodies were generated from a learned model of style, and evaluated in detail by expert musicians [29].

François Pachet's team has produced the most thorough CC music systems to date, working from chords and melodies right through to studio production [40].

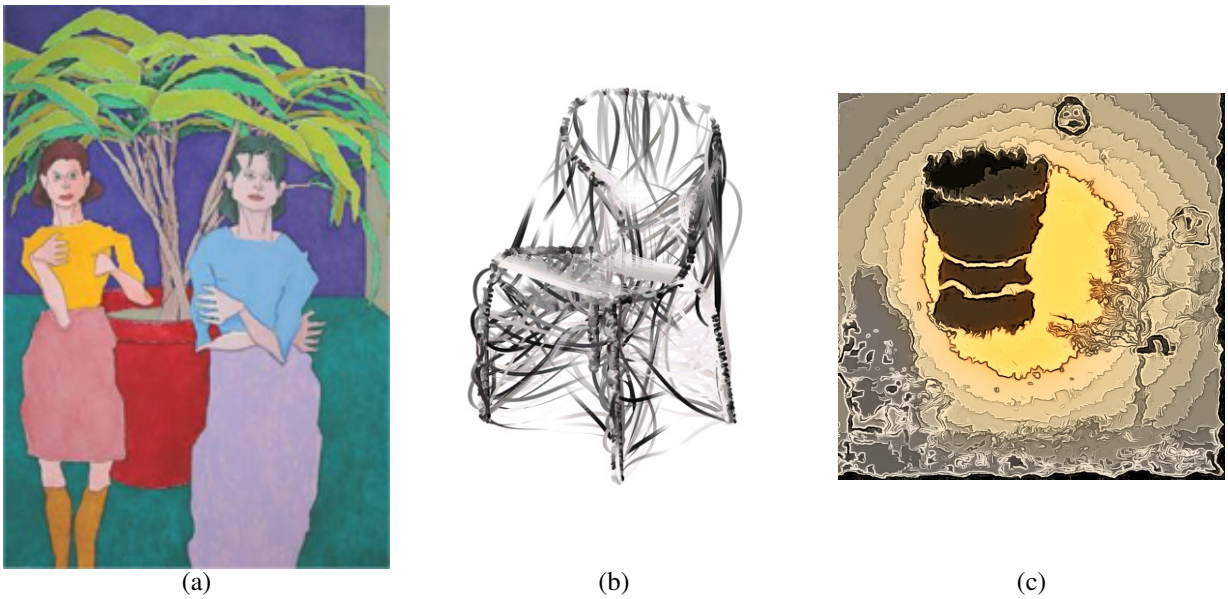
3.5 Scientific creativity

It is often forgotten that human creativity is evident in science and engineering, as well as in the arts and humanities. One of the earliest successes in CC was the HR system of Colton [8]. This was an exploratory creativity system, which invented new integer sequences with properties that mathematicians find interesting; 17 of the sequences it dicovered were novel and interesting enough to be included in the Journal of Integer Sequences, which records these structures and acts as an encyclopedia of them. It also made conjectures about some of these sequences that were subsequently proven correct.

Another successful project in scientific creativity was funded by the EU FP7 programme: BISON studied the application of *bisociative reasoning* [20] to medical text analysis (see Section 4.1).

3.6 Concept creation

Concept creation arises as a separate category in the objective analysis because it is central to all creative domains.



(a) (b) (c)

Figure 2: Three computationally created images. (a) *Untitled* from AARON's middle period output. (b) The Painting Fool's *Uneasy*. (c) DARCI's *Always Be A Gloomy Cauldron, Even in Creepy Stone*.

There are too many approaches to survey here; however, a recurring theme is conceptual blending [44], which has been carried forward with some success. An example is the *Divago* system [30], a computational model that uses conceptual blending. The key idea here is somewhat similar to Koestler's *bisociation* [20]: new concepts are created from combinations of features of existing and/or imagined ones. A recent EU FP7 project, ConCreTe, focused on Concept Creation Technology (see Section 4.3).

3.7 Creative systems evaluation

Evaluation is a particularly difficult problem in comptuational creativity, which attracts commensurate attention in the literature. There are two distinct ways that computationally creative systems involve evaluation: first, in the conventional scientific sense, where the correctness and value of work is assessed; and, second, in the sense of *reflection* within the system, that allows it to make intelligent creative decisions. Quite often, but certainly not always, these two aims coincide.

The value of a creative act is a function of four aspects [51]: Context, Observer, Creator and Artefact, forming the acronym COCA. But this does not give detail of how creativity might actually be assessed. Ritchie [39] gives a detailed set of criteria that can be used to assess the creativity of a computer program, which have been used in several projects. Jordanous [e.g., 18] and van der Velde et al. [e.g, 45] have made substantial contributions in this area.

4 Computational creativity in Slovenia

To the best of our knowledge, the only Computational Creativity research in Slovenia has been performed by the members of Department of Knowledge Technologies at Jožef Stefan Institute (JSI) in Ljubljana. Most of the research, including the work summarised in Section 2, has taken place within three distinct EU-funded projects and the PROSECCO networking action, all supported by the European FP7 funding programme. We summarise this work, with a special focus on Slovenian contributions.

4.1 Bisociation networks for creative information discovery (BISON)

BISON[8] was a research project from the field of scientific creativity, which deals with the bisociation-based scientific knowledge discovery. Arthur Koestler [20] argued that the essence of creativity lies in "perceiving of a situation or idea ... in two self-consistent but habitually incompatible frames of reference", and introduced the expression *bisociation* to characterise this creative act. The key vision of the BISON project was to develop a fundamentally new ICT paradigm for bisociative information discovery. JSI's main contributions were related to scientific literature mining aimed at creatively forming new hypotheses based on yet uncovered relations between knowledge from different, relatively isolated fields of specialization. We developed CrossBee[9], a literature-based discovery support tool [19], where different elementary and ensemble heuristics

[8] http://cordis.europa.eu/project/rcn/86374_en.html

[9] http://crossbee.ijs.si/

are implemented for bisociative bridging term (b-term) discovery. The heuristics are defined as functions that numerically evaluate the term quality by assigning it a bisociation score (measuring the potential that a term is actually a b-term). Other methodologies developed for cross-domain literature based discovery focus on exploration of outlier documents [34; 42]. JSI's methods were tested on standard datasets (e.g., migraine-magnesium studied in early research by [43], but also actually led to new hypotheses in understanding autism [23] and Alzheimer's disease [4].

4.2 The What-If Machine (WHIM)

The WHIM project was concerned with the automated generation, understanding and evaluation of fictional ideas. Fictional ideas are propositions of situations that are unrealistic or commonly considered as unplausible, such as: "What if there was a little fish who couldn't swim?" which are a central part of various creative works and products. Artificial production of What-if ideas is creative work that is inherently hard to automate, but there are now some generators available (e.g., [22]). In the generation process, there is usually a trade-off between a template driven process (with a relatively narrow covering of the fictional ideation space) or more open and autonomous generative process (producing more interesting and valuable ideas, but larger amount of lower quality results).

The What-if Machine was also the inspiration for a real musical show Beyond The Fence, billed as "the world's first computer-generated musical", that performed in London in 2016. In this artistic project—containing the musical and a documentary—several computational creativity research prototypes were combined and used in the artistic process [10].

JSI's main role in the WHIM project was in automated modelling of human evaluations. The main tasks included the design of a large crowd-sourcing data gathering exercise, resulting in more than 10,000 evaluated fictional ideas and next, to build data mining models, which would allow differentiation between the sentences, appreciated by human evaluators as good/creative (regarding their novelty and narrative potential) or bad. We tested also an alternative approach for gathering human evaluations through interaction with the robot Nao [35]. Other contributions of Slovenian researchers to the WHIM project included bisociative generation of fictional ideation [32] and the RoboChair[10] system for enhancing scientific creativity by generating questions regarding decisions made by authors when writing scientific articles [37].

4.3 Concept creation technology (ConCreTe)

The ConCreTe[11] project focused on AI technology for concept construction, identification, and evaluation. ConCreTe addressed several forms of conceptual blending (CB), a basic cognitive mechanism by which two or more mental spaces are integrated to produce new concepts [15]. Optimality principles (OPs), a key element in the CB framework, are responsible for guiding the integration process towards good blends. The role of OPs was studied from the point of view of computational systems [24], as well as within a study of human perception of visual animal blends[12] [25], performed with the aim of better understanding of creative artefacts reception.

The main contribution of JSI to ConCreTe was the ConCreTeFlows platform[13] [48] for collective CC workflows construction. It is a platform built on top of the existing ClowdFlows infrastructure [21], but it is specialised at supporting (primarily text-based) computational creativity tasks, such as conceptual blending and poetry generation. It currently contains more than 35 native widgets for supporting creativity by developers from five different institutions participating in ConCreTe. The asset of a web-based system is that it integrates creative software written in a large variety of programming languages (e.g., components written in Python, C#, Java, PROLOG). An interesting example of multimodal conceptual blending [48] is available as an interactive workflow.[14]

4.4 Other projects and activities

We have described the main projects from the field of CC in which we were actively involved. Other project were closely related to computational creativity. For example, within the EU project MUSE[15], the question of interactive story-telling was addressed. Our main role was the integration of the developed components in the online workflow environment [33].

The PROSECCO[16] networking action had a crucial role in building the European CC community, with a number of events including the organisation of summer schools, code camps, etc. Computational Creativity has became an important research topic in Slovenia. A large number of activities were organised also by Slovenian researchers and held place in Ljubljana, including the 5th edition of the ICCC conference[17], with material available through VideoLectures[18], and the Symposium on Computational Creativity[19]. We have also organised the Computational Creativity art exhibition entitled You/Me/It.[20]

Since 2016, a Computational Creativity course has been offered at the International Postgraduate School Jožef Ste-

[10] http://kt-robochair.ijs.si/

[11] http://www.conceptcreationtechnology.eu

[12] http://animals.janez.me/

[13] http://concreteflows.ijs.si

[14] http://concreteflows.ijs.si/workflow/137/

[15] http://www.muse-project.eu/

[16] http://prosecco-network.eu/

[17] http://procsecco-network.eu

[18] http://videolectures.net/iccc2014_ljubljana/

[19] http://videolectures.net/ktsymposium2013_ljubljana/

[20] http://computationalcreativity.net/iccc2014/you-me-it-art-exhibition/

fan[21].

As CC related outreach activity, a large number of events for children and youth were organised for science promotion by means of a Nao robot, for which the main developer Vid Podpečan received the Slovenian "Prometej znanosti" (Prometheus of Science) science dissemination award.

5 Conclusion

This paper presented a brief review of historic and current activity in Computational Creativity, an exciting and relatively new sub-field of Artificial Intelligence. In particular, we have highlighted contributions from Slovenian researchers.

Computational Creativity is in some sense a final frontier for AI [11], because it pulls the field away from comfortably defined problem-solving activity such as classification, into the areas that are more challenging to formulate. Much of the work in this developing field is focused not so much on "What is the answer?" but rather on "What is the question?", and this makes for exciting prospects for the future, both in Slovenia and elsewhere. In 2008, the Association for Computational Creativity[22] (ACC) was founded to manage the ICCC conferences and support the CC community into the future.

Acknowledgements

We acknowledge the support of the Slovenian Research Agency (core funding no. P2-0103), the European projects Prosecco (grant no. 600653) and ConCreTe (grant nb. 611733). GW is very grateful to the International Postgratuate School Jožef Stefan internationalisation grant for funding a sabbatical visit in Autumn 2017, which enabled his contribution to this paper.

Literature

[1] Boden, M. (1977). *Artificial Intelligence and Natural Man*. Harvester Press.

[2] Boden, M. A. (2004). *The Creative Mind: Myths and Mechanisms* (2nd ed.). Routledge.

[3] Browne, C. (2008). *Automatic Generation and Evaluation of Recombination Games*. Ph. D. thesis, Queensland University of Technology.

[4] Cestnik, B., E. Fabbretti, D. Gubiani, T. Urbančič, and N. Lavrač (2017). Reducing the search space in literature-based discovery by exploring outlier documents: A case study in finding links between gut microbiome and alzheimers disease. *Genomics and Computational Biology 3*(3), 58.

[21] https://www.mps.si/splet/studij.asp?lang=eng&main=1&left=4&id=721&m=4

[22] http://computationalcreativity.net

[5] Cohen, H. (1979). What is an image? In *Proceedings of the 1979 International Joint Conference on Artificial Intelligence*.

[6] Cohen, H. (1988). How to draw three people in a botanical garden. In *Proceedings of the 1988 Conference of the American Association for Artificial Intelligence (AAAI-88)*.

[7] Cohen, H. (1999). Colouring without seeing: A problem in machine creativity. *AISB Quarterly 102*, 26–35.

[8] Colton, S. (2012a). *Automated Theory Formation in Pure Mathematics*. Distinguished Dissertations. Springer London.

[9] Colton, S. (2012b). The painting fool: Stories from building an automated artist. In J. McCormack and M. d'Inverno (Eds.), *Computers and Creativity*. Springer-Verlag.

[10] Colton, S., M. T. Llano, R. Hepworth, J. W. Charnley, C. V. Gale, A. Baron, F. Pachet, P. Roy, P. Gervás, N. Collins, B. L. Sturm, T. Weyde, D. Wolff, and J. R. Lloyd (2016). The Beyond the Fence musical and Computer Says Show documentary. In *Proceedings of the Seventh International Conference on Computational Creativity, UPMC, Paris, France, June 27 - July 1, 2016.*, pp. 311–321.

[11] Colton, S. and G. A. Wiggins (2012). Computational creativity: The final frontier? In de Raedt L. et al. (Ed.), *Proceedings of ECAI Frontiers*.

[12] Cook, M., S. Colton, A. Raad, and J. Gow (2013). Mechanic miner: Reflection-driven game mechanic discovery and level design. In A. I. Esparcia-Alcázar (Ed.), *Applications of Evolutionary Computation: 16th European Conference, Proceedings*, pp. 284–293. Springer.

[13] Cope, D. (1992). Computer modelling of musical intelligence in EMI. *Computer Music Journal 16*(2), 69–83.

[14] Ebcioğlu, K. (1988). An expert system for harmonizing four-part chorales. *Computer Music Journal 12*(3), 43–51.

[15] Fauconnier, G. and M. Turner (2002). *The Way We Think*. New York: Basic Books.

[16] Fortuna, B., D. Mladenič, and M. Grobelnik (2006). Semi-automatic construction of topic ontologies. In *Semantics, Web and Mining: Joint International Workshops, EWMF 2005 and KDO 2005, Revised Selected Papers*, pp. 121–131. Springer.

[17] Gervás, P. (2015). Computational drafting of plot structures for Russian folk tales. *Cognitive Computation*.

[18] Jordanous, A. (2012). A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation 4*(3), 246–279.

[19] Juršič, M., B. Cestnik, T. Urbančič, and N. Lavrač (2012, may). Cross-domain literature mining: Finding bridging concepts with crossbee. In *Proceedings of the Third International Conference on Computational Creativity*, Dublin, Ireland, pp. 33–40.

[20] Koestler, A. (1976). *The Act of Creation*. London, UK: Hutchinson.

[21] Kranjc, J., V. Podpečan, and N. Lavrač (2012). ClowdFlows: A cloud based scientific workflow platform. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*, pp. 816–819.

[22] Llano, M. T., S. Colton, R. Hepworth, and J. Gow (2016). Automated fictional ideation via knowledge base manipulation. *Cognitive Computation 8*(2), 153–174.

[23] Macedoni-Lukšič, M., T. Urbančič, I. Petrič, and B. Cestnik (2016). Autism research dynamic through ontology-based text mining. *Advances in Autism 2*(3), 131–139.

[24] Martins, P., S. Pollak, T. Urbančič, and A. Cardoso (2016). Optimality principles in computational approaches to conceptual blending: Do we need them (at) all? In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*, Paris, France. Sony CSL: Sony CSL.

[25] Martins, P., T. Urbančič, S. Pollak, N. Lavrač, and A. Cardoso (2015). The good, the bad, and the aha! blends. In *Proceedings of ICCC*, pp. 166–173. computationalcreativity.net.

[26] McCorduck, P. (1991). *AARON'S CODE: Meta-Art, Artificial Intelligence and the Work of Harold Cohen'S CODE: Meta-Art, Artificial Intelligence and the Work of Harold Cohen*. Freeman.

[27] Norton, D., D. Heath, and D. Ventura (2013). Finding creativity in an artificial artist. *Journal of Creative Behavior 47*(2), 106–124.

[28] Pearce, M. T. (2005). *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*. Ph. D. thesis, Department of Computing, City University, London, London,UK.

[29] Pearce, M. T. and G. Wiggins (2007). Evaluating cognitive models of musical composition. In A. Cardoso and G. Wiggins (Eds.), *Proceedings of the 4th International Joint Workshop on Computational Creativity*, London, pp. 73–80. Goldsmiths, University of London.

[30] Pereira, F. C. (2007). *Creativity and Artificial Intelligence: A Conceptual Blending Approach*. Berlin: Mouton de Gruyter.

[31] Pérez y Pérez, R. and M. Sharples (2001). Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence 13*(2), 119–139.

[32] Perovšek, M., B. Cestnik, T. Urbančič, S. Colton, and N. Lavrač (2013). Towards narrative ideation via cross-context link discovery using banded matrices. In *IDA*, Volume 8207 of *Lecture Notes in Computer Science*, pp. 333–344. Springer.

[33] Perovšek, M., V. Podpečan, J. Kranjc, T. Erjavec, S. Pollak, N. Q. Do Thi, X. Liu, C. Smith, M. Cavazza, and N. Lavrač (2015). Text mining platform for NLP workflow design, replication and reuse. In *Proceedings of IJCAI Workshop on Replicability and Reusability in Natural Language Processing: From Data to Software Sharing, Buenos Aires, Argentina, 26 July 2015*.

[34] Petrič, I., B. Cestnik, N. Lavrač, and T. Urbančič (2012, January). Outlier detection in cross-context link discovery for creative literature mining. *Comput. J. 55*(1), 47–61.

[35] Podpečan, V. (2015). The What-If machine robot interface (WHIMBOT). In *Show, tell imagine: A day to explore computational creativity together*, pp. 17. Queen Mary, Univ. of London.

[36] Pollak, S., B. M. Boshkoska, D. Miljkovic, G. Wiggins, and N. Lavrač (2016). Computational creativity conceptualisation grounded on iccc papers. In V. C. F. a. G. François Pachet, Amilcar Cardoso (Ed.), *Proceedings of ICCC 2016*, pp. 123–130. Association for Computaitonal Creativity.

[37] Pollak, S., B. Lesjak, J. Kranjc, V. Podpečan, M. Žnidaršič, and N. Lavrač (2015). RoboCHAIR: Creative assistant for question generation and ranking. In *Proceedings of SSCI*, pp. 1468–1475. IEEE.

[38] Propp, V. (1968). *Morphology of the folktale*. Austin: University of Texas Press.

[39] Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds and Machines 17*(1), 67–99.

[40] Sakellariou, J., F. Tria, V. Loreto, and F. Pachet (2017). Maximum entropy models capture melodic styles. *Scientific Reports 7*(9172).

[41] Salton, G. and C. Buckley (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management 24*(5), 513–523.

[42] Sluban, B., M. Juršič, B. Cestnik, and N. Lavrač (2012). *Exploring the Power of Outliers for Cross-Domain Literature Mining*, pp. 325–337. Berlin, Heidelberg: Springer Berlin Heidelberg.

[43] Swanson, D. R., N. R. Smalheiser, and V. I. Torvik (2006). Ranking indirect connections in literature-based discovery: The role of medical subject headings. *Journal of the American Society for Information Science and Technology 57*(11), 1427–1439.

[44] Turner, M. and G. Fauconnier (1995). Conceptual integration and formal expression. *Metaphor and Symbolic Activity 10*(3), 183–203.

[45] van der Velde, F., R. Wolf, M. Schmettow, and D. Nazareth (2015, 6). A semantic map for evaluating creativity. In H. Toivonen, S. Colton, M. Cook, and D. Ventura (Eds.), *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, pp. 94–101. WordPress. Open access.

[46] Veale, T. (2012). *Exploding the Creativity Myth*. New York, NY: Bloomsbury Academic.

[47] Veale, T. and G. Li (2016, Apr). Distributed divergent creativity: Computational creative agents at web scale. *Cognitive Computation 8*(2), 175–186.

[48] Žnidaršič, M., A. Cardoso, P. Gervás, P. Martins, R. Hervás, A. O. Alves, H. G. Oliveira, P. Xiao, S. Linkola, H. Toivonen, J. Kranjc, and N. Lavrač (2016). Computational creativity infrastructure for online software composition: A conceptual blending use case. In *Proceedings of the Seventh International Conference on Computational Creativity, UPMC, Paris, France, June 27 - July 1, 2016.*, pp. 371–379.

[49] Wiggins, G. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Journal of Knowledge Based Systems 19*(7), 449–458.

[50] Wiggins, G. (2007). Models of musical similarity. *Musicae Scientiae 11*, 315–338.

[51] Wiggins, G., P. Tyack, C. Scharff, and M. Rohrmeier (2015). The evolutionary roots of creativity: mechanisms and motivations. *Philosophical Transactions of the Royal Society of London B: Biological Sciences 370*(1664).

Towards Creative Software Blending: Computational Infrastructure and Use Cases

Matej Martinc[1,2], Martin Žnidaršič[1], Nada Lavrač[1,3] and Senja Pollak[1]
[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia
[3] University of Nova Gorica, Nova Gorica, Slovenia
E-mail: matej.martinc@ijs.si, nada.lavrac@ijs.si, martin.znidarsic@ijs.si, senja.pollak@ijs.si

Keywords: computational creativity, software blending, visual programming platforms

Received: October 31, 2017

Numerous visual programming platforms support the generation, execution and reuse of constructed scientific workflows. However, there has been little effort devoted to building creative software blending systems, capable of composing novel workflows by autonomously combining individual software components or even entire workflows originally designed for solving tasks in different research fields. Based on the review of relevant computational creativity research and of contemporary web platforms for workflow construction, this paper defines the desired functionality of a software blending system. Considering the required autonomy of the system and the workflow complexity limitations, we investigate the necessary conditions for the implementation of a creative blending system within the existing visual programming platforms.

Povzetek: Številne platforme za vizualno programiranje podpirajo gradnjo, izvajanje in ponovno uporabo zgrajenih znanstvenih delotokov. Dosedanje raziskave niso posvečale pozornosti izdelavi kreativnih sistemov za spajanje programske opreme, ki bi bili sposobni avtonomnega sestavljanja posameznih programskih komponent ali celo celotnih delotokov, prvotno izdelanih za reševanje nalog na različnih znanstvenih področjih. Na podlagi pregleda raziskav s področja računalniške ustvarjalnosti in obstoječih spletnih platform za gradnjo delotokov v tem članku definiramo želeno funkcionalnost sistema za kreativno spajanje programske opreme. Upoštevaje zahteve po avtonomnosti sistema in dovoljeno kompleksnost delotokov preučimo tudi pogoje za implementacijo takega sistema v obstoječih platformah za vizualno programiranje.

1 Introduction

Creativity was defined by M. Boden [3] as "the ability to come up with ideas or artefacts that are new, surprising, and valuable". It is considered as an aspect of human intelligence, grounded in everyday abilities such as conceptual thinking, perception, memory and reflective self-criticism.

Software is usually not considered creative because it follows explicit instructions of the programmer [4]. However, writing software is considered to be a creative task. If a program could define its own instructions, this would clearly mean that the program has some level of creativity.

A subfield of artificial intelligence has recently emerged, in which one of the main goals is the creation of software that is able to model, simulate or replicate human creativity. This field, called *computational creativity*, has been defined by S. Colton and G. Wiggins [6] as "the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative."

Note that the field of computational creativity should not be confused with the field of *creative computing*. Although these two research areas partly overlap, creative computing differs from computational creativity by generally not being considered as a subfield of artificial intelligence, since it mostly addresses the task of creative development of computing products and with how to write software that would better serve the needs of the creative community [13].

Infrastructures supporting computational creativity and the generation of creative systems are scarce, although some recent research attempts has tried to fill this gap. One of the recent developments is FloWr [4], a system for implementing creative systems as scripts over processes and manipulated visually as flowcharts. Another is the ConCreTeFlows infrastructure [27], which was developed to enable the construction, sharing and execution of computational creativity (CC) workflows, composed of software ingredients of different partners of European project ConCreTe[1]. Both of these infrastructures use different types of resources (e.g., musical, pictorial and textual inputs) in order to support the development of some typical CC task such as poetry generation, metaphor creation, generation of narratives, creation of fictional ideas and conceptual blending.

[1] http://conceptcreationtechnology.eu

These platforms, which enable the user to build procedures capable of producing a variety of different creative artefacts, could hardly be called creative systems, since they do not exhibit creative behavior in terms of automated workflow development. The arguably most creative system for automated workflow construction, optimization and alteration, which is implemented in the FloWr platform, requires a lot of manual user input and could only be called creative with some major reservations.

To fill the identified gap, this paper addresses the task of developing an infrastructure capable of autonomously composing novel scientific workflows by creatively combining individual software components or even entire workflows originally designed for specific tasks in different research fields. We consider the process of autonomous workflow composition—which we name *creative software blending* in this paper—to be an important first step towards a long term goal of creating software that could write code directly. The proposed system would be able to bridge different scientific fields by combining methods from specific fields into novel interdisciplinary workflows. It would ideally also be capable of automated interdisciplinary research by autonomously discovering novel scientific procedures.

This paper presents the design principles underlying a creative system described above. Section 2 introduces the research topic and presents the infrastructures suitable for the implementation of a creative software blending system. Section 3 motivates this research by presenting two existing hand-blended workflows. Section 4 presents the related software blending and computational creativity research, followed by an outline of the desired system functionality, investigating the necessary conditions for the implementation of a creative system for autonomous creative workflow generation. The paper concludes by presenting plans for future work.

2 Research background and infrastructures

As background to our creative software blending research, this section first outlines some creativity support tools, followed by a brief description of a selection of easy-to-use workflow management systems that allow the user to compose complex computational pipelines in a modular visual programming manner.

2.1 Creative software

As Colton's and Wiggins' definition of computational creativity [6] is hardly operational for measuring creativity of a program, G. Ritchie [23] proposed some empirical criteria for attributing creativity to a computer program. The main idea is to use empirically observable and comparable factors, such as the properties of the generated output of the creative system, when trying to assess the creativity of a system. These observable factors can be judged by two quantifiable and essential criteria:

Novelty of an output determines to what extent is the produced item dissimilar to existing examples of its genre.

Quality of an output determines to what extent is the produced item a high quality example of its genre.

Using these criteria, we can say that the system for creative software blending is creative if it outputs novel and high quality scientific workflows.

Another relevant question is what types of creative behaviors exist and how can they be computationally modeled. Boden [3] distinguishes three basic types of creativity:

Combinational creativity involves making unfamiliar combinations of familiar ideas.

Exploratory creativity involves exploration of a conceptual space, which is characterized as a structured style of thought, and coming up with a new idea or artefact within that thinking style.

Transformational creativity refers to the modification of the conceptual space so that new kinds of ideas and artefacts can be generated.

Combinational creativity is the easiest one to be modeled on a computer. However, created combinations should be meaningful and interesting, which usually requires a solid background knowledge and the ability to form and evaluate relations of many different types. Several programs exist that can explore a given space and invent new artefacts with a certain style, for example, a program for automatic music generation [19] or a program for generating game designs [7]. Some programs can even transform their conceptual space by altering their own rules; for example, evolutionary algorithms can make random changes in their current rules and by this evolve new structures.

Another important distinction made by Boden [3] is a distinction between *psychological creativity* (P-creativity) and *historical creativity* (H-creativity). P-creativity relates to creation of surprising, valuable ideas and artefacts that are new to the person who comes up with it. However, if an artefact or idea has arisen for the first time in human history and (so far as we know) nobody else has had it before, then we are talking about H-creativity. We anticipate that if the targeted creative software blending system is to be an active participant in scientific discovery or artefact creation, it should ideally be H-creative, although even a P-creative system can play a very useful supporting role in scientific research and its development is therefore a worthy research goal.

2.2 Infrastructures

A system for creative software blending would best be implemented inside an already existing infrastructure enabling interdisciplinary and creative scientific workflow

composition. In this section we present the ClowdFlows and ConCreTeFlows platforms that host the two motivational use cases, but other platforms, such as FloWr [4], Rapid Miner [18], KNIME [2], ORANGE [8] are also worth exploring as potential infrastructures for creative software blending.

ClowdFlows [16] is a cloud-based web application[2] for composition, execution and sharing of interactive data mining workflows. It has a web based user interface for building workflows, runs in all major browsers and requires no installation. It contains a large set of workflow components called *widgets*, which can be connected in a specific meaningful order to create a *workflow*. ClowdFlows enables visual programming and has a graphical user interface which consists of a widget repository and a workflow canvas.

ConCreTeFlows [27] is a platform[3] built on top of the ClowdFlows infrastructure. It is specialized in computational creativity tasks, including conceptual blending based on textual or visual input or text generation tasks, such as poetry generation.

The specialization of ConCreTeFlows in computational (and especially text-based) creativity, as well as a smaller number of implemented widgets, makes it less appropriate for the implementation of the proposed system for creative software blending, but it is appropriate to showcase the creative blending process. On the other hand, ClowdFlows is not specialized in a single specific research field and contains widgets from the fields of text mining, machine learning and NLP, which makes it appropriate for the implementation of a creative software blending system since combining tools from different research fields would most likely increase the chance of the system to be H-creative. As a basis of automated software composition, ClowdFlows already includes a—somewhat loosely defined—ontology of its components (named widgets), which should be enhanced and elaborated in further work, to enable ClowdFlows to actually become a useful infrastructure for software blending.

3 Motivational use cases

This section presents two hand-crafted motivational workflows, which illustrate the usefulness of blending software from different scientific fields in order to develop new innovative scientific methods. In this sense, they represent the type of workflows that a system for creative software blending would be capable to produce.

3.1 Wordification use case: Blending data mining and text mining in ClowdFlows

Propositionalization [15] is an approach to inductive logic programming (ILP) and relational data mining (RDM), which offers a way to transform a relational database into a propositional single-table format. Consequently, learning with propositionalization techniques is divided into two self-contained phases: (1) transformation of relational data into a single-table format and (2) selecting and applying a propositional learner to the transformed data set. As an advantage, propositionalization is not limited to specific data mining tasks such as classification, which is usually the case with ILP and RDM methods that directly induce predictive models from relational data. This section motivates creative software blending by outlining the Wordification workflow [22], implemented in ClowdFlows, which performs propositionalization by combining data mining and text mining techniques.

In the Wordification workflow, shown in Figure 1, given a MySQL relational database as input, the user selects the target table from the initial relational database, which will later represent the main table in the Wordification component of the workflow. The user is able to discretize each of the tables using one of the available discretization techniques. These discretized tables are used by the Wordification widget, where the transformation from the relational tables to a 'corpus of documents' is performed.

Several elements of blending data mining and text mining techniques are incorporated in the Wordification: i.e. transforming attribute values into bags of word-like items, using TF-IDF weighting of items, and the possibility of using n-grams of items where n-gram construction is performed by taking every combination of length n of items from the set of all items corresponding to the given individual. Nevertheless, the element of the workflow that most clearly illustrates the software blending potential is the inclusion of a *word cloud* visualization (an approach developed in text mining research), together with decision tree construction and visualization (an approach developed in data mining research).

3.2 Conceptual blending use case: computational creativity in ConCreTeFlows

The elements of the *conceptual blending* theory [12], described in more detail in Section 4, are an inspiration to many algorithms and methodologies in the field of computational creativity. In brief, according to this theory, two different concepts for which we can define (find) a similarity, can be blended into a new concept in the context of knowledge that is necessary to represent and generalize the two concepts.

[2] Available at `http://clowdflows.org`
[3] Available at `http://concreteflows.ijs.si`

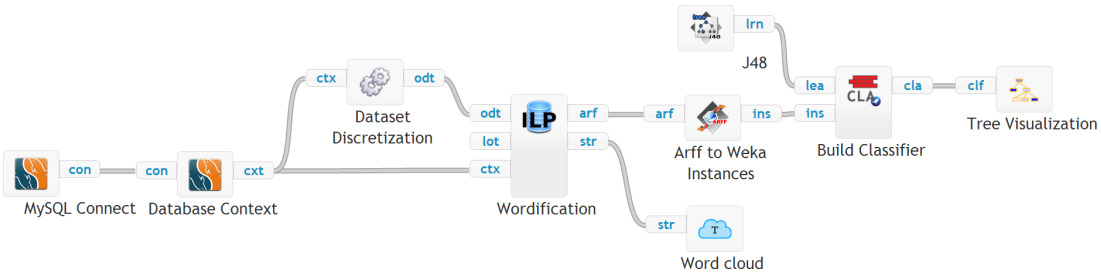


Figure 1: Clowdflows Wordification workflow with additional analyses after the wordification process, available at `http://clowdflows.org/workflow/1455/`.

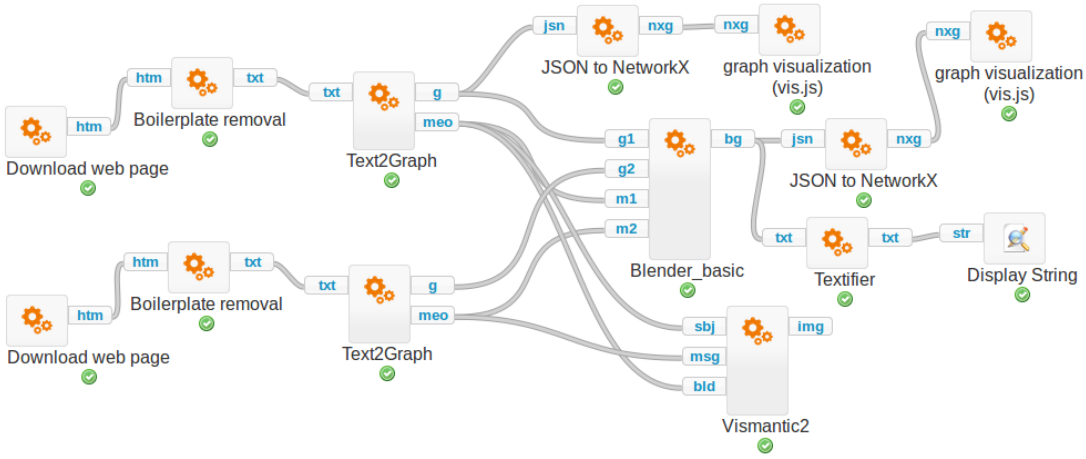


Figure 2: Workflow implementation of multimodal blending in ConCreTeFlows, available at: `http://concreteflows.ijs.si/workflow/137/`.

Let us present a conceptual blending CC workflow [27], implemented in the ConCreTeFlows platform by different partners of the ConCreTe project. Its process components are implemented either as internal functions, wrapped standalone programs or as Web services. The publicly available workflow, presented in Figure 2, can be executed, changed and extended with additional functionality.

The workflow presents conceptual blending by constructing conceptual graphs from textual input and representing the results (blends) as graphs, natural language descriptions and visual representations. Two textual inputs are transformed into conceptual graphs by a series of widgets: the *Download web page* for obtaining the Web page source from a given URL (In the example, these are the Wikipedia pages for two animals: hamster and zebra.), *Boilerplate removal* and *Text2Graph* transforming the textual content into conceptual graphs (output *g*). The outputs of Text2Graph widgets enter *Blender_basic*, which blends the two graphs together and outputs a combined blended graph (output *bg*). This one gets served to the *Textifier widget*, which produces a textual description of the blend. Its output is presented by a standard *Display String* widget. The two main entities from Text2Graph widgets enter also the *Vismantic2* visual blending widget [28], which either changes the texture of one input space to the texture of the other (see Figure 3a), or puts one in the usual surroundings of the other. (Figure 3b). Its outcome is shown in an output similar to the ones shown in Figure 3.

4 Towards design principles for creative software blending

There are two major paradigms in artificial intelligence research: problem solving and artefact generation [5]. While the problem solving paradigm deals with a series of problems that needs to be solved, in the artefact generation paradigm the task is to generate a series of valuable artefacts. This study is more related to the latter and the artefacts of our interest are functional workflows.

A creative software blending system should be able to build new workflows composed of software components from different fields, leading to novel ways of software composition for computational purposes that were not expected in advance. Such blending of software would best be implemented in an existing infrastructure for interdisciplinary scientific research with already implemented components for specific and well defined tasks.

As shown in Section 2.2, much effort in the fields of data mining and NLP has already been devoted to the development of infrastructures that provide support for easier and quicker experimentation. One of the biggest challen-

(a) (b)

Figure 3: Two outputs of the Vismantic2 widget for the example of blending the concepts of *hamster* and *zebra*: left is a result of exchanging hamster's texture with zebra's and the right is an example of exchanging zebra's with a hamster's common visual context.

ges in implementation and use of these infrastructures has been the integration of different components into functional workflows. Combining different tools and technologies in a common infrastructure is a difficult task because of software incompatibility and inappropriately defined ontologies.

4.1 Related software blending research

To design an appropriate creative software blending system one should consider three fields of study. First, one has to reflect upon the concept of creativity and how to build software that exhibits creative behavior (see the related research in Section 2.1). Next, one has to be aware of strengths and limitations of the existing infrastructures that could be used as a platform for the implementation of our system (see the related infrastructures in Section 2.2). Finally, one has to become aware of potentially existing implemented approaches for software blending, surveyed below.

While the FloWr framework [4] is conceptually very similar to the two infrastructures described in Section 2.2, it is currently the only one with a specifically defined aim of being able to automatically optimize, alter and ultimately generate novel workflows presented as flowcharts. This automatic workflow generation via the combination of code modules means that FloWr has the potential to innovate at the process level and the manifested long-term goal is a software system that can write program code for itself [4]. Although the platform currently does not support fully functioning software blending, some preliminary experiments to automatically alter, optimize and generate flowcharts have been conducted.

One of the FloWr experiments dealt with an automatic construction of a system for producing poetic couplets from scratch. In order to reduce the number of possible combinations of different workflow components, only a subset of all the available components were manually selected for blending in the experiment. Possible options for the input parameters were manually reduced and the number of components in the generated workflow was limited to 3 to 5. Despite these limitations, at the end there were still over 261 million variable definition combinations. For this reason the brute-force approach of testing all combinations was intractable, so a depth-first search for all possible workflows was implemented in a way that just one node combination and one parameter setting were randomly selected from a set of allowed combinations. The compatibility of sequential components and some other restrictions were taken into account, which reduced the number of possible workflow candidates. The algorithm was run 200 times resulting in 200 workflows. A manual evaluation showed that 18.5% of workflows worked successfully and produced poetic couplets.The conducted experiment required a lot of human intervention in order to be successful and the evaluation of produced artefacts was done by humans. Because of this we can question the creativity of the proposed software blending approach since a software should — at least in our opinion — have the capacity to evaluate its own performance in order to be called creative.

While FloWr belongs to CC research, several attempts have been made to develop support systems also in the field of knowledge discovery. These systems are to some extent related to our research, since they either support the users workflow composition by recommending the new components that could be attached to an existing workflow, or by generating entire workflows according to user requirements.

Zakova et al. [26] proposed a semi-automatic system for workflow generation that is based on a background knowledge ontology in which all workflow components are described together with their inputs, outputs and pre-/post conditions. The system uses a planning algorithm and returns just one optimal workflow with the smallest number of processing steps. Given that alternative workflows are not generated, this is not in accordance with a desired system for creative software blending. Complexity limitations are another problem, which is common to all the systems that use planning approaches for workflow generation.

The IDEA system by Bernstein et al. [1] is based on an ontology of data mining components that guides the workflow composition and contains heuristics for the rankings of different alternatives. The system does not enable fully

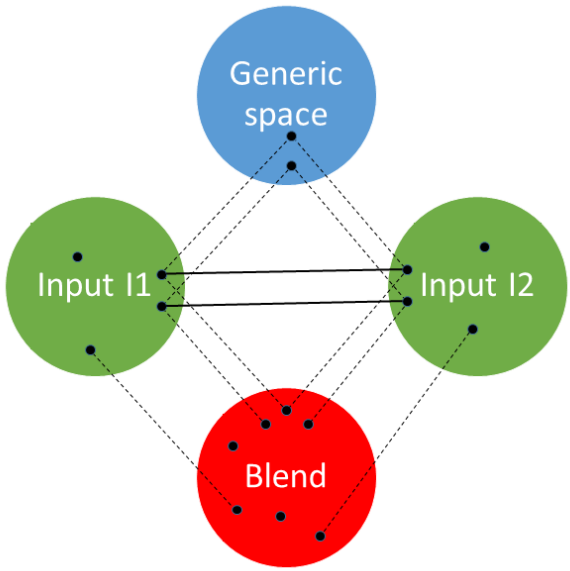


Figure 4: The conceptual blending network [11].

automatic workflow generation but was implemented as a support system for the user who decides on the weights to determine the trade-off between different performance criteria (e.g., speed, accuracy, comprehensibility). IDEA is limited to proposing fairly simple workflows.

Kietz et al. [14] proposed a KDD support system that uses a data mining ontology. The ontology contains information about the objects manipulated, the meta data, the operators (i.e. components containing algorithms for specific tasks) and a description of the goal, which is a formalization of the user desired output. The system takes a goal description as an input and returns a workflow together with all the evaluation and reporting needed to let the user assess if it fulfills the user defined success criterion. The system, implemented in the RapidMiner platform, is not fully autonomous, as it was designed as a support system for the user.

4.2 Design principles

As the above survey shows, no adequate solution for the autonomous creative software blending currently exists. To build such a system, first, an ontology with well-defined rules and relations needs to be created, in order to enable combining software components in a meaningful way. Next, a system for creative blending of software components would be created, enabling automated combination of components in functional workflows.

Computational creativity, which is still in early phrases of its development [25], provides some methodological apparatus and inspiration for designing the guiding principles for a creative software blending system. One of the very productive fields of research in computational creativity is the *conceptual blending* (CB) theory [12], which inspired many algorithms, methodologies and discussions in the field (e.g., [24, 21, 17]).

CB is a basic mental operation that leads to new meaning, global insight, and conceptual compressions useful for memory and manipulation of otherwise diffuse ranges of meaning [9]. A key element is the *mental space*, a partial and temporary structure of knowledge built for the purpose of local understanding and action [10].

To describe the CB process, the theory [12] makes use of a network of four mental spaces (see Figure 4). In blending, structure from two input mental spaces (*Input* I_1, *Input* I_2), is projected to a new space, the blend. A partial mapping between elements of input spaces—that are perceived as similar or analogous in some respect—is performed. The third mental space, called *generic space*, encapsulates the conceptual structure shared by the input spaces, generalizing and possibly enriching them. This space provides guidance to the next step, where elements from each of the input spaces are selectively projected into the *blend*, i.e. the new blended mental space. Emergent structure arises in the blend that is not copied there directly from any input.

The conceptual blending model is not directly transferable from the human cognition to the blending of software. However, the methodology, together with the optimality principles [11] that optimize the blending process—which were already addressed also in computational models [20]—should be considered when implementing the software blending algorithm and the workflow ontology. For example, in software blending the two inputs would not represent concepts but rather two workflows from two different scientific domains. The "generic space" could then be adapted to software blending in order for the blending system to find all the compatible widgets from two different input workflow domains. Finally, the blend would be a newly produced workflow containing new emergent structures not copied from original workflows. The optimality principles, such as the relevance principle (which dictates that all elements in the blend should be relevant) and integration principle (which states that the final blend should be perceived as an enclosed unit) should be kept in mind when designing the ontology.

Another important aspect to be considered in the implementation of the system is its creative part. In order for the system to be recognized as creative, its produced artefacts should be novel and of good quality [23] and the human interference in the production and evaluation of these artefacts should be minimal. Three criteria are proposed for attributing creative autonomy to a system [25]:

Autonomous evaluation The system should be able to evaluate new creations autonomously and possess its own "opinion" on which creations are better than others.

Autonomous change The system should be able to change its evaluation function without explicit directions.

Non-Randomness (Aleatoricism) Random behavior is not creative, so evaluation and change should not be

completely random, although some randomness can be involved.

In order to satisfy these criteria and since most of the aforementioned platforms contain a large set of manually built workflows that could be used as a training set, we propose a combination of an evolutionary algorithm and a classification model induction. An evolutionary algorithm would operate directly on representations of workflows and generate new workflow candidates, according to the constraints defined by ontology rules. These constraints would enforce a minimum quality for the produced workflows (corresponding to the criterion of quality [23], which is, as explained earlier, one of the guiding principles in the construction of creative artefacts).

The initial population of the evolutionary algorithm would consist of manually built workflows that would be "blended" into new workflow candidates with the help of mutation and crossover. The fitness function used for evaluating the fitness of the generated workflow candidates would contain following elements:

A binary classification model trained on the features extracted from successful and unsuccessful workflows would serve as an additional workflow quality check.

A similarity function for determining the similarity between a generated workflow candidate and existing workflow would be used for evaluating the novelty of the candidate.

In this way the system would be able to generate new—possibly creative—workflows and even propose changes in the existing rules for workflow generation, which would make this system capable of transformational creativity according to [3].

5 Conclusions

In this study we elaborate the initial design principles of a system for automatic workflow generation that would be capable of autonomous composition of novel workflows from existing software components. We have presented two workflows with human-designed blending, implemented in the ClowdFlows and ConCreTeFlows platforms for online workflow composition. The first workflow clearly illustrated the potential for the composition of computational creativity solutions. The second use case presents several computational creativity software components that were combined in a collaborative effort to implement an interesting conceptual blending solution, resulting in conceptual, visual and textual blends. The benefits of a unifying workflow for blending are twofold: the user can get blends of various kinds through the same user interface and the components can affect one another to produce a more coherent and orchestrated set of multimodal blending results. The presented prototype solution is fully operational and serves as a proof of concept that such an approach to multimodal conceptual blending is possible.

On the other hand, the sketched evolutionary algorithm approach to blending workflows and workflow components shows, that the theory of conceptual blending can be transferred to the problem of creative software blending. We also demonstrated that the system will be capable of self evaluation by using the empirical criteria of novelty and quality in the fitness function.

In our future work we will first design an ontology capable of supporting the planned widget recommender system. We also plan to integrate a larger number of widgets and workflows in the presented platforms. Moreover, we will undertake the challenging task of the implementation. We realize that creation of software that can innovate at a process level is a very demanding task and we can expect many challenges during this phase. Anyhow, we do believe that the effort will be fruitful and bring us closer to the long-term goal of creating software that could write novel and valuable code directly.

Acknowledgments

We acknowledge the support of the Slovenian Research Agency through research programme Knowledge Technologies (grant number P2-0103), and project ClowdFlows Data and Text Analytics Marketplace on the Web (CF-Web), which has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 754549. We would like to thank Pedro Martins and Amilcar Cardoso for numerous discussions on the topic of conceptual blending.

References

[1] Bernstein, A., Provost, F., Hill, S.: Toward intelligent assistance for a data mining process: An ontology-based approach for cost-sensitive classification. IEEE Transactions on Knowledge and Data Engineering 17(4), 503–518 (2005)

[2] Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinl, T., Ohl, P., Thiel, K., Wiswedel, B.: Knime–the Konstanz Information Miner: Version 2.0 and beyond. ACM SIGKDD Explorations Newsletter 11(1), 26–31 (2009)

[3] Boden, M.A.: Creativity in a nutshell. Think 5.15, 83–96 (2007)

[4] Charnley, J., Colton, S., Llano, M.T.: The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In: Proc. of the Fifth International Conference on Computational Creativity. pp. 315–323 (2014)

[5] Colton, S., Ramezani, R., Llano, M.: The hr3 discovery system: Design decisions and implementation

details. In: Proc. of the AISB Symposium on Computational Scientific Discovery (2014)

[6] Colton, S., Wiggins, G.: Computational creativity: The final frontier? In: Proc. of the 20th European Conference on Artificial Intelligence. pp. 21–26 (2012)

[7] Cook, M., Colton, S.: Multi-faceted evolution of simple arcade games. IEEE Conference on Computational Intelligence and Games (CIG) pp. 289–296 (2011)

[8] Demšar, J., Zupan, B., Leban, G., Curk, T.: Orange: From experimental machine learning to interactive data mining. In: European Conference on Principles of Data Mining and Knowledge Discovery. pp. 537–539. Springer (2004)

[9] Fauconnier, G., Turner, M.: Conceptual blending, form and meaning. Recherches en communication 19(19), 57–86 (2003)

[10] Fauconnier, G.: Mental Spaces: Aspects of Meaning Construction in Natural Language. Cambridge University Press (1994)

[11] Fauconnier, G., Turner, M.: Conceptual integration networks. Cognitive Science 22(2), 133–187 (1998)

[12] Fauconnier, G., Turner, M.: The way we think: Conceptual blending and the mind's hidden complexities. Basic Books (2002)

[13] Hugill, A., Yang, H.: The creative turn: new challenges for computing. International Journal of Creative Computing 1(1), 4–19 (2013)

[14] Kietz, J., Serban, F., Bernstein, A., Fischer, S.: Towards cooperative planning of data mining workflows. In: Proc. of the Third Generation Data Mining Workshop at ECML/PKDD-2009. pp. 1–12 (2009)

[15] Kramer, S., Lavrač, N., Flach, P.A.: Propositionalization approaches to relational data mining. In: Džeroski, S., Lavrač, N. (eds.) Relational Data Mining, pp. 262–292. Springer (2001)

[16] Kranjc, J., Podpečan, V., Lavrač, N.: ClowdFlows: A cloud based scientific workflow platform. In: Proc. of ECML/PKDD (2). pp. 816–819. Springer (2012)

[17] Martins, P., Pollak, S., Urbancic, T., Cardoso, A.: Optimality principles in computational approaches to conceptual blending: Do we need them (at) all? In: Proceedings of the Seventh International Conference on Computational Creativity, UPMC, Paris, France, June 27 - July 1, 2016. pp. 346–353 (2016)

[18] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: Yale: Rapid prototyping for complex data mining tasks. In: Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 935–940. ACM (2006)

[19] Monteith, K., Martinez, T., Ventura, D.: Automatic generation of music for inducing emotive response. In: Proc. of the International Conference on Computational Creativity. pp. 140–149 (2010)

[20] Pereira, F.C., Cardoso, A.: Optimality principles for conceptual blending: A first computational approach. AISB Journal 1, 4 (2003)

[21] Pereira, F.C.: Creativity and AI: A Conceptual Blending approach. Ph.D. thesis, Dept. Engenharia Informática da FCTUC, Universidade de Coimbra, Portugal (2005)

[22] Perovšek, M., Vavpetič, A., Cestnik, B., Lavrač, N.: A wordification approach to relational data mining. In: Proc. of the International Conference on Discovery Science. pp. 141–154. Springer (2013)

[23] Ritchie, G.: Some empirical criteria for attributing creativity to a computer program. Minds and Machines 17.1, 67–99 (2007)

[24] Schorlemmer, M., Smaill, A., Kühnberger, K.U., Kutz, O., Colton, S., Cambouropoulos, E., Pease, A.: COINVENT: Towards a computational concept invention theory. In: Proc. of the 5th Int. Conference on Computational Creativity. pp. 288–296 (2014)

[25] Toivonen, H., Gross, O.: Data mining and machine learning in computational creativity. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 5.6, 265–275 (2015)

[26] Žakova, M., Kremen, P., Železnỳ, F., Lavrač, N.: Planning to learn with a knowledge discovery ontology. In: Proc. Planning to Learn Workshop (PlanLearn 2008). vol. 951 (2008)

[27] Žnidaršič, M., Hervás, R., Alves, A.O., Oliveira, H.G., Xiao, P., Linkola, S., Toivonen, H., Kranjc, J., Lavrač, N.: Computational creativity infrastructure for online software composition: A conceptual blending use case. In: Proc. of the 7th International Conference on Computational Creativity (2016)

[28] Xiao, P., Linkola, S.: Vismantic: Meaning-making with images. In: Proceedings of the Sixth International Conference on Computational Creativity. pp. 158–165. ICCC2015 (Jun 2015)

Graph Theoretical View on Text Understanding

Jure Zupan
National Institute of Chemistry, Ljubljana
E-mail: jure.zupan@ki.si

Keywords: graph theory, cyclic-connected graph, topological distance, network text analysis, information content

Received: October 6, 2017

The system STAVEK-02 described in the contribution is concentrated on yielding supplemental information (besides parsing/tagging of words) for text understanding through the clustering of nouns and/or verbs according to their meanings and common features. The system consists of two word processing blocks. The first block is a vocabulary of 149,000 Slovenian word-roots and 3,100 endings and assigns the grammatical feature to the words by the grammatical rules without any link to pre-tagged lexical corpora. The second block is a Network of meanings of Slovenian words which in principle is a graph connecting 45,000 and 15,000 noun and verb lexemes, respectively, all of them hierarchically clustered into larger and larger groups having /exhibiting specific features and/or common properties of the words encompassed Such formations are in a similar lexical systems usually called synsets. Due to the complete connectivity between the synsets (groups) in the graph it is possible to find all possible property/feature paths between any pair of two words (nouns and/or verbs) in the network. Because clustering of words according to their meanings is made during the parsing of one, a pair, or several consecutive sentences, the features and properties that appear on the closest path between the particular words within the sentence are quite informative for their interpretation of the text. Clustering of the words according to their meanings during the parsing of text is a novel concept of the text interpretation. Ob the basis of a simple example of parsing a sentence and clustering of the nouns within it the concept using the network of meanings in the program STAVEK-02 is described and discussed.

Povzetek: Opisani sistem STAVEK-02 je orientiran na širše izločanje informacij iz slovenskih besedil, kot je samo besedna analiza in označevanje besed. Osnova sta dva programska dela. Prvega sestavljata podatkovna baza (149.000 korenov besed in 3.100 končnic), drugega pa 45.000 samostalnikov in 15.000 glagolov, ki so s skupinami teh besed grupirani po različnih skupnih značilnostih v ciklični graf (connected cyclic graph). Prvi del izvrši slovnično označevanje besed v tekstu, drugi pa med posameznimi besedami, ali v grafu hierarhično povezanih skupin besed (synsets) s podobnimi lastnostmi in značilnostmi izračuna topološke razdalje in nariše shemo povezovanja skupin samostalnikov ali glagolov. Izkazalo se je, da topološko izračunana razdalja med besedami dobro predstavi pomensko razliko/sličnost med njimi. Obe besedni zbirki skupaj vsebujeta in obdelujeta pretežni del najpogostejših slovenskih besed (cca 149.000 slovenskih besed). V prispevku so razložene nekatere pasti slovenščine pri obvladovanju več-smiselnosti besedila. Opisana je tudi struktura cikličnega grafa besed (samostalnikov in glagolov) in način izračuna topološke razdalje med besedami Poudarjena je dvosmernost poti in sprehodov (paths and walks) v omenjenem grafu besed. Dodan je kratek primer analize stavka, ki se konča z matriko topoloških razdalj med besedami stavka in drevesom podobnosti. Na koncu so omenjene nekatere možnosti razvoja sistema STAVEK-02 in hierarhične mreže za določanje pomenov slovenskih besed.

1 Introduction

The parsing or tagging of words in the sentence provides the user with all relevant grammatical features of each word, which itself is a very hard task to implement either by the computer or by hand alone. The fact hat most of the modern parsing programs today rely on large corpora of previously parsed data does not mean that the efforts and programs solving the tagging of sentences by hand are either unnecessary or outmoded. Even if one forget that the testing of parsing-algorithms based on previously parsed corpora first relay on the hand-made parsing, the ab-initio, i.e., parsing by exclusively using grammatical rules will always be necessary. It should not be forgotten that statistical solutions mostly ignore the occurrences of rare specific cases. Such problems can be solved easier by considering and combining both methods (corpora driven and rule-based tagging) consecutively and/or iteratively. For example: the problem of the words having two or more clearly different meanings of which at least two can have grammatically correct but for any kind of machine parsing or rule-based tagging completely indistinguishable forms. Unfortunately, in Slavic languages with a much higher degree of flexibility

of words than in English the problems of the word senses begin already on the parsing level. In the case of a grammatically correct sentence with two completely different interpretations of word senses it is possible that no parsing can correctly identify even the word classes of the constituent words, not to talk about the senses. The possible solution of such problems is to list all possible meanings or senses of each word and leave this information for further consideration when the context of the following sentences allow to single-out the actual meaning. For example, neither the sentence To je dobro za vas nor the title of the well-known Slovenian story Martin Krpan can be tagged correctly by the computer. In the first case the word vas can be interpreted either as for you or, alternatively, as the village, hence, the sentence can mean either: This is good for you, or This is good for the village. In the second example, the title of the well known Slovenian story Matin Krpan introduces the name of the main character. However, the title has, unfortunately, a second grammatically correct meaning of the word Martin, not as a noun (name Martin) but as the adjective meaning belonging to female Marta, which implies that man of the name Krpan is a husband of Marta or at least involved with Marta. Of course, the machine interpretation based on the pre-tagged corpora will always yield grammatically 'correct', i.e., the most often used variation, but at the same time always omit the less probable, but grammatically correct possibilities, witch nevertheless can appear in the spoken or written communication, and should therefore be at least considered. Such cases are handled better by the rule-based tagging compared to the statistical ones.

In order to bring attention to such possibilities and to provide the tool for helping the developers of man-machine dialog to handle such cases the program STAVEK-02 with options of showing *all* grammatical possibilities and additionally provide the user with clusters of various word meanings at each sentence (or group of sentences) was developed and is described in this paper.

2 Related work

The most closely related system to the PMSB (*Pomenska mreža slovenskih besed* [1], (Engl. Network of Meanings of Slovenian Words) used by the program STAVEK-02 is the well-known WordNet [2,3] lexical collection developed by the Princeton University with its graphic visualization VisuWords [4] based on the Thinkmap, data visualization technology. In order to handle the difficulties in the cross-language differences in the meanings of lexical words the Universal Word Net (UWN) Project was launched [5,6]. According to the UWN suggestions and guidelines specific versions for close to 200 different languages are now under development. Similar to the other Slavic languages (see Polish [7], or Bulgarian [8], for example) the Slovenian version named sloWNet [9] is as well progressing. At the moment the version described in the present paper is not included into sloWNet. There are several features of the PMSB that are similar to the WordNet but some of them are not. The organization of synsets for nouns in the hipo- hypero-, mero-, and holonym groups (the word A is a *meronym* of B if A is a part of B; the nose is a part of head, while head is a *holonym* of nose) is very similar, while the verbs in PMSB follow closely the six branch division (*to exist, to have, to move, to do/to, to think/to create, and to sense/to*) as suggested by Vidovič Muha [10] is quite different. The way the distances between the word senses in PMSB are calculated compared to the similarity evaluation between two synsets in WordNet is practically the same: it calculates the length of the shortest path between two nodes in the graph. It is worthwhile to mention that the distance measure used in our case is the length of the shortest path between two nodes (synsets) in a graph. This graph theoretical path distance is not related to the distances between objects (words) represented by the multi-dimensional distributed representations of word vectors as obtained by the word2vector software [11] developed by Thomas Mikolev at Google. The number of words and meanings (synsets), 60,000 and 110,000, respectively, in PMSB is already large enough to cover a large variety of texts.

A considerable difference with WordNet is in the design of our network STAVEK-02. Although the PMSB can act as a stand-alone program in the role of a sort of thesaurus of Slovenian language, its is actually designed as a subroutine to support the system STAVEK-02 which goal is to enhance and/or to improve the machine-man dialog, by pinpointing and/or explaining the *meanings* of specific words.

The mentioned goal can be clearly seen through the selection of hyper- and hyponym groups of the PMSB network which is described in the following paragraph more in detail.

3 Hierarchical Network of Meanings of Slovenian Words (PMSB)

The solution to the discussed information enhancing problem seems to be the organization of words into network of words linked according to the common features or some other commonly present or absent property(ies). Therefore, the links (branches) between nodes in the graph must contain meaningful information about the relation between the nodes they connect. For example: if one node is labeled *tool* and the other one *object* (*man-made*) the link between them must exhibit the property that the first node (synset) labeled *tools* is a part of the second node labeled *all man-made object*) and not *vice versa*. At the same time these two nodes should occupy positions in the work much closer to each other than they have to the synset labeled *insect*, for example. Either individual words or clusters of words could simultaneously be members of several groups (synsets with larger number of meanings) what makes the network to contain cyclic paths (circular paths between clusters) in the structure (Figure 1).

	VERBS (24,626)
Verbs of existing (3,405)	to exist on a specific way (542), verbs to sustain living (1,427), to end existence (299), emission verbs (949), weather verbs (187)
Verbs of having (1,339))	to posses (154), to obtain/take (333), to use possession (288), to negotiate possession (461), to spend possession (102)
Verbs of moving (3,129)	to move (general) (804), to move (specific way) (692), to move (body/parts) (629), to arrive/leave (676), to change movement (206), to do while moving (121)
Verbs of doing (9,663)	to put (2,416), to do (general) (669), to assemble/disassemble (1,340), to change (2,164), to use force/influence (1,322), to do complex tasks (1,751)
Verbs of thinking/creating (1,583)	to create (intellectually) (550), to think (general) (145), to think (specific) (407), to expressing thoughts with symbols (480),
Verbs of communication (5,507)	to exchange of information (2,770), verbs of perception (322), to have/response to feelings (883), verbs of social contact (1,531),
	NOUNS (86,799)
nature (31,988)	**nature (non-living)**(3,130) is divided into: nature (general) (10), nature (phenomenon) (521), nature (physical parameter) (151), nature (space) (82), matter (general) (1,359), matter (Earth) (933), matter (outer-space) (84) **nature (living) (28,847)** is divided into: nature (general/broader) (4,218), nature (plant kingdom) (3,111), nature (animal kingdom) (3,431), nature (human) (18,087)
product (19,222)	**product (origin) (552)** divided into: product (origin (human)) (40), product (origin (nature)) (53), product (origin (plant)) (258), product (origin (animal)) (201) **product (human) (18,670)** divided into: product (human (material)) (13,190), product (human (intellectual)) (5,352) product (human (commodity)) (29), creation (general) (5), creation (limitation) (94)
concept (35.589)	**activity (11,645)** is divided into: activity (general) (101), activity (to do something) (3,507), activity (society) (3,045), activity (emotion) (76), activity (sense) (15), activity (existence) (1,068), activity (movement) (1,240), activity (communication) (1,912), activity (possession) (582), activity (mind) (97) **property (5,943)** is divided into: property (action) (323), property (animal) (45), property (broader meaning) (357), property (company) (17), property (device) (90), property (form) (62), property (general) (37), property (human) (2,774), property (mind) (128), property (matter) (267), property (nation) (35), property (number) (13), property (object) (482), property (phenomenon) (42), property (plant) (34), property (procedure) (390), property (religion) (15), property (ruling) (52), property (society) (111), property (sound) (39), property (space) (309), property (status) (159) property (word/speech) (123), group of properties (38), and 8 other groups: **event (1,208), form (3,169), group (1,958), phenomenon (526), procedure (992), result (5,342), space (1,532), state (2,910).**

Table 1. The first two levels of verbs (upper part of the Table 1) and nouns (lower part of the table) are shown according to their common features. In the parentheses the number of words in each group is given. Because individual word can have several meanings or senses it is listed in as many groups (synsets) as there are meanings. Therefore, the sum of words given in parenthesis is larger than the number of meanings in the network. The largest groups are printed bold.

The PMSB Network consists of 45,000 noun and 15,000 verb dictionary lexemes (words) forming 85,000 and 25,000 different entries of noun and verb meanings, respectively. For example, if *'konj'* (Engl. *horse*) is one of the 45,000 lexemes the four senses of the word '*horse*' in Slovenian language (*horse* – an *animal*, *horse* – a *clumsy man*, *hors*e – a *chess-piece*, and *horse* – a gymnastic equipment, *paddle-horse*) are four of 85,000 noun meanings or senses.

Using the above kind of reasoning, a graph of about their meanings and properties containing close to 4,500 clusters of words (nodes) was generated [1]. The closest collection to our database is the Levine's collection of verb classes [12] and Dornseiff's Wortschatz [13]. There are various Internet versions like WordNet [2,3]) and for the Slovenian language the sloWNet [9]. What the size, i.e. the number of words is concerned; only the Dornseiff's [13] collection has about the same number of verbs (14,000) as our collection. The part of our network

containing verbs is based on six main groups [10] and is already well described in the literature [14,15] and is accessible on the web [16]. The complete structure of verb hierarchy in English language (16,000 verbs and 1000 groups) is given in [17]. The basic division of nouns has three groups: the *product*, the *nature*, and the *concept*. It can be seen from second part of Table 1. The clusters of verbs and nouns in all levels of hierarchy are of very different sizes (Table 1).

On the contrast to the English language, the Slovenian lexical forms of verbs can be well distinguished from those of nouns, however, due to high flexibility of Slovenian declination and conjugation (approximately 20 per each noun, verb, adjective, pronoun, and numeral) there are numerous cases where two or even three word types mix. For example the sentence *To je lepo padalo* has two meanings: a) *This is a nice parachute* and b) *It was falling nicely*. In the first case the word *padalo* is a noun (*parachute*) while in the second case it is the verb (*to fall*). To have all words together in one network (graph) both word types are linked in the network on the highest node.

It is worthwhile to mention that the same word in different languages has different synsets of meaning. This is the reason why such a hierarchy cannot be 'blue-printed' from one to another language. The effect of 'lost with translation' is unavoidable: each translated word could be connected to completely different clusters of words. For example, the English word *plant* in its botanical meaning can be linked with Slovenian counterpart *rastlina*, or German *Pflanze*, but has no connection to the second sense of a production place like Slovenian *tovarna* or German *Fabrik*).

4 Semantic distance measure

Mathematically, the network is a connected cyclic bi-directional graph. Vertices or nodes represent single words, meanings and/or or clusters of words with similar properties/features (synsets). The connected graph enables a continuous walk, described as a sequence of connected nodes (path), between any two nodes. The graph is cyclic if it contains closed paths (cycles), i.e., paths that starts and ends on the same node) with all nodes on that path different (with exception of the closing node). Hierarchical graph has one special node called top node Ntop or root, distinguished from the other ones by defining the orientation of the graph and walk directions within it. All valid paths between nodes must have one of the two directions: either towards the Ntop (up) or backwards from (down). Therefore, each node must have two lists for connections, to up and to down connected neighbors, respectively. Similar to the Ntop which is the last node of all up-paths, so at the end of any down-paths is always a node called terminal, having no down directions. The terminal nodes are individual words or senses if the word has only one sense (meaning).

The fact that the walk path is not allowed to change direction assures that from any node one can always reach either a terminal node or the N_{top}. Thus no walk with the constant direction could be captured in a cycle and thus end in an infinite loop. In the case of update of new words or relocation of nodes the described hierarchy prevents updates to generate infinite loops and self-referencing nodes. All the explained features of our graph offer the advantage of calculation the topological distance between the nodes. The topological distance D_{ij} between two nodes N_i and N_j has all four properties classifying it as a standard metric distance:

1) $D_{ij} > 0$ for all $i \neq j$

2) $D_{ij} = 0$ only for $i = j$

3) $D_{ij} = D_{ji}$, the distance is symmetrical, and

4) $D_{ij} \leq D_{ik} + D_{kj}$ triangle rule for any node k

To evaluate all topological distance D_{ij} between two arbitrary nodes N_i and N_j in the graph, one needs a complete connectivity matrix of order ($N_i \times N_j$). For a graph containing approximately 10^5 nodes this means storing and handling the matrix of about 0.5×10^{10} distances. Fortunately, instead of keeping this large connectivity and/or distance matrix, only two connectivity tables one for keeping all *up* and the other one keeping all *down* connections from each node to neighboring nodes are needed. Using these two connectivity tables it is straightforward to determine topological distance between any two nodes N_i and N_j or words i and j, respectively. The procedure is as follows:

1. Find the complete set $\{P_i\ (N_i,N_{top})\}$ of n_i paths from the node N_i to the node N_{top}.
2. Find the complete set $\{P_j\ (N_i,N_{top})\}$ of n_j paths from the node N_i to the node N_{top}.
3. Compare k pairs of paths from *both* sets

   $\{P_i\ (N_i,N_{top}),\ P_j\ (N_j,N_{top})\}$, $k = 1...n_i(n_j - 1)/2$ and for each pair determine the common node C_k
4. Determine the length l_k of the path from node N_i to the node N_j passing node C_k for each pair k.
5. Keep the shortest path.

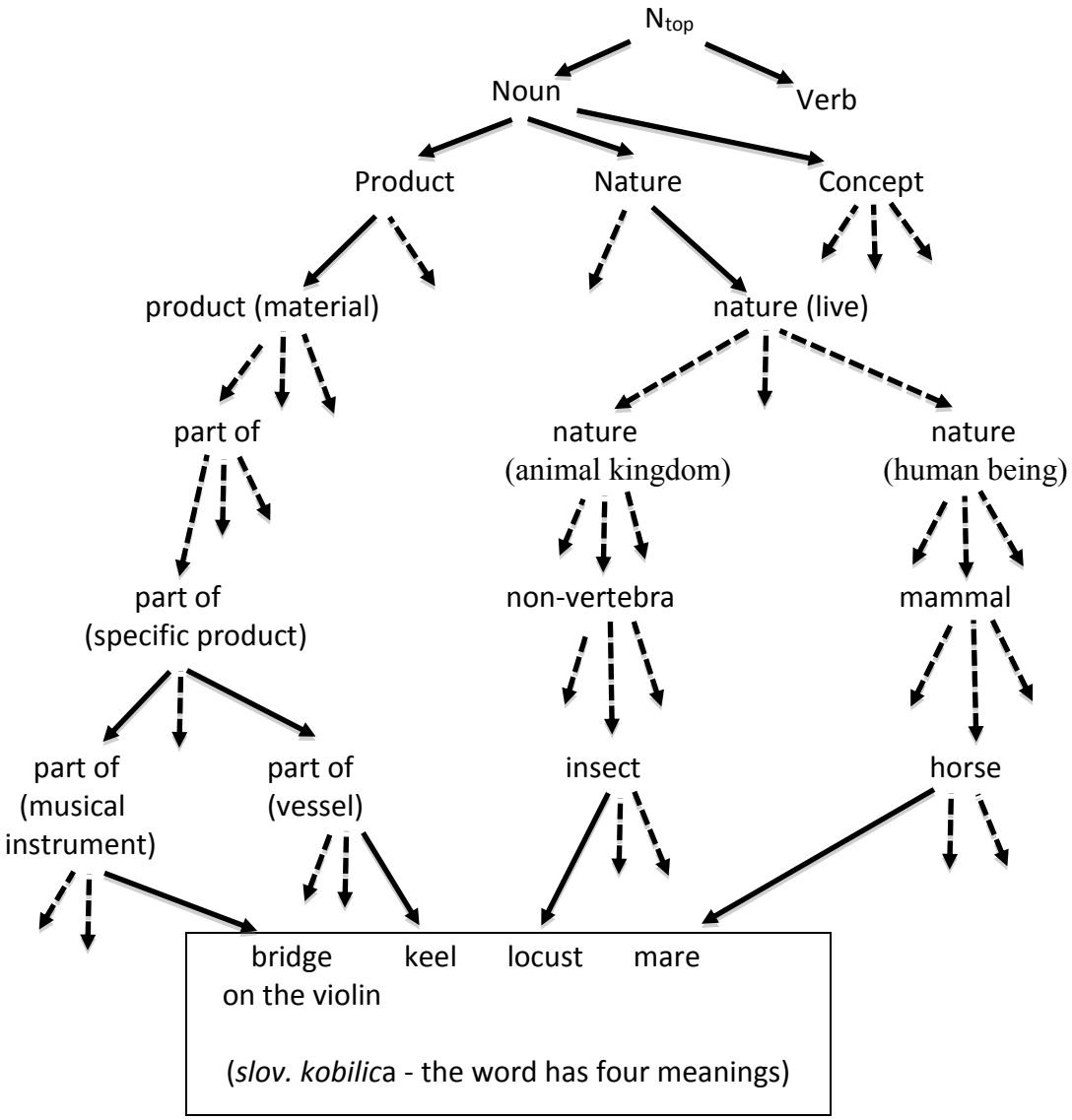
To summarize: the distance D_{ij} between two nodes N_i and N_j is the length l_k of the shortest path from node N_i to the node N_j through the common node C_k, from which both nodes N_i and N_j have access to N_{top}:

$$D_{ij} = \min \{ l_k \} \text{ of } \{P_k(N_i,C,N_j) \},\ k = 1...n_i(n_j - 1)/2 \quad /1/$$

where n_i is the number of *different* paths from the node N_i to the top node N_{top}; and $P(A,C,B)$ is the path from node A to node B passing node C.

5 The Case Study

The described system STAVEK-02 can serve as a model how to use the PMSB hierarchy of word meanings and synsets for enhancing the information in free text. The system can handle individual sentences input by the keyboard or text files of any size. The system handles sentences one by one, hence, the information are reported



.

Figure 1. A simplified part of the discussed network of words showing essential features of a cyclic bi-directional graph. Each label represents a node (synset). A cycle is a path that starts and ends on the same node. From the word *kobilica* having 4 meanings in Slovenian language, six cycles can be drawn to calculate six distances between all four meanings. Because the graph is 2-directional, only the paths in *up* or *down* to the N_{top} (opposite to arrows) or to the terminal nodes (words, along arrows), respectively, are allowed. The cycles are detected *via* the common nodes C_k on the paths.

at the end of each sentence. First, all grammatical information for all words in the sentence, are reported (Part A in Figure 2). This part, of the tagged text is similar in the content, but quite different in the form to the output provided by the public Slovenian parser [18] available on the Slovenian ZRC portal. All tasks performed by the parser are executed *ab initio*, i.e., by the grammatical rules without considering any corpus or web connection. For highly flexible language like Slovenian, the parser is very important, because for tagging of words it must convert the words into appropriate lexemes, what is mandatory form for further search in the PMSB network. Our tagging system uses 149,000 Slovenian word-roots for all types of words and combines them with about 3,100 different endings in order to obtain the correct lemma for any Slovenian word in SSKJ [19].

Second, for each noun and each verb appearing in the input sentence(s) the system searches through PMSB network and lists all the meanings/senses associated with any noun and verb together with the complete list of synsets from the lemmas to the top of the network. The list of senses for all words together with full paths through the PMSB is output after each sentence (part B in Figure 2).

Third, the program draws a clustering scheme of up to 500 nouns or verbs together with all their senses. The cluster-tree of senses is built into the system as an option and can be performed a) after each sentence, b) after a certain number of sentences providing the number of

```
Part A
==================================
 1  Na; na; predlog, (4,5);
 2  violini; violina; sam., f(d/4)(e/5);
 3  je; biti; pom.gl., (3os./f/e) pretekli čas
 4  počila; počiti; gl., nedov. nepov. (3os./f/e) pretekli čas
 5  struna; struna; sam., f(e/1);
 6  . ločilo, pika;
-----
 1  Kaj; kaj; prislov, (vzročnostni);
 2  je; biti; pom.gl., (3os./m/e) pretekli čas
 3  bil; biti; gl, nedov. nepov. (3os./m/e) pretekli čas
 4  vzrok; vzrok; sam., m(e/1)(e/4);
 5  ? ločilo, vprašaj;
-----
 1  Kobilica; kobilica; sam., f(e/1);
 2  je; biti; pom.gl, (3os./f/e) pretekli čas
 3  bila; biti; gl, nedov. nepov. (3os./f/e) pretekli čas
 4  poškodovana; poškodovan; pridevnik, (m/d/1)(m/d/4)(f/e/1)
 5  . ločilo, pika;
==================================

 Part B
==================================
/001/01: violina (violin); strings; instrument (musical (specific)), instrument (musical); product (sound emitting);
         product (communication); product (material); product/creation; Noun; N_top.
/002/01: struna (string): part of (musical instrument)), instrument (musical); product (sound emitting); product
         (communication); product (material); product/creation; Noun; N_top.
/002/02: struna (string); product (sound emitting); product (communication); product (material); product/creation;
         Noun; N_top.
/003/01: vzrok (cause); factor; measure (specific); creation (measure/unit); creation (intellectual); product/creation;
         Noun; N_top.
/004/01/ kobilica (violin's bridge): part of (musical instrument); instrument (musical); product (sound emitting);
         product (communication); product (material); product/creation; Noun; N_top.
/004/02/ kobilica (keel): part of (vessel); part of (specific device); product (machine/device); product (general part);
         product (material); product/creation; Noun; N_top.
/004/03/ kobilica (locust): insect; insect (pterygota); insect (arthropoda); insect (general); antropoda; non-vertebra;
         nature (animal taxonomy); nature (animal kingdom); nature; Noun; N_top.
/004/04/ kobilica (locust): insect; insect (pterygota); arthropoda; polimeria; animal (common name); nature (animal
         kingdom); nature; Noun; N_top.
/004/05/ kobilica: mare; horse (animal (general)); horse (animal); animal (domestic); animal (property); nature
         (animal kingdom); nature; Noun, N_top.
/004/06/ kobilica: mare; horse (animal (general)); horse (animal); odd-toed ungulate; mammal; vertebra; chordata,
         nature (animal-taxonomy); nature (animal kingdom); nature; Noun, N_top.
==================================
```

Figure 2: Output of the program STAVEK-02 after the input of three sentences representing a short dialog. *Na violini je počila struna. Kaj je bil vzrok? Kobilica je bila poškodovana.* (Eng.: *The string on the violin broke. What was the cause? The bridge was damaged.* The word types are nouns (sam.), verbs (gl.), adverbs (prislov), adjective (pridevnik), the letters *m, f, os, e,* and *d* stand for (masculine, feminine, person, singular, and dual), respectively; the numbers mark the falls. Part B shows ten chains of nodes (synsets) of words and meanings from the PMSB network as used for the distance matrix ***D*** and dendrogram calculations (see Figure 3). N_{top} is the top node of the PMSB hierarchy of meanings. In the actual output of program STAVEK-02 the synsets assigned to words of one sentence are printed immediately after one of the main three punctuation marks (full stop, question mark, or exclamation mark) is encountered.

words does not exceed 500, or c) at the end of parsing a text file after the user can selects up-to 500 nouns or verbs from the list of the most frequent word types of the scanned text.

Finally, at the end of each session (either for one sentence or for the text file) the program yields a) statistics of the input text with respect to the word frequencies of all word types and separators, b) the distribution of word-lengths (in characters) of each word

type, and c) the frequency is of 2000 most frequently used nouns, adjectives, verbs, and adverbs.

In order to show the entire procedure more in detail the output as given by the system STAVEK-02 for three short consecutive sentences is worked out and discussed more in detail. The three sentences in English translation are: The string on the violin broke. What was the cause? The bridge was damaged. (slov. Struna na violini je počila. Kaj je bil vzrok? Kobilica je bila poškodovana.) (Figure 2, parts A and B). This particular example using the word kobilica in two separate sentences was chosen deliberately to show how the graph-theoretical distances (Figure 2. and Figure 3) as obtained by the PMSB network could correctly determine the sense of a word. Similar to English the word bridge having several senses, the Slovenian word kobilica has been coded by six synsets in PMSB. It has four (4) main senses (locust, keel, mare, and the bridge on the violin) of which both animal senses have two synset paths for showing the relevant taxonomies of both species. (Figure 2, part B).

Each chain is a sequence of labels of nodes (synsets) encountered during the walk between the word and the N_{top}. The search algorithm finds all possible walks from any encountered noun or verb to the N_{top}. The reader can verify this part of the search engine in real time on-line on the link given in [20]. Mostly, the labels are organized in self-explanatory manner using structure of keywords in which each keyword is itself a cluster label with the link to the particular cluster in the network. For example, the node labeled *property* (*human*) contains words each of which marks a property of a human' (*intelligence, beauty, greed, innocence*, etc.). On the other hand, the words in the cluster with the same two keywords, but ordered differently e.g., *human* (*properties*) describe a human being with a particular property, *genius* and *liar* are in the synsets *human* (*property* (*intelligence*)) and *human* (*property* (*bad*)), respectively. Additionally, both words *human* and *property* are labels of other clusters. The cluster *property*, for example, contains 5,964 nouns with 14 sub-clusters named property ($keyword_i$), i = 1,…14. Each keyword of these clusters: *property* (*animal*), *property* (*human*), *property* (*number*), … *property* (*object*), contains again cluster descriptors with keywords. Take for example the sub cluster *property* (*object*): *property* (*object* (*color*)), *property* (*object* (*form*)), *property* (*object* (*price*)). At the end each $keyword_i$ represents a cluster with a smaller set of words.

Table 3 shows the topological distance matrix $\boldsymbol{D}$ of 45 distances between the ten meanings. All distances reflect the relation between the similarities of meanings of the words concerned very reasonable. The two main groups, the upper one representing material products (*violin, string, bridge on the violin*) and the lower one representing *locust* and *mare*: have two descriptions each, respectively. In the middle of both groups is the word *vzrok* (*cause*), representing the concept of non-material products. In the group of *material objects* the string /002/01/ *(part of the violin*) and *kobilica* /004/01/ (*part of the violin*) are joined at the lowest level. The pair goes together with the second meaning of the string /002/02/ as a sound emitting device and then three join

There is not much to say about tagging shown as part A in Figure 2), however, the tagging the second word violini as singular locative (e5) is a good example showing how the statistical approach ignores the possibility that the word violin has in the dual the same form (for example: 'Pozabil sem na violini' Engl. I forgot about two violins) of accusative in dual (d/4). STAVEK-02 tags both possibilities (d/4) and (e/5). Additionally, the rule-based tagging is considerably faster compared to the statistical pre-tagged-corpora-based one. The public Slovenian parser [18] can tag on the average 8 sentences per second, while the parser of the system STAVEK-02 managed to tag 400 sentences per second. By additionally searching for all noun and verb meanings through the database of close to 110,000 synsets makes the rule-based parser almost two orders of magnitude faster then the public one. Part B shows all the synset paths for the nouns in the sentences. In the actual output the synset paths for verbs are also given. In the print option, the paths are listed after each sentence.
together with the fourth sense *violin* combining all four into a reasonable synset *musical instrument*. As said above, the last four meanings represent the animal synsets (*animal living beings*). To this group of four meanings (*horse* (*domestic animal*)), *horse* (*taxonomy*), *locust* (*insect*), and *locust* (*taxonomy*), there is no counterparts of meanings from the rest of the considered three sentences, hence, one can safely assume that the four meanings of the word *kobilica* do not apply in this context.

It is interesting to see that the remaining two words *kobilica* /004/03/ (*keel* as a part of a vessel) and *vzrok* (*cause*) fit well between the two larger group. The sense *keel* and *violin* are linked together relatively high in the dendrogram because there are both material objects, however, the level of the link between the concept *cause* and the material object *keel* shows that there is still a lot of space for improvements of the procedure for distance evaluation.

This results help us to argue that as much the meanings of single word is important, the distance between the words is important as well. This in turn requires two things; first each word should be represented in unique and uniform way based on various kind of properties and second, the words should bi organized in a system that allows definition of a metrics.

6 Conclusion

The discussed example and hierarchical network of words PMSB present only a very simple and small part of the general solution that can be accomplished by the use of an exhaustive and therefore much more complex network of word meanings. Neither the presented network, nor the presented model for extracting broader information from the text, is the final product. Still a lot of improvements can be implemented.

Although the present network links together slightly more than 60,000 words (nouns and verbs) forming about 110,000 meanings (synsets) of various sizes, it is not the number of words that is a limiting factor, but rather more

factors like the absolute number of synsets (clusters of words with different features), the number of links to which each synset is connected, and least but not last the ability of algorithms for distance calculation to reflect the actual distinction between the meanings of word. These are the issues that should be of first concern. One should add not only more clusters presenting larger variety and number of properties, features, and/or meanings, but as well clusters of words pointing to rare, dangerous, or by any other criterion extreme features that the words represent, for example synsets containing words like non-poisonous plants, extremely hard or non combustible material, etc. The constant updating and enhancement of the networks of meaning require much more man-power and/or machine-supported feature selection efforts for addition of new groups than it has been spent for the present variations of WordNets on varieties of languages. However, for each specific language the native speakers are responsible for the growth and complexity of their specific meaning networks and no automatic procedure could completely replace their manual work and decisions. The presented PSMB network of meanings was put together by hand what requires approximately eight man-years to reach the present size. Some critics are afraid that such knowledge bases has arbitrary structure, because the meanings of the words are subjective and no objective criteria exist how to link or cluster words according to their meanings. The described example has shown the potential of such network to help understanding the context of the communication. As a matter of fact it is true, that such a hierarchy of meanings will always be subjective, but so is human mind.

7 Acknowledgement

The author wishes to thank National Institute of Chemistry for providing him with the facilities to work at the Institute as a research emeritus.

8 References

[1] Zupan, Jure; Koncept mrežnega pomenskega slovarja slovenskih besed, Jezik in slovstvo, 54, (3-4), 2009, pp. 139-151.

[2] Miller, George A, WordNet: A Lexical Database for English. Communications of the ACM. 1995, Vol. 38 (11), 39-41.

[3] Fellbaum, Christiane; WordNet: An Electronic Lexical Database, Editor, 1998, Cambridge, MA: MIT Press.

[4] Visuword™, On-line graphical dictionary and thesaurus, https://visuwords

[5] Towards a Universal Multilingual WorldNet - D5: Databases and Information Systems, Max-Planck-Institut für Informatik; Mpi-inf.mpg.de; 2011-08-14.

[6] Vossen, Piek, EuroWordNet: A Multilingual Database with Lexical Semantic Networks, Editor, 1998, Kluwer, Dordrecht, The Netherlands.

[7] Maziarz M., Szpakowicz S., Piasecki M., Semantic Relations among Adjectives in Polish WordNet 2.0: A New Relation Set, Discussion and Evaluation, Cognitive Studies / Études Cognitives, t. 12, s. 149–179, 2012.

[8] Koeva, S., G. Totkov and A. Genov. Towards Bulgarian WordNet. Romanian Journal of Information Science and Technology, Vol. 7, No. 1-2, 45-61, 2004.

[9] Fišer, Darja, Novak, Jernej. Visualizing sloWNet. Proceedings of the conference on Electronic lexicography in the 21st century: New applications for new users (eLEX2011). Bled, Slovenia, 9-12 November 2011.

[10] Vidovič Muha, Ada, *Slovensko leksikalno pomenoslovje*. Ljubljana: Znanstveni inštitut Filozofske fakultete, 2000:

[11] Mikolov, Tomas, Sutskever, Ilya, Chen, Kai, Corrado, Greg S.; Dean, Jeff; Distributed representa-tions of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 2013.

[12] Levin, Beth; English Verb Classes and Alternations, The University of Chicago Press, Chicago, 1993.

[13] F. Dorenseiff, der deutsche Wortschatz nach Sach-gruppen, 8. Edition, Ed. U. Quasthoff, W. de Gruyter, Berlin, 2004.

[14] Zupan, Jure; Problemi in nekaj rešitev računalniških obdelav slovenskih besedil, Slav. revija, 47 (3), 1999, 277-296.

[15] Zupan, Jure; Hierarhična mreža slovenskih glagolov, v Obdobja 30, Interdisciplinarity in Slovene Studies, Filozofska Fakulteta, Ljubljana 2011, pp. 551-557.

[16] Zupan, Jure; Lajovic, Andrej; PMSG – Network of Slovenian verbs, web address: http://pmsg.zrc-sazu.si.

[17] Zupan, Jure; Pomenska mreža slovenskih glagolov, Založba ZRC SAZU, 2013, pp. 31-51,

[18] Oblikoslovni označevalnik za slovenski jezik, Amebis, d.o.o. Kamnik, Inštitut Jožef Stefan, Univerza v Ljubljani, ZRC SAZU, Trojina, Zavod za uporabno slovenistiko, 2008-2013, konzorcij projekta Sporazumevanje v slovenskem jeziku: link to the network: http//www. oznacevalnik.slovenscina.eu

[19] Slovar Slovenskega knjižnega jezika (SSKJ), Bajec, Anton, et al., Eds., Državna založba Slovenije, DZS, Ljubljana, 1995.

[20] J. Zupan, A. Lajovic; PMSB, Pomenska mreža slovesnkih besed, link to the network of meanings of Slovenian words: http://mreza.andrej.ad-vega.si.

Distance matrix between ten meanings of four words. The distances are the numbers of nodes (synsets) between two meanings in the network PSMB evaluated according to the procedure and equation /1/.

	1	2	3	4	5	6	7	8	9	10
1 violina (violine) /001/01	0	6	6	13	6	12	19	19	18	21
2 struna (string, violin's part)/002/01		0	4	12	3	11	18	18	17	20
3 struna (string, sound emitter)/002/02			0	10	5	9	16	16	15	18
4 vzrok (cause) /003/01				0	12	12	17	17	16	19
5 kobilica (violin's part)/004/01					0	10	17	17	16	19
6 kobilica (keel)/004/02						0	17	17	16	19
7 kobilica (locust) /004/03							0	3	15	16
8 kobilica (locust-taxonomy)/004/04								0	15	18
9 kobilica (mare)/004/05									0	3
10 kobilica (horse-taxonomy)/004/06										0

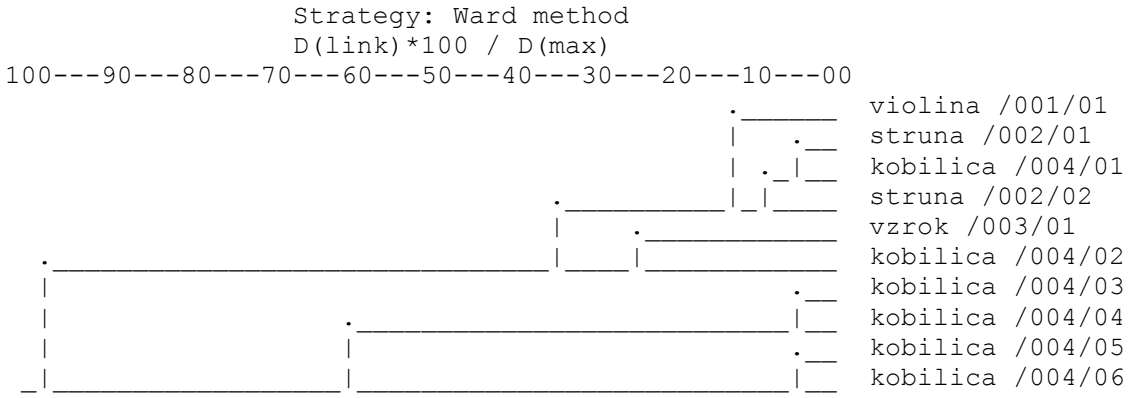


Figure 3. The distance matrix ***D*** between ten different senses of four words (*violin, string, cause and kobilica*). The word string has two meanings a) part *of the violin* and b) *sound-emitting device*. The word *kobilica* has four meanings and six synset paths from the meanings to the top of the network (see Figure 1). The distances between individual meanings are calculated using the procedure and equation /1/. The dendrograms based on the distance matrix ***D*** can be output optionally after any number of tagged sentences providing there is no more than 500 nouns or verbs.

A Segmentation-Recognition Approach with a Fuzzy-Artificial Immune System for Unconstrained Handwritten Connected Digits

Hocine Merabti
LabSTIC Laboratory, 8 May 1945 University, BP-401, Guelma, 24000. Algeria
E-mail: merabti.dr@gmail.com

Brahim Farou and Hamid Seridi
LabSTIC Laboratory, Computer Science Department,
8 May 1945 University, BP-401, Guelma, 24000. Algeria
E-mail: farou@ymail.com, seridihamid@yahoo.fr

Keywords: pattern recognition, optical characters recognition, handwritten digit recognition, handwritten numeral string segmentation, artificial immune system (AIS), fuzzy logic

Received: October 24, 2017

In this paper, we propose an off-line system for the segmentation and recognition of the unconstrained handwritten connected digits. The proposed system provides new segmentation paths by finding two types of structural features. The background and foreground features points are found from the input string image. The possible cutting paths are generated from these features points. Each candidate component is evaluated individually based on its features points and its height. The output of the segmentation module is evaluated using the fuzzy-artificial immune system (Fuzzy-AIS). The latter performs a decision function on the resulting segments, and then the hypothesis that has the best score is regarded as the global decision. The experimental results on the ell-known handwritten digit database NIST SD19 show the effectiveness of the proposed system compared with other methods in both segmentation and recognition.

Povzetek: Razvit je sistem za segmentiranje in prepoznavanje ročno pisanih števk.

1 Introduction

The handwritten numeral string recognition has become a very open research area since their introduction in a wide range of application areas such: indexing and automatic processing of documents, automatic processing of bank checks, and automatic location of addresses and postal codes [1]. The aim of these applications is to reduce the manual effort involved in these tasks.

Handwriting recognition can be divided, according to the nature of the input, into two categories: on-line and off-line [2]. In the on-line case, the handwriting is produced by a pen or a mouse on an electronic surface and acquired as a time-dependent signal. In the off-line case, the handwriting is scanned on paper. Due to the variation in writing styles and the presence of overlapping and touching characters, the off-line recognition presents a good deal of challenging problems.

For building such off-line recognition system, the first step is the acquisition of the numeral string image followed by pre-processing operations on this image. Afterward, each numeral string is segmented into individual isolated digits. Finally, these digit images are sent to the classifier which assigns the corresponding class [3, 4, 5]. The segmentation of a string into isolated digits becomes one of the important challenges of handwritten recognition systems. Indeed, a very good recognition system can be practically useless when text identification and segmentation are performed poorly [6].

The segmentation problems are mainly related to several factors. First, the slope of the images or the noise introduced by the scanner. The variability in writing style and the inking defects caused by scripters. The variability and complexity of the character string shapes illustrated in the overlapping or the joining of two consecutive digits. Second, we do not know the number of characters in the string, and consequently, the optimal boundary between them is unknown [6].

To overcome these problems, many proposed solutions combine the segmentation and recognition processes. Performing a correct segmentation of an image involves knowing what it contains. On the other hand, if the recognition of the content of an image is correct, it means that the system has all the necessary information for the segmentation process.

The segmentation process can be divided into two classes: segmentation-then recognition and recognition-based [7] (Fig. 1). In the first class, the segmentation module tries to separate the connected characters by building a segmentation path. The latter contains a unique sequence hypothesis, and each subsequence should contain a single character to be submitted for recognition [8, 9]. In the second class, the process provides a set of segmentation hypotheses and defines the segmented digits by performing recog-

nition of each provided segmentation hypothesis [10, 11]. This kind of approach gives good results because it provides several hypotheses that increase the classifier choice to find the correct recognition [6]. The segmentation can also be either explicit or implicit, as seen in Fig. 1. In the explicit methods, the segmentation is carried out prior to the recognition to provide candidate digits for the classifier [12, 13]. However, in the implicit methods, the segmentation is embedded in the recognition process, and it is performed simultaneously with recognition [14, 15]. Several works have proposed segmentation algorithms based on these two methods in the last few years. The literature has also shown that implicit segmentation offers very interesting perspectives, but explicit segmentation achieves better results [6, 16].

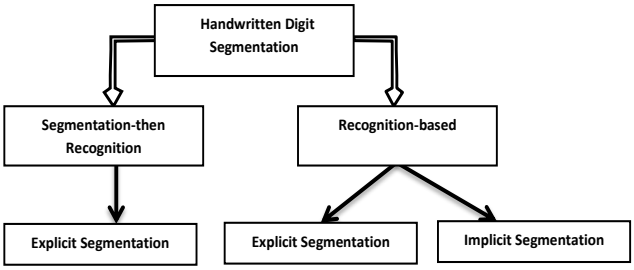


Figure 1: Segmentation and recognition of digits string methods (adapted from [6]).

Usually, segmentation can be conducted by the examination of the following three cases: connected digits, overlapped digits or distinct digits (as shown in Fig. 2). From these problems and in most instances, the connected and the overlapped digits are the most frequent situations observed in handwriting. Also, many algorithms have been proposed to deal with these situations [6, 17]. Some of them are based on features extracted from background pixels in the image [18], and others on features extracted from foreground pixels in the image [12]. Recently, several algorithms have used a combination of both these features [19, 20].

Figure 2: Main difficult examples; (a) connection (the 3 and the 5), (b) overlapping (the first and the second 5), (c) disjunction (in the 5).

Practically, to build a robust system for segmentation and recognition of connected handwritten digits, it is necessary to find: in the first, new methods to select or reduce the number of segmentation points which optimize the number of resulting segmentation hypotheses. In the second, new methods to eliminate the unnecessary segmentation paths which decrease the rejection rate. Finally, using the accurate classifiers on this type of data for keeping or increasing the recognition performance.

In this paper, we propose a new segmentation-recognition approach for handwritten numerical strings. Our work is focused on segmentation and recognition of connected digits, which present the main problematic in the segmentation through: selecting new features points for segmentation, evaluating the segmentation hypotheses to get more precise candidate segments, and using a good classifier for recognition. In segmentation process, we provide segmentation paths for separating the touched digits. This process is based on combining features from the background and foreground of the image. These features are used as segmentation points in the image. The fuzzy-artificial immune system (Fuzzy-AIS) is used for selecting the best segmentation hypotheses and properly classifying the separated digits. It also eliminates the arbitrary assignments in the decision phase when the dataset is overlapped, or the characteristics of objects are almost similar.

The paper is organized as follows. Section 2 presents a description of the proposed method. Section 3 is devoted to the experimental results. Finally, Section 4 concludes the paper.

2 Description of the proposed method

Our system consists of several stages: pre-processing, segmentation, feature extraction, and classification. The pre-processing module aims to remove the noise in the connected digits images and to simplify their further processing. The segmentation module allows providing the best set of candidate cutting paths for the input image and segmenting them into isolated individual digit images. The feature extraction module extracts some statistical and structural features from each digit image and represents them in a feature vector. Finally, the resulted features vectors are sent to the Fuzzy-AIS, and the corresponding class labels are assigned. An overview of the proposed system is shown in Fig. 3.

2.1 Pre-processing

The pre-processing module is applied to the strings image to eliminate or reduce the noise and to simplify the further processing. This module includes smoothing, binarization, dilation, and erosion.

- The smoothing is used to reduce the noise in the image.
- The binarization converts the image to a black and white.
- The dilation and erosion aim to close the disjoint edges and to smooth the global edges of the image.

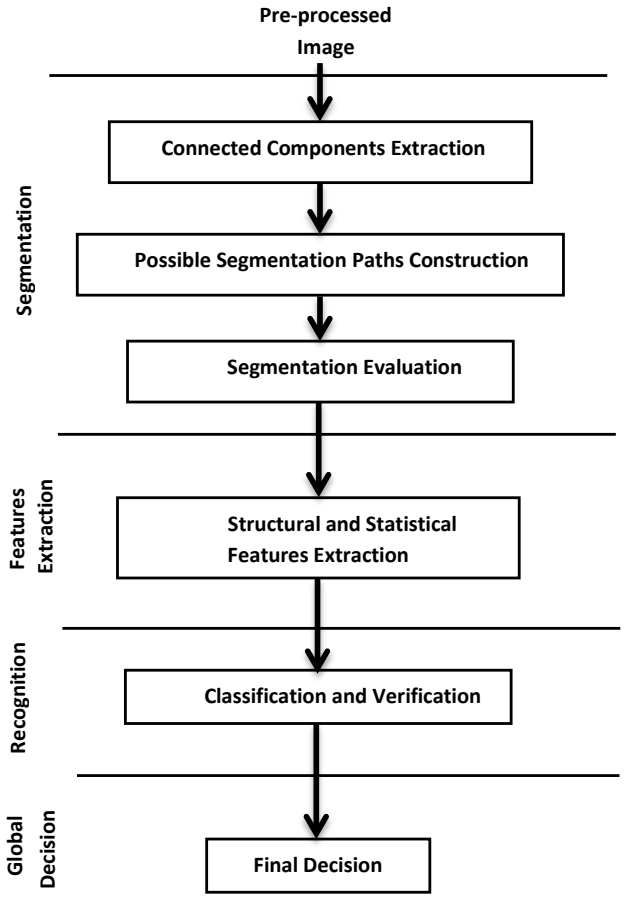


Figure 3: Flow diagram of the system.

Figure 4 shows a sample of the used database before and after the pre-processing stage.

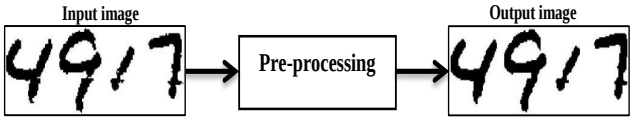


Figure 4: A sample before and after the pre-processing module.

2.2 Segmentation

The aim of the segmentation module is to segment the input string image into isolated digit images by providing the best set of candidate cutting paths. This module consists of three main steps: connected-components-extraction, touching connected-components-identification, and cutting paths constructing and evaluation of connected components (CCs) as seen in Fig. 5.

In the first step, the input image is separated into CCs. The second one allows detecting if a CC contains touched components (TC) or not by checking the following equation:

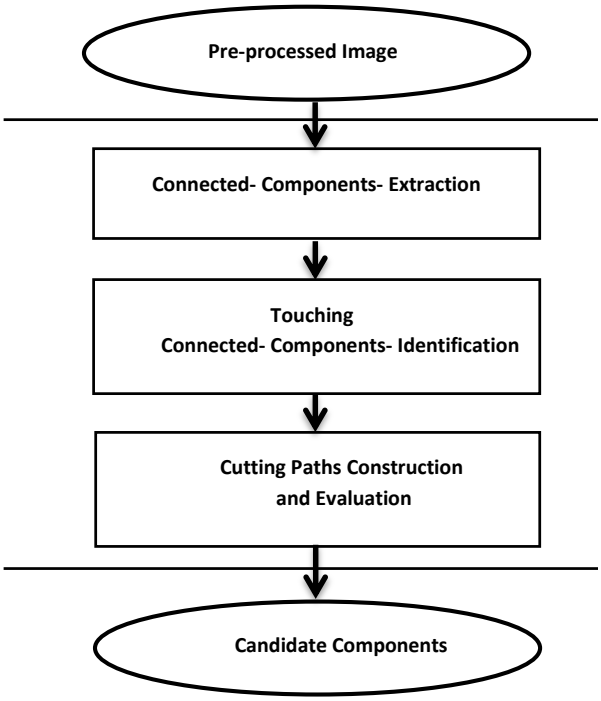


Figure 5: Block diagram of the segmentation module.

$$TC = \begin{cases} 1, & \text{if } W_{CC} > \frac{\alpha * H}{100} \\ 0, & Otherwise \end{cases} \quad (1)$$

where, W_{CC} is the width of a CC, H is the height of the numeral string image, and α is a predefined parameter set in our case to 75.

In the case where the CC does not contain TC, then this CC is very likely to be a piece of a broken digit or a single digit. Figure 6 shows the extraction and identification process.

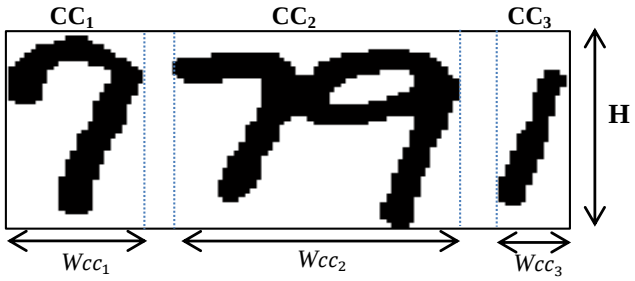


Figure 6: Extraction and identification of connected components; CC_2 with (W_{CC_2}) is higher than $\alpha\%$ of the height (H) of the numeral string image, and need further segmentation. CC_1 and CC_3 with (W_{CC_1}) and (W_{CC_3}) respectively, do not require any further segmentation.

The final step allows providing the optimum position for cutting a CC and extracting the correct candidate components. This step involves analyzing the foreground and background features of the CC to generate the segmentation points, followed by the generation of the possible cut-

ting paths. The evaluation process is used to optimize the resulting segmentation paths and get more accurate results. In the following, we explain in detail, how to construct and evaluate the cutting paths for a CC.

2.2.1 Generating segmentation points

a. Profile features

The method of finding the profile features for a CC is as follows:

- Find the vertical upper and lower projection profiles of the CC, as seen in Fig. 7(b) and (c).
- Extract the upper and lower skeletons of these profiles, which are less and higher than the middle height (H) of CC (see Fig. 7(d) and (e)).
- Extract the end points (PFs) that have just one black neighbor pixel from the skeletons. The first and the last end points in each skeleton will not be considered (see Fig. 7(f)).

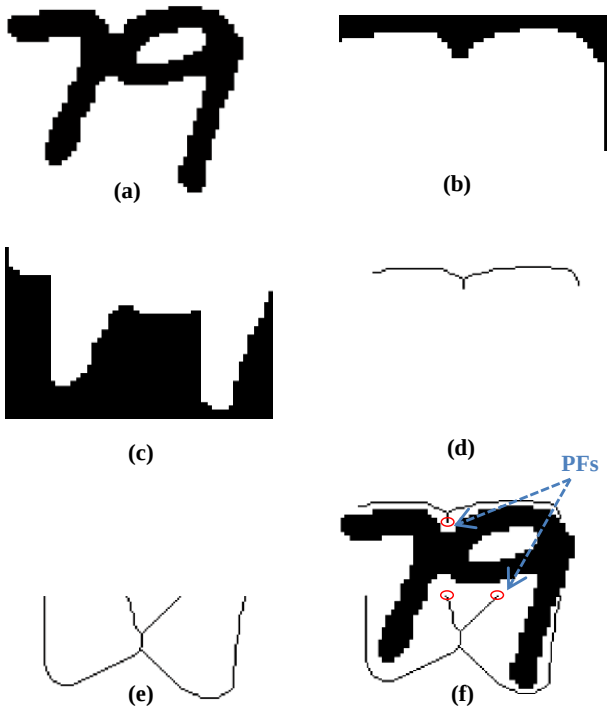


Figure 7: Profile features extraction; (a) Original image, (b) Upper projection profile, (c) Lower projection profile, (d) Upper skeleton profile, (e) Lower skeleton profile, (f) End points of the skeletons (PFs) (denoted by a red circle).

b. Skeleton and edge features

The following steps show how to find the skeleton and edge features:

- Extract the skeleton of the CC.
- Extract the intersection points (SFs) which have more than two black neighbor pixels from the skeleton (Fig. 8(b)).
- Extract the outer edge (upper/lower) from the CC, and add it to the skeleton image (Fig. 8(c)).
- Calculate the distance between the intersection point and the upper edge image, and select the points (EFs) that have the minimum value (Fig. 8(c)).
- Calculate the distance between the intersection point and the lower edge image, and select the points (EFs) that have the minimum value (Fig. 8(d)).

These feature points (SFs and EFs) show the proper location of segmentation regions.

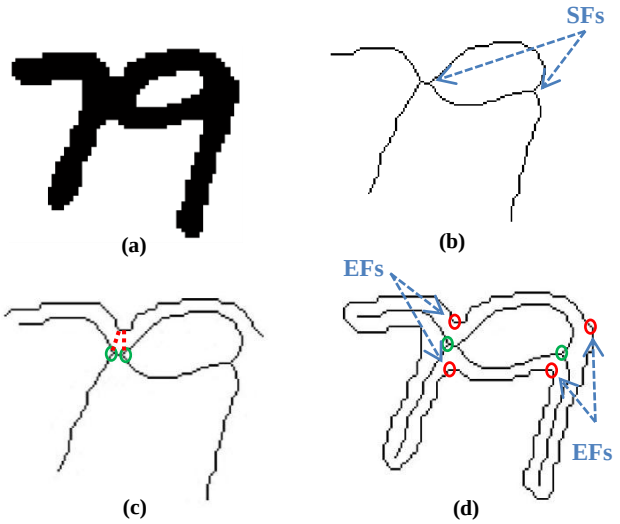


Figure 8: Skeleton (SFs) and Edge (EFs) features extraction; (a) Original image, (b) Skeleton of the CC with intersection points, (c) Upper edge of the CC superimposed on the skeleton image, (d) Edge points (denoted by a red circle).

2.2.2 Generating segmentation paths

All the feature points of the touching digits are found in the previous step. Now, the segmentation path can be generated from these points using two ways: from top to bottom, and from bottom to top. These feature points are connected together to construct the possible segmentation paths (Fig. 9). The two points P_1 and P_2 are connected according to the following equation:

$$| x_{P_1} - x_{P_2} | \leq \mu * (W_{CC}/2) \quad (2)$$

where, x_{P_1} and x_{P_2} are the horizontal coordinates of P_1 and P_2 respectively, μ is a constant parameter set empirically to 0.6, and W_{CC} is the horizontal width of the connected component.

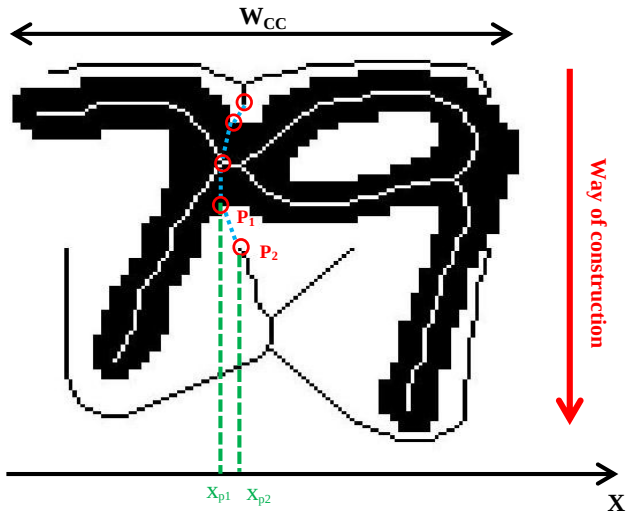


Figure 9: Construction of the segmentation path from feature points (from top to bottom).

The proposed method scans all possible relationships between PF, EF, and SF and generates the related segmentation paths according to the equation (2). Therefore, three hypotheses can be considered for the optimal segmentation path:

• **Hypothesis 1:** If the distance between the projection of the PF and the EF verifies the equation (2), then constructing a vertical segmentation path between these points (Fig. 10(a)).

• **Hypothesis 2:** If there is a skeleton path rather than one SF that linking both upper and lower EFs, this skeleton path is used as part of the vertical segmentation path (Fig. 10(b)).

• **Hypothesis 3:** If the CC does not contain SFs, the vertical segmentation path is constructed between PFs and the closest points PFs (Fig. 10(c)).

During the segmentation process, the segmentation paths may produce outliers: over-segmented parts (out-of-class) or under-segmented parts (non-digit patterns). The resulting segments that contain at least one outlier digit must be rejected using the evaluation of segmentation process.

2.2.3 Evaluation of the segmentation

After finding all possible segmentation paths, each one divides a CC into two new candidate connected components. At this stage, each candidate path is evaluated individually by using two constraints to evaluate our segmentation method and to get more precise results. The first constraint is related to the features points, while the second one is related to the height:

• **Constraint one:** if a candidate component is inside two possible segmentation paths with the same start and end points, then this candidate component is rejected (Fig. 11(a) and (b)).

•**Constraint two:** If the higher of a candidate component is lower than 20% of the height (H) of the image, then this candidate component is rejected (Fig. 11(c) and (d)).

Each segmentation hypothesis divides a CC into two or more new CCs. Now, all the new segments are normalized into a matrix of size 78×64 for preserving their aspect ratio. From each normalized segment, we extract a set of characteristics and represent them as a feature vector. The latter is introduced into the Fuzzy-AIS for the classification.

2.3 Features extraction

In this work, we extracted 39 statistical and structural features from the character. These features are based on Hu moments, zoning features, transitions histograms, and end and crossing points.

• **Hu moments:** seven invariant moments of Hu are computed from normalized and centralized moments up to order three of the segment [21]. They are invariant to translation, scaling, and rotation.

• **Zoning features:** this technique allows dividing the segment into several zones (a grid of $N \times M$), where the features are extracted from each zone [22]. We take the skeleton of each normalized segment, and we divide it into 3×2 zones. For each zone, we extract the density zoning and the gravity center. The density zoning represents the ratio of the number of black pixels on the total size of a zone [23]. The two coordinates of gravity center are used [24].

•**Transitions histograms:** this technique counts the number of transitions from foreground to Background in specified direction (horizontal, vertical and both diagonals $45°/135°$). We extract the mean, the variance, and the max from each histogram.

• **End and crossing points:** the end point is a point that has just one black neighbor pixel. A crossing point connects three or more branches.

After extracting features and representing them in feature vectors, the resulting features vectors are sent to the Fuzzy-AIS for assigning the corresponding class labels.

2.4 Fuzzy-AIS for recognition and verification

An artificial immune system (AIS) is an adaptive system inspired by the principles and functioning of the natural immune system [25]. They are classes of algorithms that have properties and abilities very useful for pattern recognition, especially the classification problem [26, 27]. In our case, we coupled one of the best-known classification algorithms based on artificial immune systems, called the Artificial Immune Recognition System (AIRS) [28], with the Fuzzy-KNN approach.

The principle of AIRS algorithm is as follows: for a given training set of samples from a data class of interest (antigens); the AIRS returns a set of memory antibodies which are used to recognize this class. It is also characterized by:

– Self-regulation: the ability of adaptation and learning,

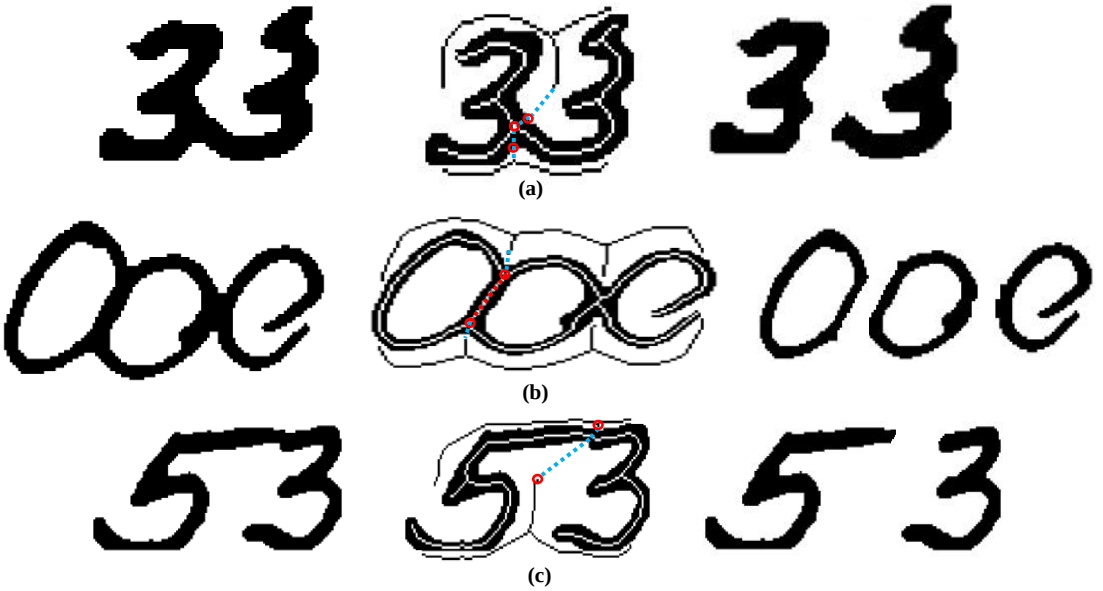


Figure 10: Hypotheses of the segmentation path; (a) Hypothesis 1, (b) Hypothesis 2, (c) Hypothesis 3.

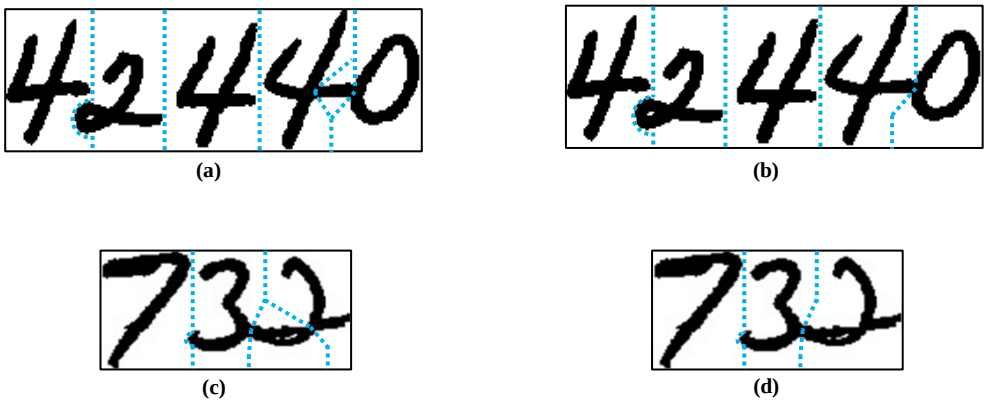


Figure 11: Effect of the evaluation method; (a) and (c) Cases of segmentation before evaluation, (b) and (d) Cases of segmentation after evaluation.

- Competitive performance: their results can be classified among the best works in the classification field,
- Generalization via data reduction: it allows reducing the database on a few training samples,
- Parameter stability: their parameter tuning on different data.

For more detail about this algorithm, the reader is referred to [28, 29].

The similarity measure is one of the most significant design choices in the development of an artificial immune system algorithm, and more precisely in their decision phase. The decision in most artificial immune systems algorithms is provided with the K-Nearest Neighbor approach. The latter has not the ability to correctly assign an object to a particular class when it belongs to other classes with the same value of similarity measure.

The decision will be random in the case when the dataset is overlapped, or the characteristics of the objects are almost similar. To overcome these limitations, the fuzzy concept is introduced in the decision phase, and it lies in the Fuzzy-KNN approach. It ensures that the arbitrary assignments are not made [30].

The Fuzzy-KNN approach finds the k Nearest Neighbors of the candidate component. Each candidate component D belongs to a class i with a membership value $mv_i(D)$. The latter depends on the class of its k Neighbors, and it is given by:

$$mv_i(D) = \frac{\sum_{j=1}^{k} mv_{ij}\left(\frac{1}{d(D,x_j)^{\frac{2}{(m-1)}}}\right)}{\sum_{j=1}^{k}\left(\frac{1}{d(D,x_j)^{\frac{2}{(m-1)}}}\right)} \quad (3)$$

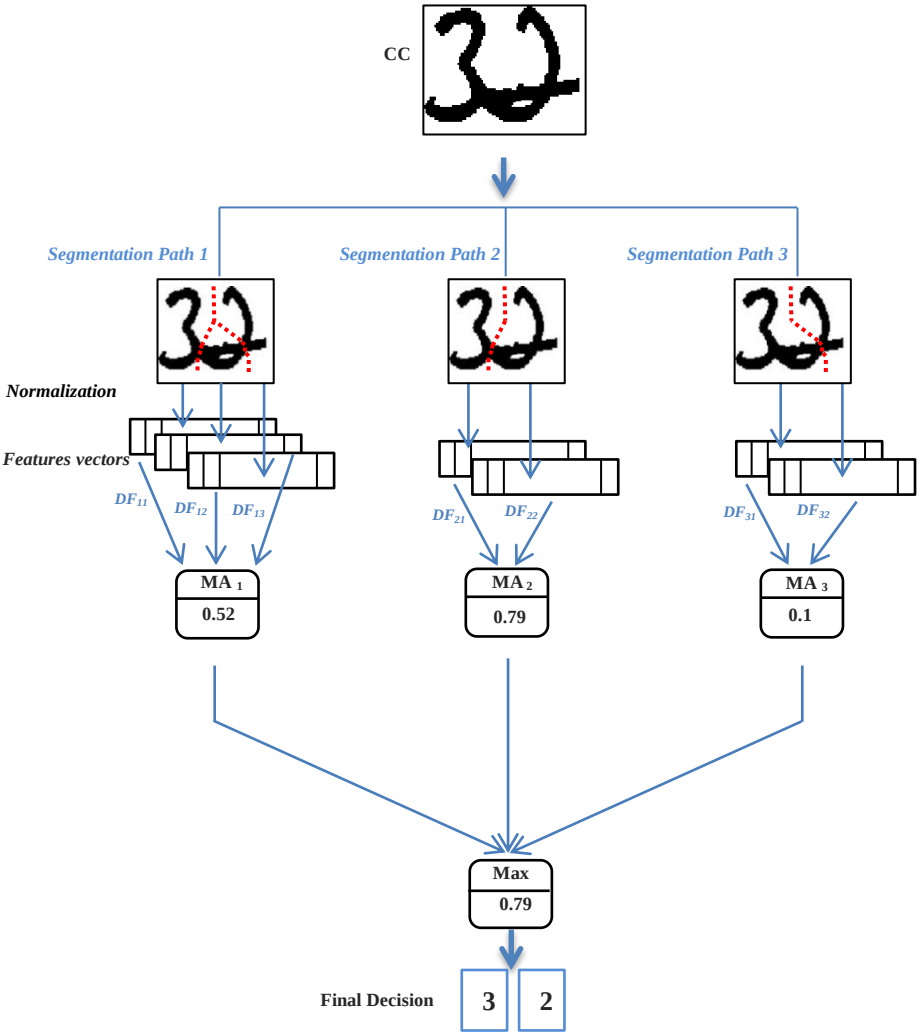


Figure 12: Segmentation followed by the Fuzzy-AIS as a recognition and verification strategy.

where, mv_{ij} is the membership in the ith class of the jth vector of the training set, $d(D, x_j)$ is the distance between D and its jth nearest neighbor x_j. The parameter m determines how heavily the distance is weighted when calculating the class membership.

In this stage, for each candidate component D, the classifier gives a membership to every class and assigns to it the class which has the highest membership value mv. For this reason, the Fuzzy-AIS classifier allows performing a set of decision functions (DF) on the segments of CC according to the following equation:

$$DF = \begin{cases} B * mv, & \text{if } mv < 0.5 \\ mv, & Otherwise \end{cases} \quad (4)$$

where, B is a predefined parameter set empirically to 0.75.

Afterward, the classifier calculates the average (MA) of DFs provided by each hypothesis. Finally, the maximum of these averages is regarded as the final decision function of the classifier, as seen in Fig. 12.

3 Experimental results

To evaluate the proposed method, we perform our experiments on the standard database NIST SD19, which contains unconstrained handwritten numeral strings with various lengths [31]. Our experiments were performed on two stages. In the first, the digit classifier was trained with isolated digit samples. Secondly, the digit classifier was applied to numeral string recognition.

3.1 Isolated handwritten digit recognition

In this stage, we divided the used database into two sets: a set of 2000 isolated digits used for the Fuzzy-AIS learning, and a set of 1500 isolated digits used for testing. The first stage of the Fuzzy-AIS learning consists in performing several tests to initialize the parameters: clonal_rate, hyper_clonal_rate, hypermutation_rate, mutation_rate and Affinity_threshold_scalar. These parameters are necessary for calculating the clones number, the ARBs resources, and the

mutation function. The parameters selection of our classifier is shown in Table 1.

Fuzzy-AIS Parameters	Values
Clonal_rate	10
Hyper_clonal_rate	4
Mutation_rate	0.1
Hypermutation_rate	15
Affinity_threshold_scalar	0.01

Table 1: Parameters selection for Fuzzy-AIS.

After the training process, we obtained a recognition rate of 98.70% on the testing set. The main target of this work is to evaluate the performance of foreground and background features with the Fuzzy-AIS. Indeed, we are not trying to train the classifier with no digits, to optimize their accuracy or to compare the result with other works. In the next stage, we will discuss these issues and compare the performance of our system with other works.

3.2 Handwritten numeral string recognition

Our experiments were performed in two phases. In the first, we examined the performance of our segmentation module without using classification information. In the second, the segmentation is integrated with the recognition process to construct a segmentation-recognition system.

• In the first phase, we perform some experiments on the 3000 string images of the NIST SD19 database for evaluating our segmentation module. All images contain touching pairs of digits, but the module does not know the length of the string. Figure 10 shows some of the results of our segmentation module and Table 2 illustrates their performances.

As shown in Table 2, after the segmentation module, we made a visual analysis and verified in 95.86% of cases, the best segmentation path is among the paths generated by the module. In this case, the module does not know the length of the input digits string and some images produce more than one cutting path (see Fig. 13(a) and (b)). In 1.77 % of cases, the correct segmentation path is not among the produced paths, so we consider these cases as errors (see Fig. 13(e) and (f)). In 2.37 % of cases, the segmentation path is not produced on images; we consider these cases as rejected images (see Fig. 13(g) and (h)). The error and rejection cases are related to the overlapping connected digits. Among 95.86% of the correct segmentation paths, 87.3 % of them have only one segmentation path (see Fig. 13(c) and (d)).

Cases of segmentation path	Visualization (%)
Correct segmentation path	95.86 %
Errors	1.77 %
Rejection	2.37 %
Exactly one segmentation path	87.3 %

Table 2: Performances of handwriting pairs digit segmentation with our method on 3000 images of NIST SD19 database.

A comparison of this result with several segmentation algorithms proposed in the literature in the last few years is shown in Table 3.

Approaches	2-digit Strings Number	Results (%)
StR [32]	2000	88.70
Rb [33]	1000	93.77
Rb [9]	3287	94.8
Rb [34]	2069	95.84
Our Approach	**3000**	**95.86**

Table 3: Performance comparison of several works on touching pairs of digits. Rb: Recognition-based, StR: Segmentation-then Recognition.

Table 3 summarizes a set of segmentation algorithms, declares the number of samples used for testing, and shows their accuracy on touching pairs of digits.

As shown in Table 3, our approach gives good segmentation results in pairs of digits compared with others works.

• In the second phase of our experiments, the recognition module is introduced. It is based on the Fuzzy-AIS approach. We used 2000 images as training samples from the NIST SD19 Database, 200 images per class. For the testing stage, we randomly selected 1500 images. For each string length from 2, 3, 4, 5, 6, and 10, we took 250 images. To determine the performances of the proposed approach, we tested the influence and effectiveness of both: the evaluation method in the segmentation module, and the fuzzy concept in the recognition module. The performance results of our segmentation-recognition system are shown in Table 4.

Table 4 summarizes the recognition rates of our system on numeral strings recognition of lengths 2, 3, 4, 5, 6, and 10 digits. The results in Table 4 show that the use of the evaluation method in the segmentation module improves the performance of the proposed system (Fig. 11). This improvement is visible in both classifiers (AIS and Fuzzy-AIS). The system segment and recognize 96.55% of string samples with the use of AIS classifier, and 96.79% with the use of Fuzzy-AIS classifier. From these results, we notice that the introduction of the evaluation method increased the recognition rate by 11.6% in the case of AIS and 11.4% in the case of Fuzzy-AIS. However, the changeover from the AIS to the Fuzzy-AIS gave a slight improvement by 0.24% in the recognition rate. This is due to the efficiency of segmentation method. To discuss and compare the effective-

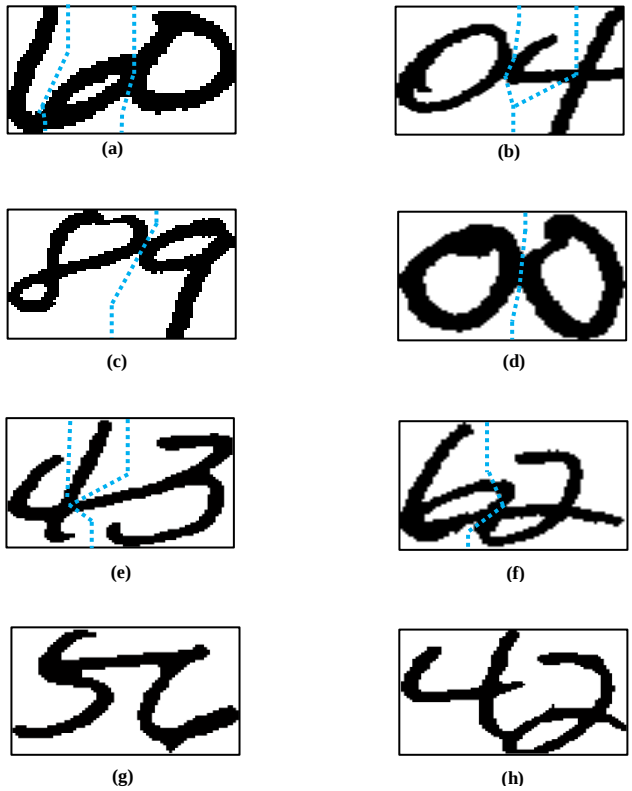


Figure 13: Some results of the segmentation module; (a) and (b) Case of correct segmentation, (c) and (d) Case of exactly segmentation, (e) and (f) Case of error, (g) and (h) Case of rejection.

ness of the proposed approach, we compare our results with others recent approaches on the same database (Table 5).

The results in Table 5 indicate that our system is promising and compare favorably with the other works.

4 Conclusion

In this paper, we proposed a new system to recognize unconstrained handwritten digit strings. We used a segmentation-recognition strategy for handwritten connected digits based on structural features and the Fuzzy-artificial immune system. First, we combined the background and foreground analysis for extracting the feature points. For the background features, we applied a thinning procedure to the vertical projection profile of the image. For the foreground features, we applied a thinning procedure on the connected component and their edge. These feature points are linked to generate the possible segmentation paths in connecting digits. The resulted candidate segmentation paths are evaluated for removing the useless among them and keeping the best. The evaluation process is based on two main constraints. The first one is related to the features points of the candidate segmentation paths and the second one is related to its height. Finally, we introduced the Fuzzy-AIS classifier for ranking all possible segmentation paths and considering the best of them as the global decision. The introduction of both the evaluation process in the segmentation module and the fuzzy concept in the decision phase allowed increasing the recognition rate.

Our experiments on the NIST SD19 database show that our system gets good results in both segmentation and recognition and compare favorably with other works in the same database.

References

[1] Gayathri, P. and Ayyappan, S. (2014) 'Off-line handwritten character recognition using Hidden Markov Model', *in Proceeding of the International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, pp.518–523.

[2] Lacerda, E. B. and Mello, C. A.(2013) 'Segmentation of connected handwritten digits using Self-Organizing Maps', *Expert Systems with Applications*, Vol. 40, no. 15, pp.5867–5877.

[3] Saba, T., Rehman, A. and Elarbi-Boudihir, M. (2014) 'Methods and strategies on off-line cursive touched characters segmentation: a directional review', *Artificial Intelligence Review*, Vol. 42, pp.1047–1066.

String Length	Recognition Rate (%)			
	Without Evaluation Method		With Evaluation Method	
	AIS	Fuzzy-AIS	AIS	Fuzzy-AIS
2	84.33	85.66	97.33	98.00
3	88.44	88.88	97.11	97.33
4	84.83	85.33	96.33	96.66
5	84.00	84.26	96.00	96.13
6	80.00	80.11	95.22	95.33
10	88.12	88.12	97.33	97.33
Average rates	**84.95**	**85.39**	**96.55**	**96.79**

Table 4: Experimental results of our segmentation-recognition approach.

String Length	Recognition Rate (%)				
	Approaches				
	[16]	[35]	[13]	[34]	Our approach
2	96.88	94.8	98.94	98.57	98.00
3	95.38	91.6	97.23	96.28	97.33
4	93.38	91.3	96.16	96.12	96.66
5	92.40	88.3	95.86	94.73	96.13
6	93.12	89.1	96.10	95.02	95.33
10	90.24	86.9	94.25	90.46	97.33
Average rates	93.57	90.33	96.42	95.63	**96.79**

Table 5: A comparison with others works.

[4] El Kessab, B., Daoui, C., Bouikhalene, B. and Salouan, R. (2014) 'A Comparative Study between the K-Nearest Neighbours and the Multi-Layer Perceptron for Cursive Handwritten Arabic Numerals Recognition', *International Journal of Computer Applications (0975–8887)*, Vol. 107, No. 21.

[5] El Kessab, B., Daoui, C., Bouikhalene, B. and Salouan, R. (2015) 'A comparative study between the support vectors machines and the k-nearest neighbors in the handwritten latin numerals Recognition', *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol. 8, No. 2, pp.325–336.

[6] Ribas, F. C., Oliveira, L. S., Britto Jr, A. S. and Sabourin, R. (2013) 'Handwritten digit segmentation: a comparative study', *International Journal on Document Analysis and Recognition (IJDAR)*, Vol. 16, no. 2, pp.127–137.

[7] Casey, R. G. and Lecolinet, E. (1996) 'A survey of methods and strategies in character segmentation', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, no. 7, pp.690–706.

[8] Shi, Z. and Govindaraju, V. (1997) 'Segmentation and recognition of connected handwritten numeral strings', *Pattern Recognition*, Vol. 30, no. 9, pp.1501–1504.

[9] Yu, D. and Yan, H. (2001) 'Separation of touching handwritten multi-numeral strings based on morphological structural features', *Pattern Recognition*, Vol. 34, no. 3, pp.587–599.

[10] Gattal, A. and Chibani, Y. (2015) 'SVM-Based Segmentation-Verification of Handwritten Connected Digits Using the Oriented Sliding Window',*International Journal of Computational Intelligence and Applications*, Vol. 14, no. 1, pp.1550005.

[11] Fujisawa, H., Nakano, Y. and Kurino, K. (1992) 'Segmentation methods for character recognition: from segmentation to document structure analysis', *Proceedings of the IEEE*, Vol. 80, no. 7, pp.1079–1092.

[12] Pal, U., Belaid, A. and Choisy, Ch. (2003) 'Touching numeral segmentation using water reservoir concept', *Pattern Recognition Letters*, Vol. 24, no. 1, pp.261–272.

[13] Sadri, J., Suen, C.Y. and Bui, T.D. (2007) 'A genetic framework using contextual knowledge for segmentation and recognition of handwritten numeral strings', *Pattern Recognition*, Vol. 40, no. 3, pp.898–919.

[14] Procter, S., Illingworth, J., and Elms, A.J. (1998) 'The recognition of handwritten digit strings of unknown length using hidden markov models', *In Proceedings of the 14th International Conference on Pattern Recognition*, pp.1515–1517.

[15] Choi, S. M. and Oh, I. S. (1999) 'A segmentation-free recognition of two touching numerals using neural networks', *In Proceedings of the 5th International Conference on Document Analysis and Recognition (ICDAR'99)*, IEEE, Bangalore, India, pp.253–256.

[16] Oliveira, L. S., Sabourin, R., Bortolozzi, F. and Suen, C. Y. (2002) 'automatic recognition of handwritten numerical strings: a recognition and verification strategy', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, no. 11, pp.1438–1454.

[17] Kulkarni, R. V. and Vasambekar, P. N. (2010) 'An overview of segmentation techniques for handwritten connected digits', *In Proceedings of the International Conference on Signal and Image Processing (ICSIP)*, IEEE, pp.479–482.

[18] Ayat, N.E., Cheriet, M. and Suen, C.Y. (2000) 'Un systme neuro-flou pour la reconnaissance de montants numriques de chques arabes', *In Colloque international francophone sur l'crit et le document*, pp.171–180.

[19] Oliveira, L. S., Lethelier, E., Bortolozzi, F. and Sabourin, R. (2000) 'A new approach to segment handwritten digits', *In Proceedings of the 7th International Workshop on Frontiers in Handwriting Recognition*, Amsterdam, The Netherlands, pp.577–582.

[20] Sadri, J., Suen, C. Y. and Bui, T. D. (2004) 'Automatic segmentation of unconstrained handwritten numeral strings', *In Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition (IWFHR-9)*, IEEE, Tokyo, Japan, pp.317–322.

[21] Cash, G. L. and Hatamian, M. (1987) 'Optical character recognition by the method of moments', *Computer Vision, Graphics and Image Processing*, Vol. 39, no. 3, pp.291–310.

[22] Hirabara, L. Y., Aires, S. B., Freitas, C. O., Britto Jr, A. S. and Sabourin, R. (2011) 'Dynamic zoning selection for handwritten character recognition', *In Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*,Pucn, Chile, pp.507–514.

[23] Parker, J.R. (1993) *Practical computer vision using C*, John Wiley and Sons, Inc., New York.

[24] Gorgevik, D. and Cakmakov, D. (2004) 'An efficient three-stage classifier for handwritten digit recognition. In Pattern Recognition', *In Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, IEEE, pp.507–510.

[25] Timmis, J., Andrews, P.S., Owens, N. and Clark, E. (2008) 'An interdisciplinary perspective on artificial immune systems', *Evolutionary Intelligence*, Vol. 1, no. 1, pp.5–26.

[26] De Castro, L. N. and Timmis, J. (2002) 'Artificial Immune Systems: A Novel Paradigm to Pattern Recognition', *Artificial Neural Networks in Pattern Recognition*, Vol. 1 , pp.67–84.

[27] Yang, Y. (2011) 'Application of artificial immune System in handwritten Russian Uppercase character recognition', *In Proceedings of the International Conference on Computer Science and Service System (CSSS)*, IEEE, pp.238–241.

[28] Watkins, A., Timmis, J. and Boggess, L. (2004) 'Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm', *Genetic Programming and Evolvable Machines*, Vol. 5 , no. 3, pp.291–317.

[29] Watkins, A. and Boggess, L. (2002) 'A New Classifier Based on Resource Limited Artificial Immune Systems', *In Proceedings of the 2002 Congress on Evolutionary Computation CEC'O2, Part of the World Congress on Computational Intelligence.*, IEEE, Honolulu, HI, USA, pp.1546–1551.

[30] Keller, J.M., Gray, M.R. and Givens, J.A. (1985) 'A Fuzzy K-Nearest Neighbor Algorithm', *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-15, no. 4, pp.580–585.

[31] Grother, P. J. (1995) 'NIST Special Database 19; Handprinted Forms and Characters Database', *National Institute of Standards and Technology (NIST)*.

[32] Suwa, M. and Naoi, S. (2004) 'Segmentation of Handwritten Numerals by Graph Representation', *In Proceedings of the 9th International Workshop on Frontiers in Handwriting Recognition*, IEEE, Tokyo, Japan, pp.334–339.

[33] Ciresan, D. (2008) 'Avoiding segmentation in multi-digit numeral string recognition by combining single and two-digit classifiers trained without negative examples', *In Proceedings of the 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'08)*, Timisoara, Romania, pp.225–230.

[34] Cavalin, P. R. (2006) 'An implicit segmentation-based method for recognition of handwritten strings of characters', *In Proceedings of the 2006 ACM Symposium*

on Applied Computing, ACM, Dijon, France, pp.836–840.

[35] Britto Jr, A. D. S., Sabourin, R., Bortolozzi, F. and Suen, C.Y. (2003) 'The recognition of handwritten numeral strings using a two-stage HMM-based method', *International Journal on Document Analysis and Recognition*, Vol. 5, No. 2-3, pp.102–117.

Load Balancing for Virtual Worlds by Splitting and Merging Spatial Regions

Umar Farooq
University of Science and Technology Bannu, Pakistan
E-mail: umar@ustb.edu.pk

John Glauert
University of East Anglia Norwich, UK
E-mail: j.glauert@uea.ac.uk

Kashif Zia
Sohar University, Oman
E-mail: kzia@soharuni.edu.om

Keywords: virtual world, OpenSimulator, spatial partitioning, under utilisation, over utilisation

Received: July 7, 2017

The aggregation algorithm, an integral part of our dynamic infrastructure (using an expansion and a contraction model) for managing scalable virtual worlds, was proposed in our previous work, to overcome the limitations of the current methods using static and hierarchical approaches. The basic aim was to get two contiguous spaces made of smaller regions while distributing the load as balanced as possible among two servers. This algorithm performs well for the perfect square shaped spaces but fails when it is applied to spaces of other shapes. The current merging algorithms also assign non-contiguous spaces to servers during the contraction phase. This is due to the unavailability of an explicit continuity check in both aggregation and merging algorithms.

In this paper, we provide state-of-the-art in scaling virtual worlds and outline their limitations. It provides both theoretical arguments and simulation results that contiguous spaces have potential benefits. This work, then, extends both the aggregation and merging algorithms and incorporates an explicit continuity check to cope with the issues introduced by allowing non-contiguous spaces. It is demonstrated with the help of results from our prototype that the extended methods strictly achieves the theoretical goals of the proposed methods.

Povzetek: Podan je pregled skalirnih metod v navideznih svetovih (VWs) in nov algoritem za razširjanje in krčenje podprostorov.

1 Introduction

Virtual Worlds (VWs) are the most advanced Virtual Environments (VEs) that allow users to immerse into 3D shared spaces. They provide real or imaginary content and users in them are represented by digital characters called avatars [14]. VWs are interactive and collaborative environments that have distinguishing features such as coherence and persistence. They are general purpose and social in nature [21, 24]. They have attracted huge attention of individuals, businesses, and organisations of various domains such as entertainment, design, government, and research and development communities. They are becoming a major tool for collaborative activities [16, 1]. Second Life (SL) [24, 27] is state-of-the-art in commercial VW development frameworks and it imitates the physical world. It is extensively used for content development by various communities such as business and entertainment industries. The research and development community has, however, shown more interest in OpenSimulator (OSm) [13, 20] - an open source alternative to SL.

Scalability is the major issue to dealt with in VEs. Traditionally, it is achieved by splitting the whole virtual space and assigning it to a set of dedicated servers for simulating it [19]. Game environments are easily scalable as they exploit the concept of sharding that allows the duplication of content [21]. However, the space in VWs, is distributed using spatial partitioning. VWs do not allow duplication of content as they have to maintain a unified coherent space [19, 21]. VWs are very complex as they integrate in them the challenges of many hard simulation problems such as large scale, real time computation and communication using a simulation centric architecture developed for standard simulation environments [21]. Therefore, they are much restricted and are able to host only a limited number of players per Simulator (Sim) [15].

Static and dynamic methods are currently in practice to assign a virtual space to a given set of servers. While a system is up and running, a statically assigned space never changes and manual reconfigurations are required to incorporate changes in current allocation. On the other hand, dynamic spatial partitioning allows re-assignments while

the system is running. This process, however, is too expensive as it involves transferring both content and players. Dynamic techniques are usually categorised into flat and hierarchical mechanisms. Flat mechanisms use either a local, global, or an adaptive strategy for load distribution [4]. Hierarchical approaches adopt a parent child hierarchy for managing resources. In our previous work, we developed a hybrid infrastructure comprises an expansion and a contraction model to cope with the issues in both static and dynamic mechanisms presented in section 2.1.1 and 2.1.2 [6, 8]. It proposed an aggregation and assignment algorithm [7, 9] for the expansion phase and merging algorithms for the contraction phase [6]. The major goal of both types of algorithms was to provide contiguous spaces for a Sim to host.

In this paper, we present the critical analysis of some of the well-known static and dynamic methods currently in use for scalable VEs including our proposed framework. It provides justification for using the continuity in spaces assigned to different Sims. It determines the limitations in both aggregation and merging algorithms and, then, extend them to overcome these limitations. Simple illustrations are used to show that the extended models successfully assign contiguous spaces and avoid non contiguous spaces.

The rest of the paper is structured as follows. Section 2 provides the Literature review, background and motivation for this work. The justification for using the continuity constraint in expansion and contraction phases is provided in section 3. The basic and extended versions of the expansion and contraction algorithms and their illustrations with examples from our prototype are presented in section 4 and 5. Finally, section 6 concludes the paper and provides future directions.

2 Background and motivation

2.1 The Literature

The mechanisms for scaling VEs found in the Literature can be categorised as static, dynamic, and hybrid in nature. These mechanisms are critically analysed in this section.

2.1.1 Static mechanisms

The underlying infrastructures for SL and OSm called SL Grid (SLG) [24], and OSm Grid (OSmG) [20] extend the Butterfly Grid (BG) [17]. They use static assignment for an improved performance and avoid the expensive transferring activities. SL architecture is much restricted and it allows a server (usually, a Simulator (Sim)) to host only up-to a maximum of four regions. OSm uses the extended architecture of SL proposed by the Linden Lab [24] and is, therefore, more open than SL. It allows a Sim to manage an arbitrary number of regions but the environment remains static. SL and OSm both lack dynamic adjustments and, therefore, introduce resource provisioning issues. Resources in this arrangement are greatly misused. Resources in some cases, when no players are visiting the content assign to them, might remain under-utilised - this case is termed as over-provisioning. On the other hand, system capacity is restricted as no additional resources are available when more players are interested to join a space - this is termed as under-provisioning [29].

2.1.2 Dynamic mechanisms

To cope with the issues in static assignment methods, a number of dynamic strategies are developed that are broadly categorised as flat and hierarchical in nature. Load balancing in mechanisms using flat orientation uses either a local, global or an adaptive strategy. Local strategies (such as the one adopted in [25]) are not scalable as each server is capable of sharing its load only with the neighbouring servers. They fail to scale when neighbouring servers are also overloaded. Global strategies (such as those used in [23, 28]) use complex procedures to re-distribute the workload evenly on all the servers and thus degrade interactive user experience. They are not suitable for those systems that involve frequent re-adjustments. Adaptive strategies adopt the simplicity of the local but the scalability of the global strategies. They scale better than local strategy as a server extends sharing its load with the servers next to the neighbouring servers, in case the neighbouring servers are also overloaded. Further, they are less complex than global strategies [22].

VEs prefer using hierarchical approaches which are, generally more flexible and scalable than flat mechanisms, as flat mechanisms put extra burden of user migration on the system [26]. Hierarchical methods (such as those presented in [18, 3, 2, 5]), however, suffer from complexity, latencies, and poor performance as they places no restrictions of the size of content assigned to a server and the levels in a resource management tree [6].

2.1.3 Hybrid mechanism: state-of-the-art in scalable VWs

In our previous work, we presented a dynamic scalable infrastructure and introduced the concept of a hybrid grid infrastructure for its implementation. When the load is normal, this hybrid mechanism behaves like a static grid infrastructure in which each Sim is hosting its assigned space. As the load increases, it dynamically adds additional resources at lower levels to cope with increasing load. The basic aim was to overcome the limitations of existing static and dynamic mechanisms. The proposed mechanism achieves this using an expansion and a contraction model. The expansion phase includes the split, aggregation, and assignment methods. The contraction phase provides two variation for merging process.

In this work, each server in start, handles almost a square shaped space and a regular square pattern is used to split the overloaded space. The number of players a server can potentially host is represented by SimCapacity. However, it

initiates a split operation based on a parameter called SplitCapacity. MergeCapacity parameter is used by a server to initiate a merge operation [8].

The Expansion Phase (Splitting)

The Split Process: When a Sim gets overloaded, it divides its assigned space into an equal sized sub-regions (normally either 4, 9, or 16 onwards) that achieves regions whose density is less than the SplitCapacity and thus eases the load but against a boundary condition. A region representing an un-partitioned but varied size of space is divided during a split operation if it is not the ultimate space that cannot be further partitioned.

The Aggregation Process: uses an aggregation algorithm [9] to determine two aggregates of the smaller spaces comprising an assigned space, provided as input by the Split Process. It aims to minimise resource utilisation, and communication and implementation cost. It tries to obtain aggregates with fair load by combining adjacent regions and avoiding the diagonal ones. It combines only those regions (even those in a diagonal) sharing physical boundaries with the regions already in an aggregate. The main objective is to obtain two contiguous areas for assignment to minimise the number of connections/disconnections between servers when players move between regions. The levels in the management tree are minimised by placing all servers handling regions obtained in a split as siblings.

The aggregation algorithm takes input in the form of a tiled grid. Keeping its goals in mind, it takes any two consecutive corner regions to start aggregation with. It uses four aggregation strategies, namely, **Row by Row (RR)**, **Column by Column (CC)**, **Row and Column in Turn (RCnT)**, and **Row and Column in Turn with Diagonal (RCnTwD)** which guarantee examining the entire set of unique and valuable combinations.

The Assignment Process: assigns one of the aggregates determined in aggregation step to an additional server. The current implementation transfers the aggregate with less regions and smaller number of players. Each server that is hosting an aggregate maintains the identity of smaller regions which are, then, re-assigned at later stages based on an increase in load until each of them is handled by an individual server. The split process is repeated at this stage on smaller regions unless the boundary conditions are met.

The Contraction Phase

The contraction phase implements the merging process and it ensures that the resources are utilised as per the requirements. Merging is triggered by a server when it notices a decrease in the number of players it manages. In current implementation of our work, a Sim is either a parent or a child. However, only a child Sim initiates a merge process.

Contraction allows two merging strategies called, Parent Merge (PntMrg) strategy and Child Merge (ChMrg) strategy. In PntMrg strategy, a child Sim initiates a merge operation only if it can return its full load to the parent Sim. However, it is believed that the system potentially holds the resources for more time and it is not efficient in terms of resources. This issue is resolved in the ChMrg strategy. In ChMrg strategy, a child Sim relocates its full load to one of its siblings, if it is unable to integrate the load with the parent Sim. When a Sim capacity goes beyond MergeCapacity, then it checks for an appropriate Sim (the parent or a sibling based on the strategy being used) and the merging is initiated if and only if, the cumulative load of both the Sims is less than or equal to the MergeCapacity. In case of a successful merge, the Sim who initiated the merge releases itself.

The Implementation, Worth and Limitations of Hybrid Grid Infrastructure

Non existence of a specialised framework for developing highly scalable VWs motivated us to develop the hybrid grid infrastructure. The main goals were to assign coherent contiguous spaces using a resource management tree with minimum additional levels for an improved communication and implementation cost while distributing the load as balanced as possible. We used OSm framework for the implementation of this work. Since, the basic architecture does not support dynamic capabilities, we extended the OSm architecture to support dynamic scalability [10]. We investigated the basic capabilities for various activities involved in the expansion and contraction phases and extended some of the costly activities [11]. We, then, developed a working prototype of this infrastructure using OSm framework by utilising its basic and extended methods [6, 12]. It moves the players in a transferring region into a transit region during the re-allocation process.

Our hybrid infrastructure achieved improvements against both static and dynamic mechanisms in multiple dimensions described using a set of parameters including scale, resource utilisation, complexity, communication and implementation costs, and interactive user experience. When compared with static assignment method that assigns multiple regions to a Sim, the proposed method scale beyond the capacity of static assignment. However, it scales exactly up-to the same capacity as the static method in which each Sim hosts a single region. In both cases, resource utilisation is improved by starting a Sim with more regions and assigning additional resources purely on current workload. The proposed mechanism, therefore, solves over-provision and under-provision of resources. By adopting a localised decentralised approach and reducing the levels in the resource management tree, it greatly reduces complexity, and communication issues. Since, the players never go off, it improves their interactive user experience. Various concepts of OSm framework and the extended methods developed for various activities involved in re-allocation process greatly reduced the implementation and transferring costs.

During the implementation of our work, we discovered that the aggregation algorithm determines the two contiguous spaces when it is applied on a square shaped space. However, it fails to get contiguous spaces when it is applied to spaces of other shapes. Similarly, the merging algorithms also permit a merge of non-contiguous spaces, a clear violation of the basic goals set earlier for our scalable

infrastructure.

2.2 Motivation, goals and contribution

Hybrid grid infrastructure got improvements in multiple aspects discussed above, however, the limitations in its current implementation greatly restricts its functionality. To get hold of the benefits of the proposed infrastructure motivated us to extend its aggregation and merging algorithms. The main goal of this work is to enable these algorithms to produce contiguous spaces for any shape of spaces comprises various regions. It also aims to justify the use of continuity in assigning spaces to servers.

This work reports justification for the contiguous spaces, and the extended algorithms for aggregation and merging followed by their illustrations with results from our implementation.

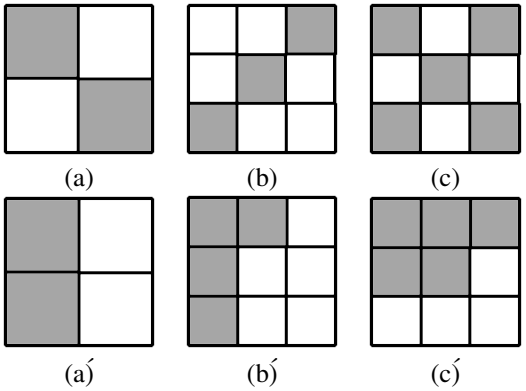


Figure 1: Odd and their equivalent valid aggregates. (a) Aggregates based on diagonals for a 4-region world; (á) Valid aggregates for Figure(a); (b) Aggregates based on a single diagonal for a 9-region world; (b́) Valid combinations for Figure(b); (c) Aggregates based on both diagonals for a 9-region world; (ć) Valid combinations for Figure(c).

3 Evaluating continuity model

This section provides justification and the benefits of the continuity model to be incorporated in basic aggregation and merging algorithms. It shows how odd and isolated cases introduce extra burden in terms of communication, implementation, and user migrations. Three parameters are used for this evaluation that are: total number of regional boundaries exposed to the external regions; total number of isolated regions managed by a single Sim; and number of user crossings between different Sims.

Three example odd cases (from a wide range of possible combinations) which are presented in Figure 1(a)-(c) are used for evaluation and comparison with equivalent valid aggregates presented in Figure 1(á)-(ć). The regions in one aggregate in Figure 1 are marked black and white in the second aggregate.

3.1 Theoretical evaluation

Current VWs treat each region as a complete isolated system and, therefore, introduce complex boundary crossings between the regions regardless of the fact that they might be on a single Sim. The concept of mega-regions is introduced in OSm to get bigger spaces and reduce intra-sim communication. It also help in reducing the number of crossings between the regions. However, the current mega-regions only integrate the neighbouring and contiguous regions. It is difficult to take advantage of this exciting feature of OSm framework when isolated regions are allowed. The inclusion of continuity model in aggregation algorithm thus allow us to get benefit of mega-regions during implementation.

Two parameters that are: the number of isolated spaces managed by a Sim, and the number of boundaries in an aggregate exposed to regions of other aggregate are used to provide theoretical justifications. Table 1 provides results for these parameters where it can be seen that non-contiguous spaces normally provide a large number of isolated spaces. However, the inclusion of continuity model reduced them to only and only two contiguous spaces. Excluded cases greatly increase the implementation complexity by managing different isolated areas compared with valid combinations. Similarly, communication and interaction in valid combinations are significantly reduced compared with odd cases. It can also be noted that when a system has more isolated regions, it generally increases the number of regional boundaries exposed to players of the external regions. It implies that the players have more spaces and chances to go across a Sim boundary to another Sim served by a different server. It potentially increases communication among regions on the same Sim. The next section justifies this claim using a simple simulation environment in terms of players crossing the boundaries between different Sims. Overall, about 50% decrease is achieved in terms of number of exposed boundaries by selecting valid combinations by the extended algorithm as shown in Table 1.

3.2 Simulation based evaluation

The most common parameter in scaling a parallel and distributed system such as a VW is to determine, how much the distribution process increases the number of crossings between the servers in a given system.

3.2.1 Simulation environment

The console window is partitioned into regions based on aggregates and different colours are used to represent the valid and odd combinations as shown in Figure 1. In each case, the odd and its corresponding valid combination are simulated for the same duration against the capacities including one, five, and ten randomly distributed objects (representing players). Each object is allowed to select a random move in one of the four directions at each step

Table 1: Comparison of isolated spaces and their exposed boundaries for both odd and valid aggregates.

Case	Description	Number of isolated spaces	Number of Exposed boundaries
1	Odd Combination (Figure 1(a))	4	4
	Valid Combination (Figure 1(a'))	2	2
2	Odd Combination (Figure 1(b))	5	8
	Valid Combination (Figure 1(b'))	2	4
3	Odd Combination (Figure 1(c))	9	12
	Valid Combination (Figure 1(c'))	2	4

where it moves a character in that direction from its current position. When it reaches either the end of a row or a column, it jumps to the other end of the corresponding row or column. The objects continues following this simple mobility model until the simulation is stopped. A crossing for a player is recorded when it moves to a different coloured region from its current region.

Table 2: Comparison of player crossings for both odd and valid aggregates.

Case	Description	Number of Players		
		1	5	10
1	Odd Combination (Figure 1(a))	5	24	46
	Valid Combination (Figure 1(a'))	2	11	18
2	Odd Combination (Figure 1(b))	9	51	86
	Valid Combination (Figure 1(b'))	5	21	46
3	Odd Combination (Figure 1(c))	18	78	138
	Valid Combination (Figure 1(c'))	7	24	51

3.2.2 Evaluation results

Table 2 summarises the simulation results for both odd and valid combinations. It can be seen in first case, that crossings for odd combination are almost twice the number of crossings for the valid combination. Case 2, has a similar outcome, however, the crossings for odd combination are slightly less than twice the number of crossings for valid combination. This is due to the player distribution, and the ratio between isolated spaces and exposed boundaries for both combinations. It can be seen in case 3, that when there are more isolated regions and exposed boundaries, there are more crossings. The crossings for odd case are almost three times the crossings for valid combinations. Overall, the simulation results revealed that odd cases greatly increases the crossings between the Sims in addition to the implementation complexity and communication overhead. In the next sections, we provide detailed illustrations of the basic and extended algorithms.

4 The extended aggregation algorithm

4.1 Limitations in basic algorithm

The basic aim of aggregation algorithm was to aggregate smaller regions into larger contiguous spaces for assignment. It initially takes regional grids of n×n dimensions as an input normally based on the split strategies of our scalable infrastructure. It repeatedly assigns different parts of the pre-processed space to additional Sims and it has to cope with varied shapes of spaces. In theory, the current aggregation algorithm should always yield valid combinations but in fact 'practically' it allows odd combinations for the non-square shaped grids obtained after the first split and assignment applied to a square grid. During implementation, it failed to discard odd cases in the following iterations. In other words, starting with a square grid, the first iteration determines valid contiguous spaces but in later iterations, when it is applied to non-square shaped worlds, it allows odd cases. Figure 2 illustrates these cases with the help of a simple square grid of nine regions (labelled A to I). The first iteration of aggregation algorithm divides this grid into two aggregates (colours are used to differentiate aggregates from each other) having A and B in the first and the rest of the regions in the second aggregate (see Figure 2(a)). However, when it is applied to the second aggregate (a 7-region world) in second iteration, it selects an aggregate comprises of region C and D, which are both isolated than each other (an obvious odd case), as shown in Figure 2(b). This is because the basic algorithm only uses SplitCapacity constraint but does not check explicitly the continuity constraint for the space comprises of smaller sub-regions. An extension to the current algorithm is presented in the next section to overcome these issues.

4.2 The extended algorithm

In each step of the aggregation process, an additional step is added to make it sure that both prospective aggregates produce valid contiguous spaces. This additional step explicitly use a flood fill algorithm to check continuity in the aggregated spaces. We use flood fill algorithm that spread in four ways as the one that spreads in eight ways consider the diagonals which are major source of odd combinations. Flood fill algorithms are normally used in bucket fill al-

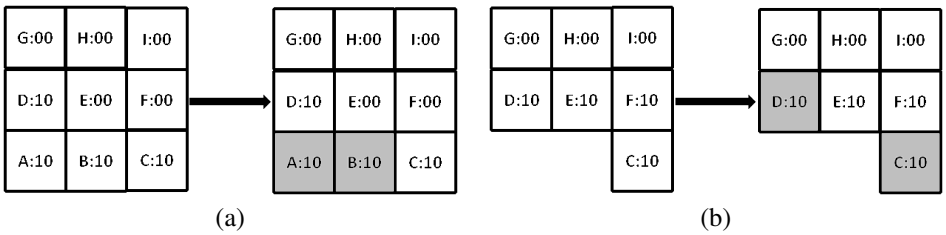


Figure 2: Illustrating limitations in the basic aggregation algorithm: (a) A valid outcome for a square grid of 9 regions; (b) An invalid outcome for a 7-region world.

gorithms of paint programmes, and they are employed in board games such as Go and Minesweeper [30]. In each step, when the possible aggregates are determined by the aggregation algorithm, it checks these aggregates against the continuity constraint, and reject them when any of them are not constituting a valid contiguous space. The extended algorithm has the capability to determine and exclude odd cases against any size and shape of a given space.

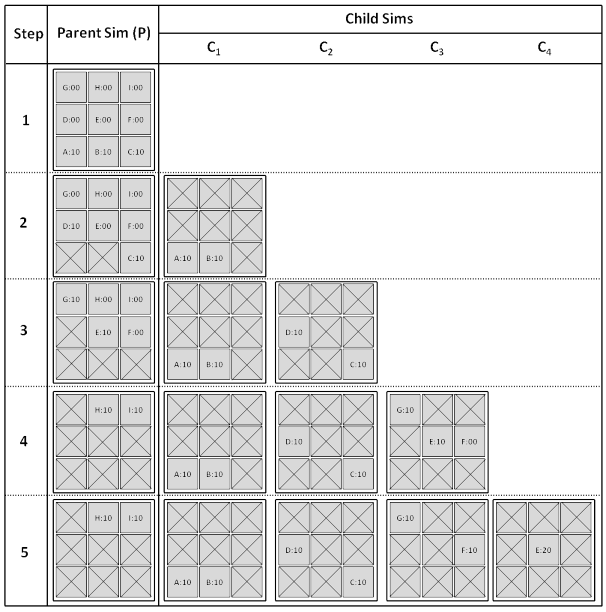


Figure 3: Expanding a 9-region world with the basic aggregation algorithm.

4.3 Illustration and comparison of basic and extended algorithms

In this section, we illustrate the limitations in basic aggregation algorithm and the worth of extended algorithm eliminating issues in the current algorithm with simple player distributions. These example illustrations use a SplitCapacity of 40 players and applies the aggregation strategies to Bottom Left (BL) and Bottom Right (BR) against a 9-region world in a grid form. This article is illustrating only the limitations of current algorithm and it is not demonstrating the aggregation strategies which are presented in detail in [7, 9]. Figure 3 illustrates odd cases allowed by basic algorithm whose equivalent valid combinations which are obtained using the extended algorithm are presented in Figure 4. The partial steps (showing expansion up-to 4 child Sims) shown in these figures are highlighting important points during split and assignment processes. A Sim includes the number of players in each named region that it hosts, and the regions hosted by other Sims are crossed with respect to this Sim.

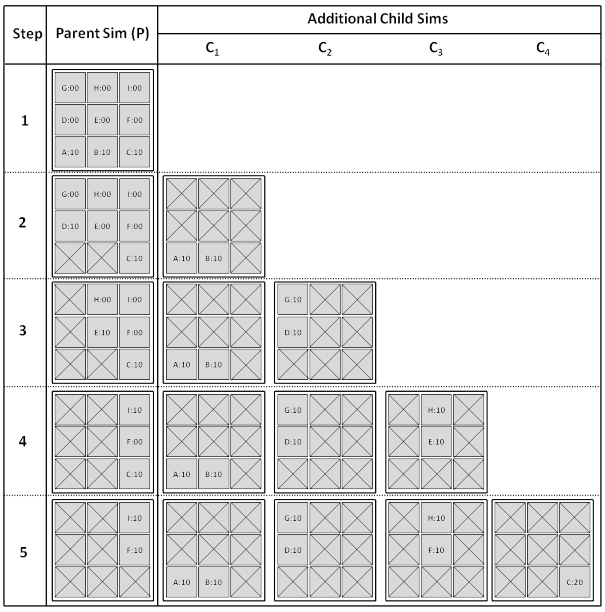


Figure 4: Expanding a 9-region world with the extended aggregation algorithm.

Figure 3, step-1, shows that the parent Sim is initially hosting the whole space comprises of nine regions. In step 2, the space is divided into two valid contiguous groups of regions and then the algorithm assigns the aggregate comprises of region A and B to child Sim C_1. However, the remaining steps assign odd combinations such as in, step 3, the parent Sim transfers region C and D to C_2. Similarly, the parent Sim in step 4, assigns an aggregate of region E, F and G to Child C_3. Further, the child C_3 assign a valid combination to child C_4, but maintains itself a non-contiguous space, in step 5. It is important to note that only the RR strategy of the first root obtained the aggregates assigned in Figure 3.

Figure 4 illustrates the extended aggregation algorithm for exactly the same player distribution used in Figure 3. It is obvious that the extended algorithm strictly allows only valid continuous spaces. The algorithm, in step-2, divides the space into two bigger spaces and assigns the aggregate comprises of region A and B to child C_1, but after verifying the other aggregate being a contiguous one as well. It can be noted in step 3, that the extended algorithm determines the non-contiguous aggregate comprises of C and D to be an odd case and it is skipped by the algorithm to be an acceptable aggregate. The RR strategy was unable to determine further aggregates for a 7-region world at this stage and the algorithm, therefore, applied the CC strategy, which determined a valid aggregate comprises of D and G, in step-3 and assigned it to C_2. Step-4, rejected the assignment of aggregate comprises of region C and E but instead assigned a contiguous space made of E and H to child C_3 determined using the CC strategy. Region C is then assigned during step 5 to child C_4.

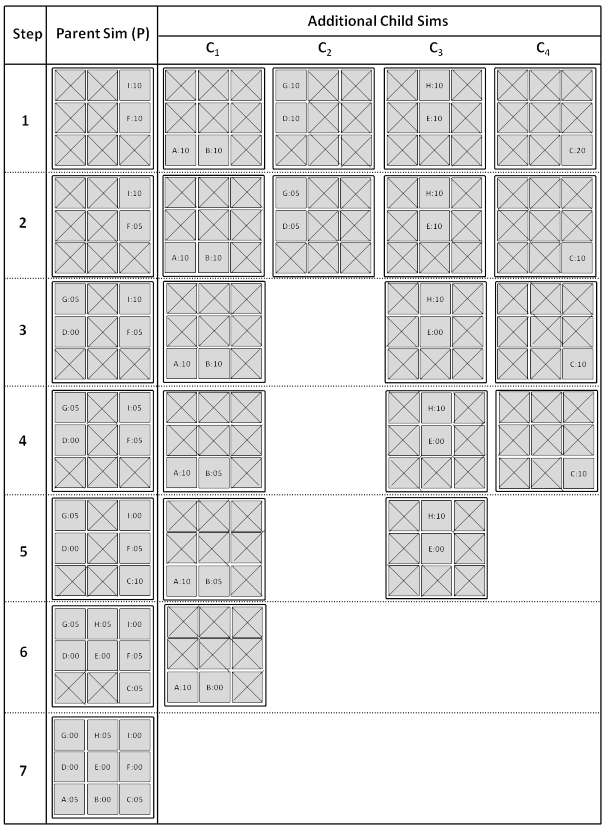


Figure 5: Contracting a 9-region world with the basic PntMrg strategy.

merging preferences either with a parent or a sibling Sim. Both a child and the parent have the capability to determine if a merge operation to be initiated when they notice a decrease in their current capacities but the merging process is always initiated by a child Sim in current implementation. Both the strategies use a MergeCapacity constraint to initiate a merge. Despite the status of a Sim being parent or a child, it first determines, if the cumulative load of both the Sims to combine their load, is less than or equal to the MergeCapacity. On satisfying this condition, the child Sim assigns its complete load (both content and players) to the participating Sim and releases itself.

Both strategies have a flaw (similar to the one for split discussed earlier) that they allow odd combinations while integrating the load which violate the basic goals of our work. The MergeCapacity value of 20 players is considered in this work, a much smaller value to avoid immediate splits.

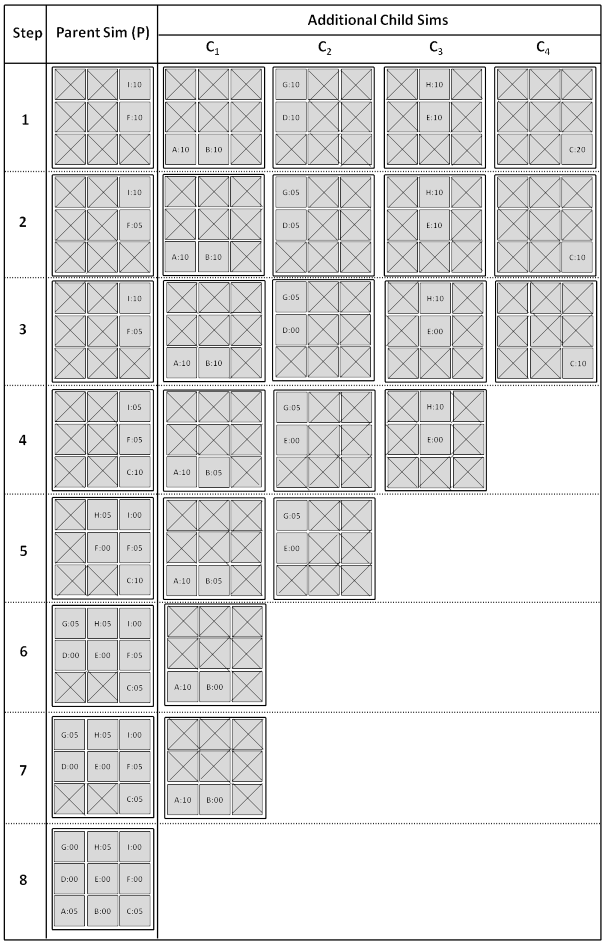


Figure 6: Contracting a 9-region world with the extended PntMrg strategy.

5 The extended merging algorithms

5.1 Limitations of the basic algorithms

The current merging process provides two algorithms (implementing a PntMrg and a ChMrg strategy) that differ by

5.2 The extended algorithms

To avoid assigning non-contiguous spaces, this work also explicitly incorporate an additional step which determines

that either a combined space of two Sims are constituting a contiguous space or not using a flood fill algorithm in addition to the MergeCapacity constraint. A merge is only allowed, if it passes through the continuity check, otherwise, the merge is rejected. This model might use more than required number of Sims for a little longer but it achieves the benefits of assigning contiguous spaces to different Sims.

5.3 Illustration and comparison of basic and extended algorithms

5.3.1 The Parent Merge (PntMrg) strategy

Figure 5 illustrates the basic PntMrg procedure. No merge is permitted with the parent Sim in initial two steps, because the cumulative load in each case is more than the MergeCapacity. However, it is clear in step 2 that child Sims C_2 and C_4 satisfies the merge condition but it is not allowed in PntMrg strategy. Child C_2 integrates its load with the parent Sim during step 3. However, it can be seen that this merge results-in a space comprises of two isolated spaces. In step 4, no merge is allowed though a merge is possible among child Sims C_3 and C_4. The aggregated space after the integration of space maintained by C_4 with the parent Sim in step 5 also gives two isolated sets of regions. The PntMrg strategy potentially holds more resources than required for longer time compared with the ChMrg strategy which tries to overcome this issue.

Figure 6 illustrates the extended PntMrg strategy highlighting the avoidance of odd cases allowed by the basic algorithm as shown earlier in Figure 5. No merge was permitted during the initial three steps. Capacity constraint did not allow merging of child Sims C_1 and C_4 with the parent Sim. The merge between child Sim C_2 and parent Sim at step-3 was rejected due to the continuity constraint. Child C_4 returned its space to the parent Sim at step-4. Child Sims C_3, C_2 and C_1 integrated their load with parent Sim at step 5, 6 and 8 correspondingly. Figure 6 shows that the extended algorithm keep resources for more time than the basic algorithm as illustrated in Figure 5.

5.3.2 The Child Merge (ChMrg) strategy

Figure 7 illustrates the basic ChMrg procedure. No Merge was allowed in step-1, due to the MergeCapacity constraint. However, C_4 was released after merging its load with C_2 at step-2 (a case which was rejected by the PntMrg strategy) but constituting an obvious odd case. Step-3 and 5 obtained valid combinations (the first between the parent and C_3, and the second one between C_1 and C_2) of space after merging, however, it was demonstrated that the basic ChMrg merging strategy allows odd combination.

Figure 8 illustrates the extended ChMrg algorithm that consider both the capacity and continuity constraints for initiating a merge. It always yields contiguous spaces and, therefore, rejected, a merge between child Sims C_2 and C_4. It improves over PntMrg strategy in a sense that it merges quicker by considering the child Sims as in step 4, where

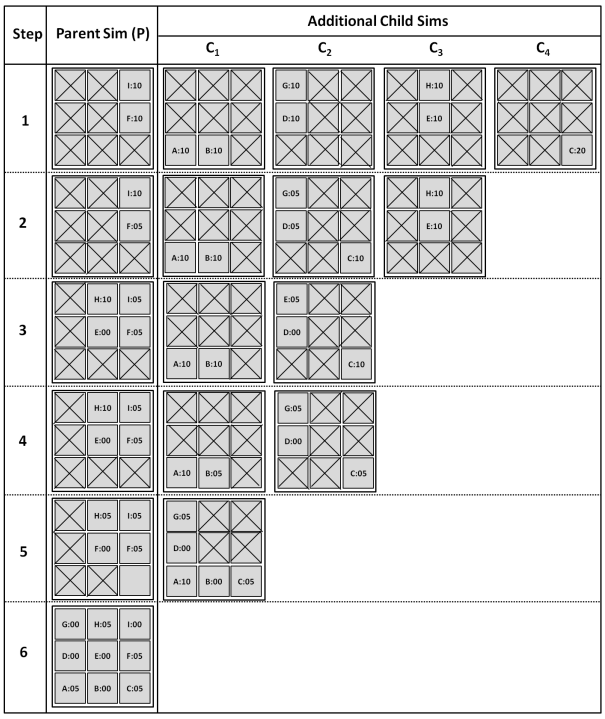


Figure 7: Contracting a 9-region world with the basic ChMrg Strategy.

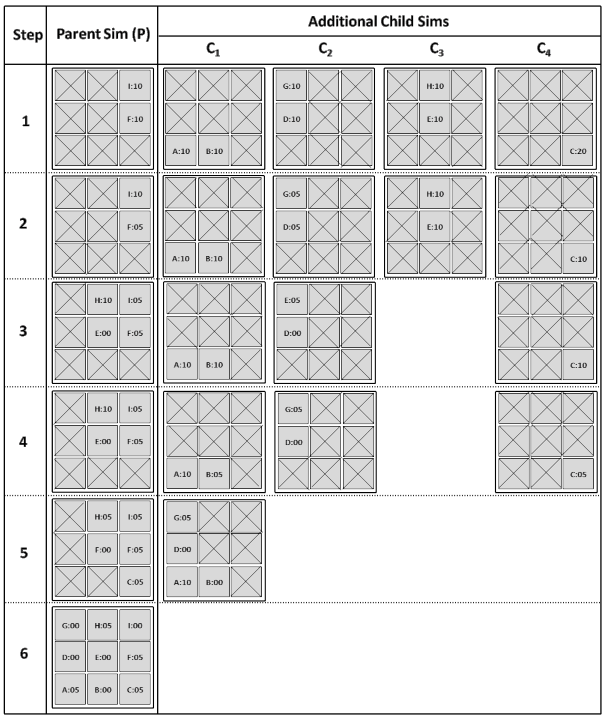


Figure 8: Contracting a 9-region world with the extended ChMrg strategy.

two integrations happened, one between the parent and C_4 and the other between C_1 and C_2. However, it potentially transfers the content and players multiple times which might degrade the overall system performance.

5.3.3 Discussion

The merging strategies (both PntMrg and ChMrg) demonstrated in this work have worth and limitations. Both of them ultimately return the whole world back to the parent Sim. Normally, a merge operation is initiated when player capacity is not high. The PntMrg strategy is simple but takes more time and holds resources for longer than the ChMrg strategy. The ChMrg strategy copes with the issues in PntMrg strategy and release resources much quicker. However, the ChMrg strategy potentially transfers regions (both content and players) between Sims multiple times and it brings a bad experience to the users. We have demonstrated both the strategies, and they could be adopted according to requirements. However, the basic strategies were unable to avoid odd cases. Odd combinations are rejected by both the extended strategies. To manage bigger worlds and the un-predictable nature of users, we suggest using ChMrg strategy as PntMrg might be blocked for longer. However, both have the potential to cope with resource under-utilisation issues. Further details on this are beyond the scope of this article and interested readers may read our detailed work on this in [6].

6 Conclusion

In this article, we presented the extended aggregation and merging processes, to cope with the limitations in basic versions of these mechanisms. It provided an overview of our scalable infrastructure comprises of splitting, merging and load distribution algorithms in comparison with other mechanisms found in the Literature. It examined current and extended operations for both the aggregation and merging, and provided a justification for the continuity model in addition to SplitCapacity and MergeCapacity in their corresponding operations. The extended operations have potential of getting aggregation and merging robustly and they are illustrated with some simple examples from the results obtained from our prototype for scalable virtual worlds.

References

[1] Pekka Alahuhta, Emma Nordbck, Anu Sivunen, and Teemu Surakka. Fostering team creativity in virtual worlds. *Journal For Virtual Worlds Research*, 7(3), 2014.

[2] Rajesh Krishna Balan, Maria Ebling, Paul Castro, and Archan Misra. Matrix: Adaptive Middleware for Distributed Multiplayer Games. volume 3790/2005 of *Lecture Notes in Computer Science*, pages 390–400. Springer Berlin/Heidelberg, 2005.

[3] A. M. Burlamaqui, M. A. M.S. Oliveira, A. M. G. Goncalves, G. Lemos, and J. C. De Oliveira. A Scalable Hierarchical Architecture for Large Scale Multi User Virtual Environments. In *IEEE International Conference on Virtual Environment, Human Computer Interfaces and Measurement Systems*, pages 114–119, 2006.

[4] Luther Chan, James Yong, Jiaqiang Bai, Ben Leong, and Raymond Tan. Hydra: A Massively-Multiplayer Peer-to-Peer Architecture for the Game Developer. In *Proceedings of the 6th ACM SIGCOMM workshop on Network and system support for games*, NetGames '07, pages 37–42, New York, NY, USA, 2007. ACM.

[5] F. Chang, C.M. Bowman, and W. Feng. XPU: A Distributed Architecture for Metaverses. Technical report, Department of Computer Science, Portland State University, 2010. Technical Report 10-04.

[6] Umar Farooq. *The Design of a Contemporary Infrastructure for Scalable and Consistent Virtual Worlds*. PhD thesis, School of Computing Sciences - University of East Anglia, 2012.

[7] Umar Farooq and John Glauert. ARA: An Aggregate Region Assignment Algorithm for Resource Minimisation and Load Distribution in Virtual Worlds. In *NDT '09: Proceedings of the first IEEE International Conference on Networked Digital Technologies*, pages 404–410, 2009.

[8] Umar Farooq and John Glauert. Joint Hierarchical Nodes based User Management (JoHNUM) Infrastructure for the Development of Scalable and Consistent Virtual Worlds. In *DS-RT '09: Proceedings of the 13th IEEE/ACM Symposium on Distributed Simulation and Real-Time Applications*, pages 105–112, Washington, DC, USA, 2009. IEEE Computer Society.

[9] Umar Farooq and John Glauert. A Dynamic Load Distribution Algorithm for Virtual Worlds. *Journal of Digital Information Management*, 8(3):181–189, June 2010.

[10] Umar Farooq and John Glauert. Scalable Virtual Worlds: An Extension to the OpenSim Architecture. In *ICCNIT '11: Proceedings of the IEEE International Conference on Computer Networks and Information Technology*, pages 29–34, 2011.

[11] Umar Farooq and John Glauert. Faster dynamic spatial partitioning in opensimulator. *Virtual Reality*, 21(4):193–202, Nov 2017.

[12] Umar Farooq and John Glauert. Integrating dynamic scalability into the opensimulator framework. *Simulation Modelling Practice and Theory*, 72(2017):118–130, 2017.

[13] Paul A. Fishwick. An introduction to opensimulator and virtual environment agent-based m&s applications. In *Winter Simulation Conference*, WSC '09, pages 177–183. Winter Simulation Conference, 2009.

[14] R. M. Fujimoto, K.S. Perumalla, and G.F. Riley. *Network Simulation*. Synthesis lectures on communication networks. Morgan & Claypool Publishers, 2007.

[15] N. Gupta, A. Demers, J. Gehrke, P. Unterbrunner, and W. White. Scalability for Virtual Worlds. In *Proceedings of the 2009 IEEE International Conference on Data Engineering (ICDE '09)*, pages 1311–1314, 2009.

[16] Marko Hakonen and Petra Bosch Sijtsema. Virtual worlds enabling distributed collaboration. *Journal For Virtual Worlds Research*, 7(3), 2014.

[17] IDC. Butterfly.net: Powering next generation gaming with on-demand computing. Technical report, IBM: An IDC e-Business Case Study, 2004.

[18] Beob Kyun Kim and Kang Soo You. A Hierarchical Map Partition Method in MMORPG based on Virtual Map. In *Frontiers of High Performance Computing and Networking - ISPA 2006 Workshops*, volume 4331/2006 of *Lecture Notes in Computer Science*, pages 813–822. Springer Berlin/Heidelberg, 2006.

[19] Dan Lake, Mic Bowman, and Huaiyu Liu. Distributed Scene Graph to Enable Thousands of Interacting Users in a Virtual Environment. In *Proceedings of the 9th Annual Workshop on Network and Systems Support for Games*, NetGames '10, pages 19:1–19:6, Piscataway, NJ, USA, 2010. IEEE Press.

[20] Charles J. Lesko and Yolanda A. Hollingsworth. Architecting scalable academic virtual world grids: A case utilizing opensimulator. *Journal For Virtual Worlds Research*, 6(1), 2013.

[21] H. Liu and M. Bowman. Scale Virtual Worlds through Dynamic Load Balancing. In *DS-RT '10: Proceedings of the 2010 14th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, pages 43–52, Washington, DC, USA, 2010. IEEE Computer Society.

[22] Qingqi Long, Jie Lin, and Zhixun Sun. Agent scheduling model for adaptive dynamic load balancing in agent-based distributed simulations. *Simulation Modelling Practice and Theory*, 19(4):1021 – 1034, 2011. ¡ce:title¿Sustainable Energy and Environmental Protection SEEP2009¡/ce:title¿.

[23] John C. S. Lui and M. F. Chan. An Efficient Partitioning Algorithm for Distributed Virtual Environment Systems. *IEEE Transaction on Parallel and Distribution Systems*, 13(3):193–211, 2002.

[24] Thomas M. Malaby. *Making Virtual Worlds: Linden Lab and Second Life*. Cornell University Press, Ithaca, United States, first edition, June 2009.

[25] Beatrice Ng, Antonio Si, Rynson W. H. Lau, and Frederick Li. A Multi-server Architecture for Distributed Virtual Walkthrough. In *ACM Symposium on Virtual Reality Software and Technology*, pages 163–170. ACM New York, NY, USA, 2002.

[26] K. Prasetya and Z. D. Wu. Performance Analysis of Game World Partitioning Methods for Multiplayer Mobile Gaming. In *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, NetGames '08, pages 72–77, New York, NY, USA, 2008. ACM.

[27] Michael Rymaszewski, Wagner James Au, Mark Wallace, Catherine Winters, Cory Ondrejka, Benjamin Batstone-Cunningham, and Philip Rosedale. *Second Life: The Official Guide*. Wiley Publishing, Hoboken, New Jersey, December 2006.

[28] Shervin Shirmohammadi, Ihab Kazem, Dewan Tanvir Ahmed, Madeh El-Badaoui, and Jauvane C. De Oliveira. A Visibility-Driven Approach for Zone Management in Simulations. *Simulation*, 84(5):215–229, 2008.

[29] Matteo Varvello, Fabio Picconi, Christophe Diot, and Ernst Biersack. Is There Life in Second Life? In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 1:1–1:12, New York, NY, USA, 2008. ACM.

[30] Wiki. Flood fill algorithm. `http://en.wikipedia.org/wiki/Flood_fill`, 2016. Accessed: December, 2016.

Microscopic Evaluation of Extended Car-following Model in Multi-lane Roads

Hajar Lazar, Khadija Rhoulami and Moulay Driss Rahmani
LRIT-CNRST (URAC No. 29)
Faculty of Sciences, Mohammed V University in Rabat, Rabat 10000, Morocco
E-mail: hajar.lazar@gmail.com

Keywords: car following models, velocity separation difference model, lane changes model

Received: February 4, 2017

This paper describes a micro-simulation model which combined car following with lane change model. For that, we proposed a new car-following model which is an extended of velocity-separation difference model (VSDM) by introducing a new optimal velocity function, named a modified velocity-separation difference model (MVSDM) which react better in braking case. The problems of collision in urgent braking case existing in the previous models were solved. Furthermore, the simulation results show that (MVSDM) can exactly describe the driver's behavior under braking case, where no collision occurs.

Povzetek: Članek opisuje mikrosimulacijski model avtonomne vožnje, ki kombinira sledenje avta z zamenjavo voznega pasu.

1 Introduction

The accelerated growth of the urban population and the extension of cities, the intensification of economic exchanges have made road traffic and its management one of the major challenges of sustainable development. Recently, there has been a strong focus on improving the efficiency and safety of transportation and this has led to the development of the Intelligent Transportation Systems (ITS) (1). Among the most notable urban transport problems:

- Traffic congestion occurs when, at a specific point in time and in a specific section, there is an imbalance between transport demand and supply .
- Environmental impacts includes the pollution and noise problems generated by circulation.
- Accidents and safety problems due to growing traffic in urban areas with a growing number of accidents and fatalities.

In this context, traffic flow modeling and simulation has become a famous area of research in recent years, and constitute efficient tools to evaluate different tasks such as traffic prediction, traffic control and forecasting, the repercussion of the construction of new infrastructure onto the global behavior of the traffic flow. For studying the traffic problem, traffic flow are classified into two different types of approaches, namely, macroscopic and microscopic ones (2). Macroscopic models describe traffic flow as a continuous fluid, which describe entities and their activities and interactions at a relatively low level of detail and established relationships between speed, flow and density. In contrast, microscopic model attempts to model the motion of individual vehicles and their interaction at a high level of detail and describe the reaction of every driver (accelerating, braking, lane changing, etc) depending on the surrounding traffic. Microscopic models are better adapted to the description of more punctual elements of the network, while macroscopic models are adapted to the representation of networks of large sizes. On the other hand, mesoscopic models characterized by the high level of aggregation, low level of detail, and typically based on a gas-kinetic analogy in which driver behavior is explicitly considered (3). Figure 1 presented the different simulation approaches of traffic flow. In this context, we are mainly interested with the microscopic approach which road traffic is modeled by individual motion of each vehicle. In this model, the speed of a vehicle is directly according to the distance that separates it from the leading vehicle, modulo a delay time. This delay time is generally assimilated to the reaction time of the driver in order to take into account the variations in behavior of his leading vehicle. This is a car-following process also known as longitudinal driving behavior. The modeling of traffic in the broader sense proposes to describe more finely the flow of vehicles on a road. For that, it is necessary to understand two behavioral sub-models which are responsible for vehicle movement inside the network: Car Following (CF) and Lane Changing (LC) models. Car-following process were developed to model the manner in which individual vehicles follow one another in the same lane where the driver adjusts his or her acceleration according to the conditions in front and following each other on a single lane without any overtaking (2). The purpose of this paper is to propose a extended car-following model taking into account the effects of lane changing behavior. The work presented in this paper is devoted to overcome the shortcomings such as the unrealistic deceleration and the collisions in braking cases of many existing car-following models. However, we implemented the proposed approach using the open source simulator for traffic flow (4), in order

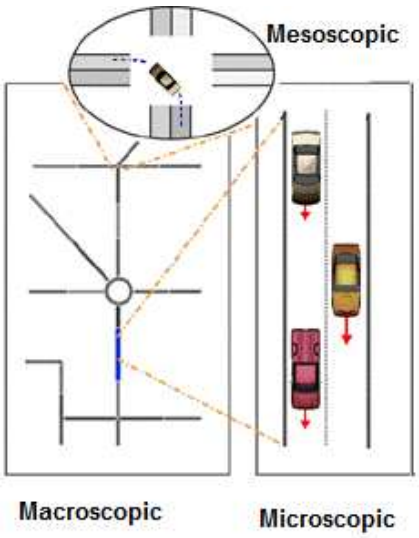


Figure 1: Traffic flow approaches

to improve the efficiency of a proposed approach compared with the existing ones.

The paper is organized as follows. The state-of-the art of car-following and lane changing models will be introduced in Section 2. The proposed approach will be presented in Section 3. In section 4, the simulation results are carried out. At last, the conclusion is given in Section 5.

2 Related work

2.1 Car-following models

The most widely known class of microscopic traffic flow models is so-called the family of car-following or follow-the-leader models. Car-following theories describe the way in which each vehicle follow another in the same lane. The most car-following models have a significant impact on the ability of traffic micro-simulations to replicate real-world traffic behavior (5). Various models were formulated to represent how a driver reacts to the changes in the relative positions of the vehicle ahead. Figure 2 describes the vehicular traffic sketch. We denote as i the car whose behavior is currently under investigation, at instant t, such vehicle is at a position $x_i(t)$, and travels with a speed $v_i(t)$, that means its instantaneous acceleration can be expressed as $a_i(t)$. Index $i-1$ identify the front vehicles with respect to i , which are located at $x_{i-1}(t)$ and travel at speed $v_{i-1}(t)$ at time t. The front bumper to back bumper distance between i and $i-1$ is identified as $S(t) = \triangle x_i = x_{i-1} - x_i$.

Since the 1990s, car following models have not only been of great importance in an autonomous cruise control system, but also as important evaluation tools for intelligent transportation system strategies (6). The car-following models have been designed for single-lane roads, based essentially on the following ordinary differential equation

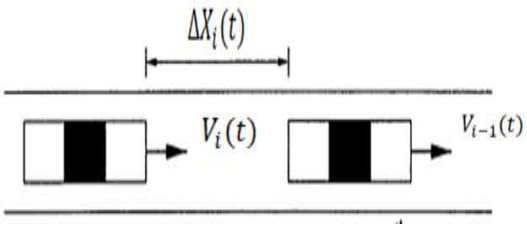


Figure 2: Car following process notation

(ODE):

$$a_i(t) = \frac{v_{i-1}(t) - v_i(t)}{T} \tag{1}$$

This model is based on the idea that the acceleration $a_i(t)$ of the vehicle i at time t depends on the relative speed of the vehicle i and its leader $i-1$ by means of a certain relaxation time T. However the previous equation describes a phenomenon is not stable enough in the case of road traffic. Hence the appearance of several variants of this model includes:

- Safe-distance models or collision avoidance models try to describe simply the dynamics of the only vehicle in relation with his predecessor, so as to respect a certain safe distance.
- Stimulus-response models based on the assumption that the driver of the following vehicle perceives and reacts appropriately to the spacing and the speed difference between the following and the lead vehicles (7).
- Optimal velocity models are another approach generally based on the difference between the driver's desired velocity and the current velocity of the vehicle as a stimulus for the driver's actions.

In this paper, we focused on optimal velocity models and we give here a state of art of the famous ones. For more detailed information with respect to microscopic models, particularly, car-following models can be found in the overview of (5) (8) (9)(10)(11) (12)(13)(14)(15). The optimal velocity models attempt to modify the acceleration mechanism, such that a vehicle's desired speed is selected on the basis of its space headway, instead of only considering the speed of the leading vehicle (16). The first model defined the optimal velocity function using an equilibrium relation for the desired speed as a function of its space headway is (17). The acceleration of Newell model is determined by the following equation:

$$a_i(t) = V_{opt}(x_{i-1}(t) - x_i(t)) \tag{2}$$

Bando et al. later improved this model, by introducing the notion of desired velocity, chosen as a function of relative spacing or headway (18). They distinguished two major types of theories for car-following regulations. The first type called follow-the leader theory which was used by (17), based on the idea that each vehicle must maintain the legal safe distance of the preceding vehicle, which depends on the relative velocity of these two successive vehicles. The other type for regulation is that each vehicle has the legal velocity, which depends on the following distance from the preceding vehicle. Based on the latter assumption, the authors (18) investigated the equation of traffic dynamics and found a realistic model of traffic flow, resulting in the following equation that describes a vehicle's acceleration behavior:

$$a_i(t) = k * [V_{opt}((S(t)) - v_i(t)] \quad (3)$$

In which $V_{opt}(S(t))$ is the optimal velocity function which depends on the headway $S(t)$ to the car in the front. The stimulus here was a function of the relative spacing and the sensitivity k was a constant. The optimal velocity function, generally, must satisfy the following properties: it is a monotonically increasing function and it has an upper bound (maximal velocity). The optimal velocity adopted here calibrated by using actual measurement data proposed by (19) as follows :

$$V_{opt}(S(t)) = V_1 + V_2 \tanh[C_1(S(t) - l) - C_2] \quad (4)$$

With V_1, V_2, C_1, C_2 parameters calibrated and l is the length of the car. Unfortunately, the model produces many problems of high acceleration, unrealistic deceleration and is not always free of collisions. For this reason, Helbing and Tilch proposed an extended model considering the headway and the velocity of the following car and the relative velocity between the preceding vehicle and the following vehicle when the following vehicle was faster than the preceding vehicle (19). To solve the OVM problems, they added a new term which represents the impact of the negative difference in velocity on condition that the velocity of the front vehicle is lower than that of the follower. The GFM formula is:

$$a_i(t) = k * [V_{opt}((S(t)) - v_i(t)] + \lambda H(-\dot{S(t)})\dot{S(t)} \quad (5)$$

Where $H(.)$ is the Heaviside function, λ is another sensitivity coefficient, and $\dot{S(t)} = v_{i-1}(t) - v_i(t)$ means the velocity difference between the current vehicle and the vehicle ahead. The main drawback of GFM doesn't take the effect of positive velocity difference on traffic dynamics into account and only considers the case where the velocity of the following vehicle is larger than that of the leading vehicle (15). The basis of GFM and taking the positive factor $\dot{S(t)}$ into account. In 2001, the authors (20) obtained a more systematic model called Full Velocity Difference Model (FVDM), one whose dynamics equation is as:

$$a_i(t) = k * [V_{opt}((S(t)) - v_i(t)] + \lambda \dot{S(t)} \quad (6)$$

In 2005, the authors in ref (21) introduced a weighting factor which makes the OV model more reactive to braking . They extended the OVM by incorporating the new optimal velocity function obtained by the combination of optimal velocity function Eq (8) with the weighting factor. The modified optimal velocity function expressed as:

$$V_{opt}^{new}(S, \dot{S}) = V_{opt}(S(t)) * W(S(t), \dot{S(t)}) \quad (7)$$

Where the weighting factor is as follows:

$$W(S(t), \dot{S(t)}) = \frac{1}{2} + \frac{1}{2}\tanh B(\frac{\dot{S}(t)}{S(t)} + C) \quad (8)$$

In which B and C are the calibrated parameters. The dynamic equation of the system is obtained as:

$$a(t) = \kappa(V_{opt}^{new}(S(t), \dot{S(t)}) - v_i(t)) \quad (9)$$

In 2006, (6) conducted a detailed analysis of FVDM and found out that second term in the right side of Eq (6) makes no allowance of the effect of the inter-car spacing independently of the relative velocity. For that, they proposed a velocity-difference-separation model (VDSM) which takes the separation between cars into account and the dynamics equation becomes:

$$\begin{aligned} a_i(t) &= \kappa(V_{opt}(S(t)) - v_i(t)) \quad (10) \\ &+ \lambda H(\dot{S(t)})\dot{S(t)}(1 + \tanh(C_1(S(t) - l) - C_2)^3 \\ &+ \lambda\Theta(-\dot{S(t)}(t))\dot{S(t)}(1 - \tanh(C_1(S(t) - l) - C_2)^3 \end{aligned}$$

2.2 Lane changing models

The transfer of a vehicle from one lane to adjacent lane is defined as lane change. Lane change, as one of the basic driver behaviors, can never be avoided in the real traffic environment. Lane changing models are therefore an important component in microscopic traffic simulation Modeling the behavior of a vehicle within its present lane is relatively straightforward, as the only considerations of any importance are the speed and location of the preceding vehicle. Therefore the understanding of lane changing behavior is important in several application fields such as capacity analysis and safety studies. These lane changing models are categorized into four groups:

- Rule-based models are the most popular ones in microscopic traffic simulators include those reported in (22),(23). For this type of models, the subject vehicle's lane changing reasons is evaluated first. If these reasons warrant a lane change, a target lane from the adjacent lane(s) is selected. The gap acceptance model used to determine whether the available gaps should be accepted.

- Discrete-choice-based models based on logit or probit models. The lane changing process is usually modeled as either MLC or DLC. Mandatory lane changes (MLC) are considered those which occur because of a blocked lane, traffic regulations or in order to follow one's route to destination. Discretionary changes (DLC) are made in order for the subject vehicle to achieve better lane conditions (24). Discrete-choice-based lane changing models follow three steps: 1) checking lane change necessity, 2) choice of target lane, and 3) gap acceptance.

- Artificial intelligence models are fundamentally different from the rule-based and discrete choice-based models. A major advantage of them is that they can better incorporate human experience and reasoning into the development of lane changing models.

- Incentive-based models have been recently proposed to modeling lane changing behavior. From their perspective, the attractiveness of a lane based on its utility to the driver, and a safety criterion captures the risk associated with the lane change (25). A variety of factors included in these models such as the desire to follow a route, gain speed, and keep right (26), in addition to politeness factors that can describe the different driver behaviors (25).

In this paper, we describe briefly one the important incentive-based lane changes models. We chose MOBIL (25) as it is the only lane changing model which takes into account the effect of lane change decisions on the immediate neighbors. This model based on the simplistic control rules and it was more appropriate to analyze the affects of usual lane change behaviors of drivers on the overall traffic (24). The lane changing algorithm MOBIL (Minimizing Overall Braking Induced by Lane Changes) is among the most important components of a microscopic traffic simulator based on a microscopic longitudinal movement model. A lane change model depends on the two following vehicles on the present and the target lane, respectively as shown in Fig. 3. A specific MOBIL lane change based on the accelerations on the old and the prospective new lanes.

To formulate the lane changing criteria shown in Fig. 3 we use the following notation: the vehicle i refers to the lane change of the successive vehicles on the target and present lane referred by n (new one in the target lane) and o (old follower in the current lane). The tildes $\tilde{a}_i$, $\tilde{a}_o$ and $\tilde{a}_n$ denotes the new acceleration of vehicle i on the target lane, the acceleration of the old and new followers after the

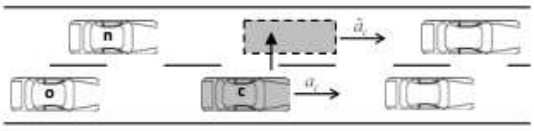


Figure 3: Vehicles involved in lane changing process

lane change of vehicle i, respectively. All the accelerations involved are calculated according to the car-following model (27). A lane change model based on a safety and incentive criterion. The safety criterion is satisfied, if the car-following braking deceleration $\tilde{a}_i$ imposed on the old vehicle o of the target lane after a possible change does not exceed a certain limit b_{safe} this means:

$$\tilde{a}_i > -b_{safe} \tag{11}$$

The second criterion determines the acceleration advantage that would be gained from the event. This criterion based on the accelerations of the longitudinal model before and after the lane change and focused on improving the traffic situation of an individual driver by letting him drive faster or avoid a slow leader (24). For symmetric overtaking rules, they neglect differences between the lanes and propose the following incentive condition for a lane changing decision of the driver of vehicle i as follows:

$$\overbrace{\tilde{a}_i - a_i} + p(\overbrace{\tilde{a}_n - a_n} + \overbrace{\tilde{a}_o - a_o}) > \triangle a_{th} \tag{12}$$

Equation (11) states that the acceleration advantage to be gained by the lane change, must be greater than both a threshold acceleration $\triangle a_{th}$ used to dampen out changes with marginal advantage, and a politeness factor p determines to which degree these vehicles influence the lane-changing decision. The factor p controls the degree of cooperation while considering a lane change, from a purely egoistic behavior (p = 0) to an altruistic one (p ¿ 1) (25). The politeness factor can be thought of as accounting for driver aggressiveness. It is this balancing of accelerations that gives rise to the name MOBIL, as Minimizing Overall Braking Induced by Lane changes (27).

3 Proposed approach

In comparison with the existing works above, our proposal in this paper provides a extended car following model with an interaction of lane change behavior that mainly important to simulating and to representing the traffic flow in the real manner. The proposed approach is detailed in the following section.

3.1 Flowchart of the proposed approach

For an ideal flow of a dynamic traffic simulation study, we proposed the basic algorithm presented in Fig. 4 which based on three major steps given as:

- Preparation of the traffic flow simulation: in this step, we must define the road environment and also we must specify the initial parameters and variables, including initialization of position, velocity, and so on.
- Implementation of the model and validation of its different scenarios: in this stage, we adopt our MVSD model to compute acceleration for each car and then compute the new speed and position on both lanes for the next time step. At the same time, we start lane changes rules, we determine which car change whereto and add these cars to the correct position on the lane and removed changed cars from their old lane.
- Analysis of results: for the next time step, we update the network and information state to get a new velocity and position state; then we jump to step 2, and we begin an another cycle.

3.2 Modified velocity separation difference model

In this paper, we proposed a modified car following model introducing the lane changing rules just as other studies. In ref (21), the authors modified an OV model, introducing the new OV function without using the lane change behavior to get a model more reactive on braking situation called modified optimal velocity model (MOVM). The motivation for our paper comes from the key idea behind the new optimal velocity function proposed by (21) which we incorporating this latter on the VSDM model using the lane changing behavior. However, the new OV function combined between the OV function the reference Eq (2) and the weighting factor Eq (8) that depends on the inverse of time to collision (TTC). The TTC concept was introduced by the US researcher (28) and it was used in different studies as a time based surrogate safety measure for evaluating collision risk (29)(30)(31). In car following situations the TTC indicator is only defined when the speed of the following vehicle is higher than the speed of the lead vehicle (31). Rear end collision risk is defined as the time for the collision of two vehicles if they continue at their present speed and on the same lane and at the same speed (see Fig. 5). The time to collision of a vehicle driver combination n at instant t with respect to a leading vehicle$n1$ can be calculated with:

$$TTC = \frac{S(t)}{\dot{S(t)}}; \forall \dot{S(t)} > 0 \tag{13}$$

The new optimal velocity function $V_{opt}^{new}(S, \dot{S})$ is expressed as the combination of the optimal velocity function proposed by (18) based only on headway stimulus and the weighting factor established the inverse of time to collision to make the model more reactive in braking case.

$$V_{opt}^{new}(S, \dot{S}) = V_{opt}(S) * W(S, \dot{S}) \tag{14}$$

Where the weighting factor is :

$$W(S, \dot{S}) = [A(1 + tanhB(\frac{\dot{S}}{S} + C)] \tag{15}$$

The weighting factor must satisfies some proprieties:

- When the relative speed is positive $\dot{S(t)} > 0$, the weighting must maintain the reference OV function unchanged.
- For negative decreasing relative speed $\dot{S(t)} < 0$, it has to be decreasing and has to go toward zero when $\dot{S(t)} -> infini$.

There are several functions which behave similarly with varying only the headway stimulus. Therefore, Here the new OV function modulates the reactivity of the car following model according to the actual headway and relative speed between the follower and ahead car. In our contribution, we revised and extended a velocity separation difference model by incorporating the new OV function to get a new model that called a Modified Velocity Separation Difference (MVSDM). The MVSD model is expressed by the equation of motion as:

$$\begin{aligned} a_i(t) &= \kappa(V_{opt}^{new}(S(t), \dot{S(t)}) - v_i(t)) \\ &+ \lambda H(\dot{S(t)}) \dot{S(t)} (1 + \tanh(C_1(S(t) - l) - C_2)^3 \\ &+ \lambda \Theta(-\dot{S(t)}(t)) \dot{S(t)} (1 - \tanh(C_1(S(t) - l) - C_2)^3 \end{aligned} \tag{16}$$

To describe real driving behavior on multilane roads, we need the car following process and the lane changing process. The lane changing behavior has a significant effect on traffic flow. Therefore the understanding of lane changing behavior is important in several application fields such as capacity analysis and safety studies. We interested, particularly, the lane changing algorithm MOBIL (Minimizing Overall Braking Induced by Lane Changes) which is among the most important components of a microscopic traffic simulator based on a microscopic longitudinal movement model (25) and is adopted here.

4 Simulation results

In this study, we carry out the simulations to investigate whether MVSDM can overcome the shortcomings of previous models and compared MVSDM with MOVM proposed by (21). In this paper, for each model we establish the

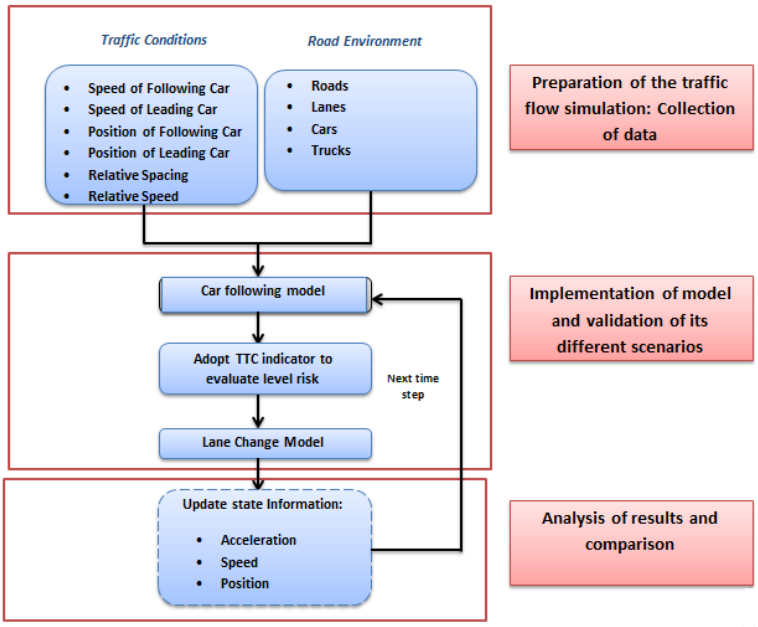


Figure 4: Flowchart of the proposed approach

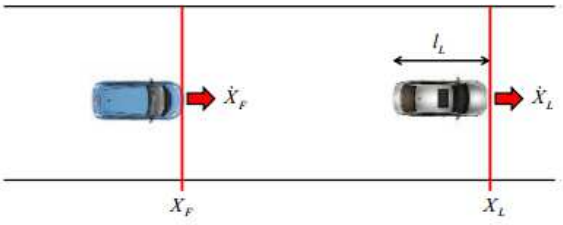


Figure 5: Time to collision for rear end collision sketch

simulation results for two different scenarios. In the following, we will test the proposed approach (accelerating and braking behavior) using an open source microscopic simulator proposed by (4) to validate our approach using these scenarios. We used two vehicle classes: cars and trucks. For all simulations, the parameter values used for optimal velocity function Eq (4) and are adapted from (19) are $V_1 = 6.75m/s$, $V_2 = 7.91m/s$, $C_1 = 0.13m^{-1}$, and $C_2 = 1.57m^{-1}$. The parameter values calibrated for weighting factor (21) are $A = 0.5$, $C = 0.5$, and $B = 5s$. The sensitivities parameters values are $a = 0.6m/s^2$, and $\lambda = 0.45m/s^2$. The parameters values for cars are the desired velocity $V_0 = 120km/h$, the safe time headway $T = 1.2s$, the minimum gap $S_0 = 2m$, and the vehicle length $l = 6m$. The parameters values for trucks are the desired velocity $V_0 = 80km/h$, the safe time headway $T = 1.7s$, the minimum gap $S_0 = 2m$, and the vehicle length $l = 10m$. The parameters values for lane changing are the politeness factor $p = 0$, the changing threshold $\triangle a_{th} = 0.2m/s^2$, the maximum safe deceleration $b_{save} = 12m/s^2$, and the bias for the slow lane $\triangle a_{bias}$.

For more information about the simulation results, we built a video to visualize clearly the validity of our proposed model MVSDM and the existing model MOVM and VSDM in the following link `https://www.youtube.com/watch?v=LJ5ddRGVbgA&feature=youtu.be`.

When starting the simulation, we extract the necessary data in excel format in order to represent them in graph form, and this is done for each car following model and for each scenario. Figure 6 shows the resulting data (speed, acceleration, position, type of car, length, etc.)

4.1 Ramp scenario: behavior in stop and go traffic

Stop and go scenario demonstrates the traffic breakdown provoking on the main road of the on-ramp. Usually, the traffic jam occur when the leading car decelerate for certain reasons. For that, it's important to study the vehicle behavior when simulating in such case. Simulation results depicted in Fig. 7d show that the proposed model avoids the collision when the leading car decelerate hardly. However, simulating traffic flow with MOV model occurs crashes between different cars as we can see in Fig. 7b.

At $t = 0$, all cars start up according to the MOVM, VSDM, and MVSDM, respectively. From Fig. 8, it can be seen that the speed maximum of MVSDM is under of MOVM and VSDM. We can see that MFVDM velocity begins to decrease before MOVM and VSDM velocity reaches its maximum. The simulation results demonstrate that MOVM and VSDM provokes crashes. In contrast, our proposal MVSDM avoid it and the traffic jams disappear.

To simulate the car motion and to describe the traffic flow, we examine certain properties of traffic from each car.

	A	B	C	D	E	F	G
1	Position	Speed	Accelerati	N° of vehicle	Time	Lane	Lenght
2	7,913354	14,57122	0,022237	2	0,2	0	6
3	10,8276	14,57527	0,020261	2	0,4	0	6
4	13,74265	14,57897	0,018487	2	0,6	0	6
5	16,65845	14,58235	0,016894	2	0,8	0	6
6	19,57491	14,58544	0,015465	2	1	0	6
7	22,492	14,58828	0,014183	2	1,2	0	6
8	0	14,56188	9,28E-10	1360544799	1,4	0	6
9	25,40966	14,59088	0,013035	2	1,4	0	6
10	2,912377	14,56188	8,35E-10	1360544799	1,6	0	6
11	28,32783	14,59328	0,012006	2	1,6	0	6
12	5,824753	14,56188	7,52E-10	1360544799	1,8	0	6
13	31,24649	14,5955	0,011085	2	1,8	0	6
14	8,73713	14,56188	6,77E-10	1360544799	2	0	6
15	34,16559	14,59755	0,010262	2	2	0	6
16	11,64951	14,56188	6,09E-10	1360544799	2,2	0	6
17	37,0851	14,59946	0,009526	2	2,2	0	6
18	14,56188	14,56188	5,48E-10	1360544799	2,4	0	6
19	40,00499	14,60123	0,008869	2	2,4	0	6
20	0	14,56188	7,45E-10	248792245	2,6	1	10
21	17,47426	14,56188	4,93E-10	1360544799	2,6	0	6
22	42,92524	14,60289	0,008284	2	2,6	0	6
23	2,912377	14,56188	6,85E-10	248792245	2,8	1	10
24	20,38664	14,56188	4,44E-10	1360544799	2,8	0	6
25	45,84582	14,60444	0,007763	2	2,8	0	6

Figure 6: Example of resulting data according to MVSDM

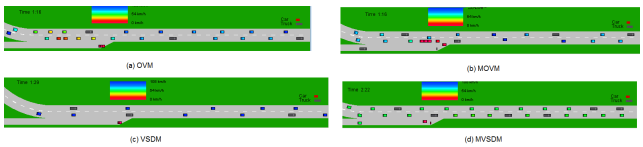


Figure 7: Simulation of ramp according to (a) OVM and (b) MOVM(c) VSDM and (d) MVSDM

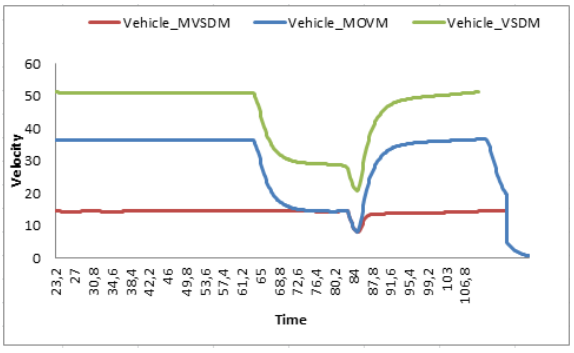


Figure 8: Time evolution of velocity variation according to MOVM, VSDM, and MVSDM

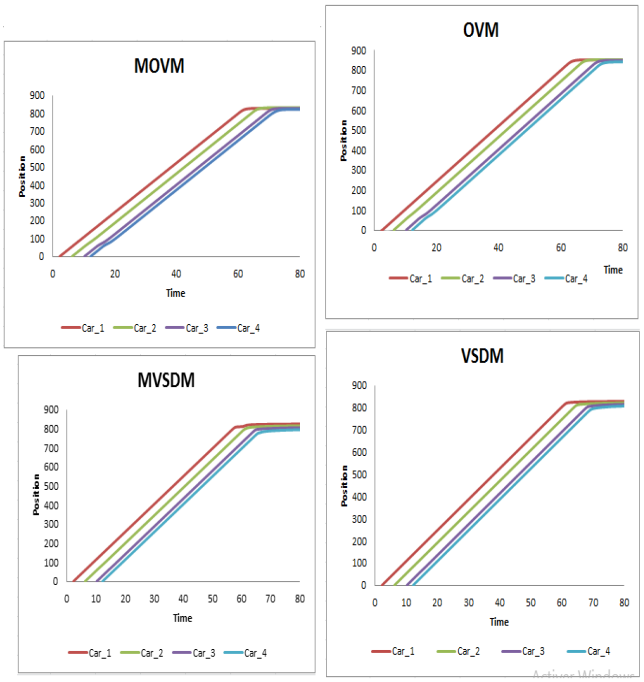


Figure 9: Position variation according to MOVM, VSDM, and MVSDM

Figure 9 gives the position evolution of four simulated cars, it's seen that the previous models provokes the collision. In contrast, our proposed approach avoids it.

4.2 Traffic lights scenario: behavior at stopping and approaching traffic signal

The traffic lights scenario describes the driving behavior of the vehicle when approaching a traffic signal. First a traffic signal is red and a queue of vehicles is waiting which the optimal velocity is 0. When the signal turn to green, at $t = 0$, vehicles start. For that, the traffic lights signal is represented by virtual obstacles in each lane which is removed when the light turns to green. Figure 10 represents the velocity variation of two vehicles using the MVSDM in the case of several changes. At the beginning, vehicle 1 follows vehicle 2 in the same lane 0, after a few moments vehicle 1 change the lane 0 towards the lane 1 that is why two vehicles show themselves in parallel when approaching traffic lights at $t = 57$. In approaching phase, and at $t = 72$ vehicles should decelerate smoothly which clearly shown that the vehicles stopped completely at a red light, and their velocity goes to 0. When the signal changes to green, vehicles begin to accelerate.

Figure 11 shows the behavior of vehicle according MOVM, MVSDM, and VSDM. Through these results, we deducted that the velocity of vehicle applying MOVM doesn't go to 0 that means all vehicles don't stop at a red light. However, when we simulate applying VSDM and MVSDM, all vehicles behave correctly by stopping at a

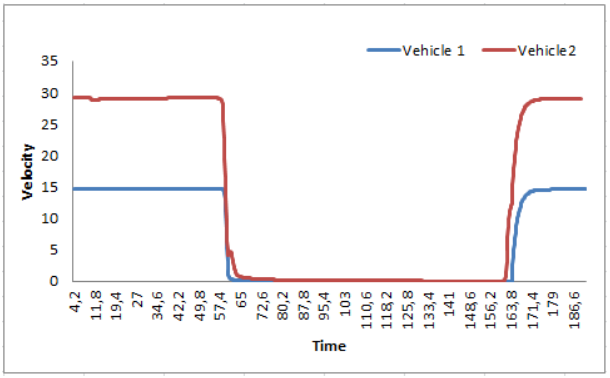


Figure 10: Driving behavior of two vehicles according MVSDM in each lane

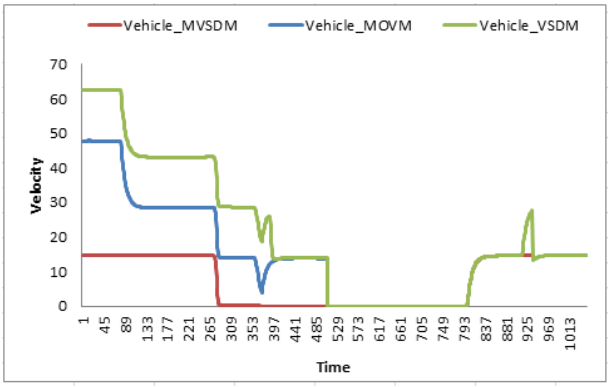


Figure 11: Simulation results according to MOVM, VSDM and MVSDM when approaching traffic lights signal

red light and moves when its turn to green. It's show clearly that the MVSD model react the realistic manner than MOVM and VSDM in braking case.

Figure 12 represents the snapshot of vehicle motion and their behavior according to MVSD, VSD, OV, and MOV models. Through these results, and when approaching traffic lights, it can be observed that the vehicles collide in the previous models. However, the problems of collision in emergency case were solved. Furthermore, the simulation results show that our proposed approach can exactly describe the driver's behavior when approaching traffic signal, where no crash occurs.

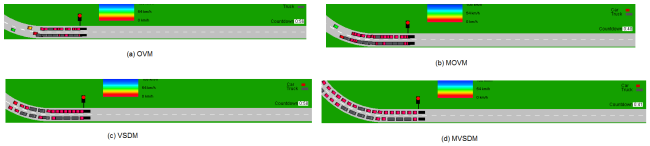


Figure 12: Simulation at traffic signal results according to OVM, MOVM, VSDM and MVSDM when approaching and stopping traffic lights signal

5 Conclusion

Through introducing the new optimal velocity function which takes into account not only the headway, but also the relative speed parameter into the VSDM, the modified velocity-separation difference model (MVSDM) is presented considering the driving behavior of the vehicle in braking case. In addition, to simulate in a realistic manner, we proposed to combine the proposed model with lane change model. The MVSDM can exactly describe the driver behavior under two proposed scenarios: when approaching traffic signal and an on ramp road, where no collision occurs. We can see that MVSDM is much close to the reality. However, the collision and crashes occur in the previous models. We proposed as a future work, to validate the model in bidirectional road scenario with multilane.

Literature

[1] Muhammad, J. F., (2015) Modeling and Analysis of Inter-vehicle Communication: A Stochastic Geometry Approach. Thesis. pp. 1-100.

[2] Zhu, W., Liu, Y., (2008) A Total Generalized Optimal Velocity Model and Its Numerical Tests. J. Shanghai Jiaotong Univ. (Sci.). 13(2), pp. 166170. DOI: 10.1007/s12204-008-0166-9.

[3] Sven, M., Bart, D. M., (2005) Transportation Planning and Traffic Flow Models. ArXiv preprint physics/0507127. pp. 1-51. DOI:https://arxiv.org/abs/physics/0507127.

[4] Treiber, M. and Kesting, A. (2010) An Open-Source Microscopic Traffic Simulator. IEEE Intelligent Transportation Systems Magazine, 2(3), pp.6-320. DOI:10.1109/mits.2010.939208.

[5] Aghabayk, K., Sarvi, M., Young, W. (2015) A State-of the-Art Review of CarFollowing Models with Particular Considerations of Heavy Vehicles. Transport Reviews. 35(1), pp. 82-105. DOI: 10.1080/01441647.2014.997323.

[6] Zhi-Peng, L., Yun-Cai, L. (2006) A velocity-difference-separation model for car-following theory. Chinese Physics. 15(7), pp. 1570-1576. DOI: 10.1088/1009-1963/15/7/032.

[7] Jabeena, M. (2013) Comparative Study of Traffic Flow Models And Data Retrieval Methods From Video Graphs. International Journal of Engineering Research and Applications. 3(6), pp. 1087-1093.

[8] Brackstone, M., McDonald, M. (1999) Car-following: a historical review. Transportation Research Part F: Traffic Psychology and Behaviour. 2(4), pp. 181-196. DOI: 10.1016/s1369-8478(00)00005-x.

[9] Darbha, S., Rajagopal, K., Tyagi, V. (2008) A review of mathematical models for the flow of traffic and some recent results. Nonlinear Analysis: Theory, Methods and Applications. 69(3), pp. 950-970. DOI: 10.1016/j.na.2008.02.123.

[10] Bellomo, N., Dogbe, C. (2011) On the Modeling of Traffic and Crowds: A Survey of Models, Speculations, and Perspectives. SIAM Review. 53(3), pp. 409-463. DOI: 10.1137/090746677.

[11] Hoogendoorn, S., Bovy, P. (2001) State-of-the-art of vehicular traffic flow modelling. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering. 215(4), pp. 283-303. DOI: 10.1177/095965180121500402.

[12] Wilson, R., Ward, J. (2011) Car-following models: fifty years of linear stability analysis-a mathematical perspective. Transportation Planning and Technology. 34(1), pp. 3-18. DOI: 10.1080/03081060.2011.530826.

[13] Papageorgiou, M. (1998) Some remarks on macroscopic traffic flow modelling. Transportation Research Part A: Policy and Practice. 32(5), pp. 323-329. DOI: 10.1016/s0965-8564(97)00048-7.

[14] Orosz, G., Wilson, R., Stepan, G. (2010) Traffic jams: dynamics and control. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 368(1928), pp. 4455-4479. DOI: 10.1098/rsta.2010.0205.

[15] Lazar, H., Rhoulami, K., Rahmani, M. D. (2016) A Review Analysis of Optimal Velocity Models. Periodica Polytechnica Transportation Engineering, 44(2), pp.123-131. DOI : 10.3311/pptr.8753.

[16] Helbing, D. (2001) Traffic and related self-driven many-particle systems. Reviews of Modern Physics, 73, pp.10671141. DOI:https://arxiv.org/abs/cond-mat/0012229.

[17] Newell, G. F. (1961) Nonlinear effects in the dynamics of car-following. Operations Research. 9(2), pp. 209-229. DOI: 10.1287/opre.9.2.209.

[18] Bando, M., Hasebe, K., Nakayama, A., Shibata, A., Sugiyama, Y. (1995) Dynamical model of traffic congestion and numerical simulation. Physical Review E. 51(2), pp. 1035-1042. DOI: 10.1103/physreve.51.1035.

[19] Helbing, D., Tilch, B. (1998) Generalized force model of traffic dynamics. Physical Review E. 58, pp. 133-138. DOI: 10.1103/physreve.58.133.

[20] Jiang, R., Wu, Q., Zhu, Z. (2001) Full velocity difference model for a carfollowing theory. Physical Review E. 64(1). DOI: 10.1103/physreve.64.017101.

[21] Mammar, S., Mammar, S., Haj-Salem, H. (2005) A Modified Optimal Velocity Model for vehicle following. IFAC Proceedings Volumes, 38(1), pp.120-125.DOI:10.3182/20050703-6-cz-1902.02043.

[22] Gipps, P. G. (1986) A model for the structure of lane-changing decisions. Transportation Research B. 20(5),pp.403-414.DOI:https://doi.org/10.1016/0191-2615(86)90012-3.

[23] Halati, A., Lieu, H., Walker, S. (1997) CORSIMCorridor traffic simulation model. in Proc Traffic Congest Traffic Safety 21st Century Conf., pp. 570576

[24] Umer, K., Pavlos, B., Lars, Sc., Alexandros, N., Dimitrios, K. (2014) Analyzing Cooperative Lane Change Models for Connected Vehicles. International Conference on Connected Vehicles and Expo (ICCVE). DOI:10.1109/ICCVE.2014.136

[25] Kesting, A., Treiber, M., Helbing, D. (2007) General Lane Changing Model MOBIL for Car-Following Models. Transportation Research Record: Journal of the Transportation Research Board, 1999, pp.86-94.DOI:http://dx.doi.org/10.3141/1999-10.

[26] Schakel, W.,Knoop, V.,VanArem, B. (2012) Integrated lane change model with relaxation and synchronization. Transportation Research Record, pp. 4757. DOI: https://doi.org/10.3141/2316-06.

[27] Caprani, C. C., Enright, B., Carey, C.(2012) Lane changing control to reduce traffic load effect on long-span bridges. F. Biondini, D.M. Frangopol, Eds, 6th International Conference on Bridge Maintenance, Safety and Management, Stresa, Italy. Taylor and Francis.

[28] Hayward, J.C. (1972) Near miss determination through use of a scale of danger (traffic records 384). Highway Research Board, Washington,DC.

[29] Behbahani, H., Nadimi, N., Alenoori, H., Sayadi, M. (2014) Developing a New Surrogate Safety Indicator Based on Motion Equations. PROMET - Traffic and Transportation, 26(5).DOI:10.7307/ptt.v26i5.1388.

[30] Minderhoud, M., Bovy, P. (2001) Extended time-to-collision measures for road traffic safety assessment. Accident Analysis and Prevention, 33(1), pp.89-97.DOI:10.1016/s0001-4575(00)00019-1.

[31] Vogel, K. (2003) A comparison of headway and time to collision as safety indicators. Accident Analysis and Prevention, 35(3), pp.427-433.DOI:10.1016/s0001-4575(02)00022-2.

Prediction of Sentiment from Macaronic Reviews

Sukhnandan Kaur and Rajni Mohana
Department of CSE, JUIT, Waknaghat, 173234, India
E-mail: sukhnandan.kaur@mail.juit.ac.in, rajni.mohana@juit.ac.in

Technical paper

Keywords: macaronic language, sentiment analysis, supervised learning, normalization

Received: March 11, 2017

Web-sphere is the vast ocean of data. It allows its users to write their opinion, suggestions over various social platforms. The users often prefer to write in their native language or some hybrid content (i.e., combination of two or more languages). It's also observed that people use a word or two of their native language in a text of base language. The presence of native words along with base language is known as macaronic languages. For example: Dunglish (Dutch and English), Chinglish (Chinese and English), Hinglish (Hindi and English) The use of macaronic languages over the web is on the rise these days. This type of text generally doesn't follow any syntactic structure, thus making processing of the content difficult. This paper deals with extracting meaningful information of a text containing macaronic content. It also facilitates the need of expert analysers for the processing of such content to take effective decisions. The performance of various decision support systems is dependable over these analysers. Therefore, this paper presents an algorithm which initially normalizes the content to its base language; later performs sentiment analysis over it. The experimental results using proposed algorithm indicates a trade-off between various performance aspects.

Povzetek: Prispevek predstavi iskanje razumevanja makaronskega besedila, tj. besedila z dodanimi besedami drugega jezika.

1 Introduction

Online review communities successfully allow its users to write their opinion, suggestions over various social platforms. These reviews greatly affect the decision to buy or sell any product and to use any service. It is fruitful to the manufacturer or service provider to enhance the productivity. Automatic decision support systems take these reviews into account for sentiment analysis. However, it is extremely difficult to have reviews in uniform language. During an automatic processing of reviews written online, it is found that 2/3 of the internet users are non-English [5] . The reason behind this is that most of the people have the ability to learn only 2 or 3 languages proficiently. In this technological world, people have equal priority to write over the internet among different languages. People who write reviews belong to different communities from different regions of the world; they have the freedom to use their native language too. When a text contains more than one language, it is called as multilingual text. If a single sentence contains more than one language, then it is called as macaronic text[18].

Example 1: Samsung अच्छा cellphone ,

In the above mentioned text, it is taken as macaronic content containing Hindi and English languages.

These irregularities found in the data over the internet make the processing more complex. Due to the scarcity of the language resources over the web, it becomes very difficult to handle all the possible languages over the globe. It is a challenging task of a natural language processing group. The formalism in sentiment analysis limits the system to specific users. The reviews from all the users of a particular entity are valuable. It increases the need of automated systems to handle multilingual content. Derkacz et.al.[12] stated some of the requirements to have a multilingual automated system. These requirements are further taken care by language processors to build a multilingual system. In case multilingual systems, the language of whole document is taken into account whereas for macaronic language processing, we need to detect the language of each word. This paper proposes a sentiment analyser which deals with the macaronic text. Initially, reviews are to be normalized during pre-processing stage. Later, these reviews are processed through sentiment analyser.

This paper is organised as: section 2 describes the state of the art sentiment analysers. In section 3, system design and algorithm is proposed. Experimental analysis using various performance metrics are presented in section 4. Finally, the whole work is concluded in section 5.

2 Related work

Numerous researchers have worked in the field of natural language processing. Kaur et.al.[14] presented sentiment analysis of reviews written in Punjabi language. The researchers collected the reviews written in Punjabi which afterwards segregated into positive or negative reviews. Das et.al.[8] found the need of having SentiWordNet for Bengali language. Their work helped the researchers in the field of sentiment analysis. The researchers annotated the required lexicon. Das et.al.[7] worked for sentiment analysis of reviews written in Bengali language. In this paper, the researchers have used support vector machine (SVM) with Bengali SentiWordNet. The paper presents the feature extraction for Bengali language. Das et.al.[6] developed subjectivity clues based on theme detection techniques. Bengali corpus is used in their work and later compared the results with English subjectivity detection. Das et.al.[9] developed a gaming theory by which researchers can easily build the SentiWordNet in the required language. This work demands the respective linguistic experts. Joshi et.al.[13] used supervised learning approach for their work by using Hindi- SentiWordNet for their work. In this paper, researchers used standard translation techniques to preserve the polarity of each document while translating it. Bakliwal et.al.[2] worked for detecting subjectivity based on graph theory. Researchers explored the effect of synonym and antonym over the subjective nature of the document. The results were good for Hindi and English. The researchers claimed that their strategy will work well in other languages too. Das et.al.[10] developed a system for deducing the emotion and intensity of emotion based on sentiment hidden in the data. In this work, researchers have used supervised learning methods for their work. Richa et.al.[21] presented a survey for sentiment analysis in Hindi language. The results have shown that sentiment analysis in Hindi language is complex as compared to English language. The reason behind this complexity is the non-uniform nature of the Hindi language. Various research challenges are also discussed. Researchers[21] developed a system which depicts the polarity of the text and tested their system over the Hindi movie reviews. Parul et.al.[1] developed a sentiment analyser for movie reviews written in Punjabi language using various machine learning algorithms. Raksha et.al.[20] used semi-supervised technique for polarity detection in Hindi movie reviews. In their work, researchers reported 87% accuracy of the proposed system by using bootstrapping and graph based approach for sentiment analysis. Pooja et.al.[17] used Hindi SentiWordNet for finding opinion orientation of the reviews. Researchers used unsupervised learning for their work. Kerstin et.al.[11] developed a system for multilingual text for obtaining the polarity of reviews written in language other than resource rich language English. Researchers used a standard translation methodology and supervised learning for sentiment analysis. C. Banea et.al.[3] developed a system which focused on the sentiment analysis based on translation of input document other than English. In their work, researchers used English as a source language. They used supervised learning approach for their work. For the translation of the text correctly various available translators are used. i.e. Goggle, Moses, Bing translators.

The work by different researchers is summarized into table 1 . It is noticed that researchers are focusing well in the area of multilingual sentiment analysis. Researchers focused in finding document language for translating any document into base language instead of language of individual word. This sometimes discard the opinion bearing word written in any foreign language. As in example 1, the word अच्छा, means good is discarded if the document language is detected as English. The efficient processing of such documents is required to increase the effectiveness of the decision support system.

2.1 Motivation

After looking into the scenario, we found that we need SentiWordNet in almost every language all over the global. It is very complex task. The motivation behind the proposed system is that the existing system for multilingual sentiment analysis is unable to process macaronic data. The rise in the volume of macaronic data over the internet arise the need of proposed system. The reasons for having macaronic content over the web in huge volume are as follows:

1. Scarcity of Resources: Sentiment analysis task demands for the availability of lexicons or data of any particular language. There is huge variation in every language model. This makes the model used for one language cannot be used for other languages.
   For example: Chinese language model does not consider spaces while as other models focus mainly over spaces to tokenize.

2. Lack of uniformity of languages: Most of the languages often follow their own traditional structures. Thus, processing of each language data with the general structure model gives unsatisfactory results.
   For example: English language use Subject-Verb-Object(SVO) while Hindi Language model follow Subject-Object-Verb (SOV)

3. Freedom of writing in native language: People these days have number of followers from different countries through various online applications. They are also able to propagate their ideas through it. Sometimes, few words they prefer writing in their own native language, which may not be understandable by some of the followers. In case of an automated system, during pre-processing through one language model, these native words may be neglected taken as foreign language words. Sometimes, we may lose meaningful information during this type of pre-processing.
   For example: सैमसंग is on great demand.
   सैमसंग(Samsung) is negected by English language

Author	Work	Level	Language	Results	Technique	Corpus	Year
Danet et.al.[5]	Classification of reviews into positive or negative opinion	Document level	Punjabi	Accuracy = 75%	Machine Learning	Blogs	2014
Derkacz et.al.[18]	Classification of reviews into positive, negative, neutral or emotion (sad, happy, etc)	Document level	Bengali	Precision = 70.04%, Recall = 63.02%	Machine Learning	Custom Lexicon	2010
Das et. al.[14]	Document are separated based on Domain independent subjectivity and factual content	Sentence Level	Bengali	Precision = 70.04%, Recall = 63.02%	Machine Learning	Custom Lexicon	2009
Bandyopadhyay et. al.[6]	Sentiment analysis of Hindi reviews, English reviews using Hindi SentiWordNet	Document Level	Hindi, English	Precision = 70.04%, Recall = 63.02%	Supervised	Movie reviews	2012
Joshi et. al.[9]	Subjectivity clues based on antonym and synonym using graph theory	Document Level	Hindi, English	Accuracy = 79%	Supervised	Movie reviews	2012
Sharma et. al.[10]	Polarity detection of movie reviews using unsupervised techniques	Sentence Level	Punjabi	NA	Unsupervised	Movie reviews	2015
Arora et. al.[21]	Sentiment orientation of reviews written in Hindi language	Document Level	Hindi	Precision = 70.04%, Recall = 63.02%	Unsupervised	Movie reviews	2014
Sharma et. al.[1]	Sentiment analysis using Semi-Supervised techniques	Document Level	Hindi	Accuracy = 87%	Semi-Supervised	Movie reviews	2014
Pandey et. al.[20]	Opinion orientation of Hindi movie reviews is deduced using Hindi-WordNet	Document Level	Hindi	NA	Unsupervised	Movie reviews	2015
Denecke et. al.[17]	Polarity detection from reviews using standard translation of German reviews in English afterwards find the polarity	Document Level	German, English	Accuracy = 66%	Supervised	Movie review	2008
Banea et. al.[11]	Enabling Multilingual question answering system	Document Level	French, German and Spanish	NA	Supervised	Question Answers	2016

Table 1: State of Art Multilingual Sentiment Analysis

model. Thus, it becomes difficult to extract samsung as an entity.

4. For getting point of attraction: People use the multilingual content or some fancy words in various applications like product advertisements, shop names, etc. This makes the task of processing such web content complex. For example:
सamसung (Samsung) is on great demand. ■
सamsung (Samsung) is on great demand.
म ona(Mona) is feeling so good.
Hence, from the above examples, Samsung is hard to detect as it is being neglected by chosen language model.

Due to the above mentioned reasons, it is very much necessary to have an efficient system to process macaronic language content. Our contribution is to enhance the performance i.e.precision, recall and accuracy using supervised sentiment analysers. The proposed system is with less fallout which shows its high efficiency.

3 System design

The proposed system as shown in Figure 1 applies a variant of techniques for normalization of macaronic text and classification of reviews. The system consists three major components:

1. Language Processing
2. Text Processing and
3. Sentiment Analysis

A component based on language detection is carried out using algorithm 1. The core idea of this component is to normalize the macaronic content. Other two components are carried out using algorithm 2. It normalizes the content to extract the SentiStrength of each document. Combination of these two algorithms (Algorithm 1 and Algorithm 2) is used to carry out sentiment analysis for multilingual or macaronic language documents.

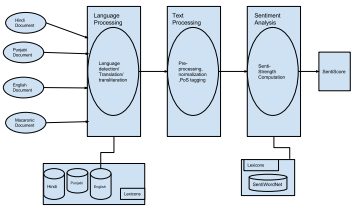


Figure 1: Proposed System Design

1. Language Processing: It is the primary component of the proposed system. In this component tokenization, language detection and conversion of tokens to its base language is carried out. These sub-components are described as follows:

   (a) Tokenization: It is the basic unit of any language processing task. A sequence of sentences, words or characters are passed as an input to any system.
   The output of this phase is tokens. It can be done at different levels depending upon the level of granularity: sentence level, word level, character level as shown in table 2. The proposed system is based on word level tokenization for macaronic language.
   E.g. Samsung has a good market value. Users are happy with its mobile products.

   (b) Language Detection/ Translation: For language detection, we have used PoS[19] tagging, as shown in table 3. The unrecognized or untagged tokens can be passed through language detection module. The output of this phase is the tokens in the base language of the system. i.e. Taking English as a base language. If the token is found in Hindi WordNet then Hindi to English translator is applied to it. On the other end, if the word belongs to Punjabi language, it is passed through the Punjabi to English translator. It is a general procedure which can be applied to various other languages too.

2. Text Processing: It is the second important component of the proposed system. It carries various sub-tasks described as follows:

   (a) Normalization: After filtration of subjective sentences, normalization is to be done. The process of normalization is to regularize or process the grammatical variants present in the sentence. Grammatical variants include past verbs (regular and irregular) / present verbs, classification of noun phrases in singular and plural. In normalization, finding the abbreviations, case folding, etc.Normalization is a process having data in a well format as required for appropriate processing. It includes:

Level of Processing	Number of Tokens
Sentence Level	2
Word Level	13
Character Level	74

Table 2: Tokenization at different levels

i. Handling Slangs: Slangs are playing indispensable role in opinion mining. So, it will be worthless to reject all the slangs by counting them as stop words. Various algorithms are applied to handle different types of slangs.Types of slangs[5]:
   - Emoticons: Bad☹, happy☺
   - Interjections:Mmmmm-pleasure, hmmmm-wondering, Mhmmm-confirmation
   - Intensionally misspelled: cooooooool, goooooooood, nyt, etc
   - Alphanumeric strings: gr8, 9t, etc.

   Test sentence:
   She is flying high by having this cellphone.☺
   She is flying high by having this cellphone. Happy

ii. Idiomization / Replacement of idioms with their actual meaning: In English literature, idioms play very important role in fixing the opinion from sentence about the particular entity. If the stops words are removed then some words which may or may not the part of the idiom can be rejected. In reality, these words are highly contributed to the opinion.
   Test sentence:
   She is flying high by having this cellphone. Happy
   She is very happy by having this cellphone. Happy

(b) Tokenization: In our work, we have used word level tokenizer as mentioned in table 2 . The reason behind this to process each token according to its own language instead according of language of the document.

(c) PoS Tagging: Part of speech tagging plays a vital role in natural Language processing tasks. Initially, we have tried to focus whether the state of art PoS taggers are able to recognise a foreign word. For this purpose, we have used NLTK tagger[15] and Stanford Tagger[16]. We have shown the results of both the taggers for various test sentences in table3. We have found various untagged tokens which are then processed through language processing phase.

3. Sentiment Analysis: In this module, the potency of each review is calculated. The magnitude of the sentiment associated with each document is calculated by aggregating all the review's sentiscore corresponding to that document. SentiWordNet is the base for getting the actual magnitude of the sentiment of a document. For our work, we have used SentiWordNet v3.0.0. Sentiscore corresponds to each document is taken as an output as shown in table4 .

■

4 Evaluation

4.1 Dataset

We have extracted a corpus of reviews of 10 movies containing 200 movie reviews i.e.100 positive and 100 negative; 160 reviews were used for training and 40 for testing. Each review has a size ranges from 500 to 1000 words. Initially, classification of the corpus is elaborated according to user′s scoring: reviews are marked between 3 and 5 star rating are classified as positive whereas reviews marked between 0 and 2 are taken as negative. This prior classification is based on the assumption that the star rating is correlated to the sentiment of the review. For experiment evaluation, the data was pre-processed with the TreeTagger5, POS tagger and lemmatization tool. We have used Support Vector Machine (SVM), Nave Bayes, kNN and convolutional network as classification models to train the system and classify movie reviews. The reviews are not monolingual. These reviews are macaronic in nature i.e. it consists of more than one language i.e. Hindi and English in a single review. We manually annotate the reviews based on language of each token. The guidelines for annotation are stipulated the need of retaining the semantic structure of tokens. Five different graduate students participated in the reviewing process to formulate Gold Standard. To evaluate the inter-personnel disagreement, we have used kappa measure[4] and score 0.61 is obtained.

4.2 Performance

Formally, the performance of proposed sentiment analyser, PSA is a function of four factors as follows:
PSA(l,L_d,t,E_s)
Where L_d is Language Detection
l is a Learning Algorithm
t is a Tagger
E_s is a Experimental Setup

The performance of the analyser is directly affected by the choice of optimal parameters for each factors mentioned above. In the case of optimal parameters choice for each of the factor, sentiment analyser gives maximum performance (PSAmax).

On the other end, training consists machine translated data and testing of the learning algorithm is based on the human annotated dataset i.e. Gold Standard. The performance of sentiment analyser (PSA) is negatively affected by error in language detection phase (E_{Ld}) as given in equation 1 .

$$PSA = PSA_{max} - E_{Ld} \quad (1)$$

In case of optimal parameters, $E_{Ld} \rightarrow 0$, PSA = PSA_max

Test Sentence	Pos tagging by NLTK tagger	Stanford tagger
मीड़िया गयान का एक - अच्छा सरोत हैं	मीड़िया—NN गयान—:का—:एक—:- अच्छा—सरोत—हैं—	मीड़िया/VBZ गयान/NNP का /NNP एक /NNP - अच्छा/NNP सरोत /NNP हैं /NNP
media is अच्छा source of knowledge	media—NNS is—VBZ - अच्छा—: source—NN of—IN knowledge—NN	media/NNS is/VBZ अच्छा/JJ source/NN of/IN knowledge/NN
मीड़िया गयान का एक good सरोत हैं	मीड़िया—NN गयान—:का—:एक—:good —JJ सरोत —हैं—	मीड़िया/VBZ गयान/NNP का /NNP एक /NNP good/JJ सरोत /NNP हैं /NNP
media गयान का एक - अच्छा सरोत हैं	media—NNS गयान—:का—:एक—:- अच्छा—सरोत—हैं—	media/NNS गयान/NNP का /NNP एक /NNP अच्छा/NNP सरोत /NNP हैं /NNP

Table 3: Tagging of various test sentences using NLTK and Stanford Tagger

Test Sentence	SentiStrength
texttt मीड़िया is good source of knowledge	0.47
media is good source of knowledge	0.47
मीड़िया गयान का एक - अच्छा सरोत हैं	0
media is अच्छा source of knowledge	0
मीड़िया गयान का एक good सरोत हैं	0.47
media गयान का एक - अच्छा सरोत हैं	0

Table 4: Sentiscore Associated With Review

Metric	Target	Target
Selected	tp	fn
Selected	fp	tn

Table 5: Confusion metric used to evaluate performance

4.3 Performance metric

For the analysis of results, the following performance metrics are used by various natural languages processing task including sentiment analysis. It includes precision, recall, F-measure and accuracy. These measures can be calculated using the confusion metric given in table 5.

Precision: It is defined as fraction of retrieved documents that are relevant. It is calculated using equation 2.

$$P = \frac{\text{number of correct positive or negative documents detected by the system}}{\text{no. of positive/negative documents detected by the system}} \quad (2)$$

Recall: It is defined as fraction of relevant documents that are retrieved. It is calculated using equation 3.

$$R = \frac{\text{number of positive or negative documents detected by the system}}{\text{no. of positive/negative documents present in the Gold Standard test set}} \quad (3)$$

F-measure: It is a harmonic mean with takes precision and recall both into account. It is a consecutive average of precision and recall. F-measure with $\alpha = 0.5$, means taking precision and recall at equal weightage.It is calculated using equation 4.

$$F = \frac{(\alpha^2 + 1) \times P \times R}{\alpha^2(P + R)} \quad (4)$$

Accuracy: it is the fraction of classifications that is correct. . It is calculated using equation 5.

$$A = \frac{t_p + t_n}{t_p + t_n + f_n + f_p} \quad (5)$$

Fall-out: It is a measure of the proportion of mistakenly selected non- targeted items. . It is calculated using equation 6

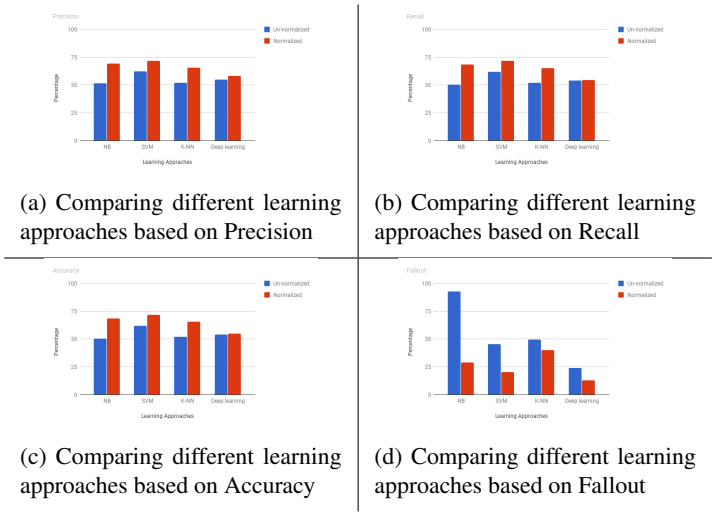
$$FO = \frac{f_p}{t_n + f_p} \quad (6)$$

4.4 Results and analysis

The outcomes of our experimental study are presented in Table 6 and Table 7. We can easily notice that every machine learning approach has its own pros and cons. Each of them is valuable in different aspects i.e. precision, recall, accuracy, fallout and execution time. To validate our results we have used 10-fold cross validation. For the experimental setup, we have used Support Vector Machines

Learning Approaches	Precision	Recall	Accuracy	Fallout	Time(sec)
NB	51.58	50.4	50.4	92.8	422
SVM	62.29	62	62	45.6	428
kNN	52.01	52	52	49.6	421
Convolutional network	54.96	54	54	24	751

Table 6: Un-normalized Macaronic Sentiment Analysis



(a) Comparing different learning approaches based on Precision

(b) Comparing different learning approaches based on Recall

(c) Comparing different learning approaches based on Accuracy

(d) Comparing different learning approaches based on Fallout

Figure 2: comparision of various methods

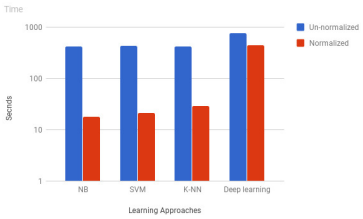


Figure 3: Comparison of execution time various machine learning Algorithms based on Proposed Scheme for normalized and unnormalized data

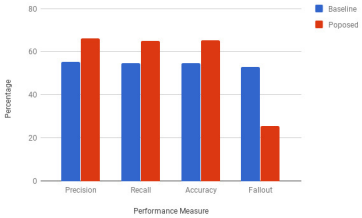


Figure 4: Comparison of proposed technique with State of art

(SVM), Nave Bayes (NB), kNN and Convolutional network(Deep Learning) to analyse the performance of proposed algorithm. The results are shown in Table 6 and Table 7. Precision, recall, accuracy, fallout are taken in percentage and time is taken in seconds. Time taken by each of the learning technique is very dependent on data size, data types, number of columns, computer hardware, memory, background running processes, cores, etc. This may vary with the change in any of the mentioned attribute. Hence, the time taken in table 6 and table 7 helped in deducing the time trend of each learning model. It is shown as an increasing order and noticed the reduction in the time to the marginal level in normalized content.

Order for unnormalized content:
$kNN < NaiveBayes < SVM < Convolutionalnetwork$
Order for normalized content:
$NaiveBayes < SVM < kNN < Convolutionalnetwork$

The results have shown in Figure 2 clearly evident the performance of proposed system using various learning approaches. These figures highlight the proposed system performance in various aspects. The proposed scheme outperforms the existing system using Nave Bayes by the rise in the values of precision, recall by 17.88% and 18.22%. Observing the results of other classifiers i.e. SVM, kNN and convolutional network also shows significant impro-

Learning Approaches	Precision	Recall	Accuracy	Fallout	Time(sec)
NB	69.46	68.62	68.63	28.79	18
SVM	71.72	71.69	71.75	20.21	21
kNN	65.41	65.31	65.47	40.21	29
Convolutional network	58.03	54.56	55.00	13.04	440

Table 7: Proposed normalized Macaronic Sentiment Analysis

Approach	Precision	Recall	Accuracy	Fallout
Baseline	55.21	54.6	54.6	53
Proposed	66.15	65.04	65.21	25.56

Table 8: Comparison with Existing Sentiment Analysis

vements in performance levels. Using SVM and kNN more than 9% and 13% improvement is noticed in precision and recall values using proposed approach. It is also noticeable that there is a trade-off between various performance aspects. The effectiveness of system is shown by convolutional network but it takes more time than other classifiers for macaronic sentiment analysis.

Through observing Figure 3, we have found that the proposed algorithm also greatly affect the time taken by each model. It is noticeable that the normalized content reduces the training time in every learning approach. By observing Table 8, results are compared to the baseline approaches; the average value of precision, recall is increased while the fallout is decreased significantly. Figure 4 shows that how effective the proposed approach is as compared to the state of the art sentiment analysis for macaronic language.

5 Conclusion

Over the web where huge user generated content has already existed; the need for sensible computation for decision support system is rising. The multilingual online content has led to the increase of web debris, which is inevitably and negatively affecting information retrieval and extraction for decision support systems. To analyse this negative trend and propose possible solution, this paper focused on the evolution of sentiment analysis based on bag-of-words for macaronic reviews. Different supervised machine learning approaches gave different cross validated results. This is done by borrowing the concept of training and testing from the field of machine learning. After successful evaluation, it is concluded that there is a trade-off between various performance measures. In this study, we have investigated the need to normalize the macaronic text. We have also performed sentiment analysis over the macaronic language text consists English and Hindi. We have found an average of about 11% rise in precision and recall values. It is also noticeable that training time is also reduced significantly using proposed approach. We further plan to develop a system to handle with more than two languages as a macaronic text for sentiment analysis. We also plan to apply our proposed algorithm for entity extraction.

References

[1] Arora, P. and B. Kaur (2015). "Sentiment Analysis of Political Reviews in Punjabi Language." International Journal of Computer Applications 126(14).

[2] Bakliwal, A., P. Arora, et al. (2012). Hindi subjective lexicon: A lexical resource for hindi polarity classification. Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC).

[3] Banea, C., R. Mihalcea, et al. (2008). Multilingual subjectivity analysis using machine translation. Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics.

[4] Bunt, H., V. Petukhova, et al. (2016). Dialogue Act Annotation with the ISO 24617-2 Standard. Multimodal Interaction with W3C Standards, Springer: 109-135.

[5] Danet, B. and S. C. Herring (2003). "Introduction: The multilingual internet." Journal of Computer Mediated Communication 9(1): 0-0.

[6] Das, A. and S. Bandyopadhyay (2009). Theme detection an exploration of opinion subjectivity. Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on, IEEE.

[7] Das, A. and S. Bandyopadhyay (2010). Opinion-Polarity Identification in Bengali. International Con-

ference on Computer Processing of Oriental Languages.

[8] Das, A. and S. Bandyopadhyay (2010). "SentiWordNet for Bangla." Knowledge Sharing Event-4: Task 2.

[9] Das, A. and S. Bandyopadhyay (2010). "SentiWordNet for Indian languages." Asian Federation for Natural Language Processing, China: 56-63.

[10] Das, D. and S. Bandyopadhyay (2010). Labeling emotion in Bengali blog corpusa fine grained tagging at sentence level. Proceedings of the 8th Workshop on Asian Language Resources.

[11] Denecke, K. (2008). Using sentiwordnet for multilingual sentiment analysis. Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on, IEEE.

[12] Derkacz, J., M. a. Leszczuk, et al. Definition of Requirements for Accessing Multilingual Information and Opinions. Multimedia and Network Information Systems, Springer: 273-282.

[13] Joshi, A., A. Balamurali, et al. (2010). "A fall-back strategy for sentiment analysis in hindi: a case study." Proceedings of the 8th ICON.

[14] Kaur, A. and V. Gupta (2014). "Proposed Algorithm of Sentiment Analysis for Punjabi Text." Journal of Emerging Technologies in Web Intelligence 6(2): 180-183.

[15] Kothapalli, M., E. Sharifahmadian, et al. "Data Mining of Social Media for Analysis of Product Review." International Journal of Computer Applications 156(12).

[16] Nguyen, D. Q., D. Q. Nguyen, et al. "A robust transformation-based learning approach using ripple down rules for part-of-speech tagging." AI Communications 29(3): 409-422.

[17] Pandey, P. and S. Govilkar (2015). "A Framework for Sentiment Analysis in Hindi using HSWN." International Journal of Computer Applications 119(19).

[18] Renduchintala, A., R. Knowles, et al. "Creating interactive macaronic interfaces for language learning." ACL 2016: 133.

[19] Seih, Y.-T., S. Beier, et al. "Development and Examination of the Linguistic Category Model in a Computerized Text Analysis Method." Journal of Language and Social Psychology: 0261927X16657855.

[20] Sharma, R. and P. Bhattacharyya "A Sentiment Analyzer for Hindi Using Hindi Senti Lexicon."

[21] Sharma, R., S. Nigam, et al. (2014). "Polarity detection movie reviews in hindi language." arXiv preprint arXiv:1409.3942.

Algorithm 1:

Input: $Document\ D\ where\ D = d_1, d_2, d_3, ..., d_k$
'k' is the total no. of documents
'm' is the total number of words in a document
$L_s\ =\ language\ of\ segment$
$L_b\ =\ Base\ language\ (English)$
Output:
$W_s(weightedSentiStrengthofeachdocuemnt)$
Begin
for $k = 1$ to k **do**
$Tokenization$
for $i = 1$ to m **do**
Encoding based on $UTF8$
end for
{Similar category segments are combined}
Segmentation based on encoding.
Language detection for each segment.
if $L_s = L_b$ **then**
$goto\ S1$
else
$Apply\ translation$
end if
$S1$ Assemble segments
Compute SentiStrength
end for

Algorithm 2:

Input: $Document\ D\ where\ D = d_1, d_2, d_3, ..., d_k$
'k' is the total no. of documents
'm' is the total number of words in a document

Output:
$W_s(weightedSentiStrengthofeachdocuemnt)$
{Token list(TL) = (t1,t2,.....,tn)}
{Word List(WL)= (w1,w2,w3,.....wx)}
{'q' is the total number of tokens in a document}
{P = list of 'positive category words}
{N = list of 'negative category words}
{Pw = weight assigned to a token belongs to positive category as per SentiWordnet}
{Nw = weight assigned to a token belongs to negative category as per SentiWordnet}
Begin
for $d = 1$ to k **do**
Tokenization
Stemming
Normalization
for $k = 1$ to m **do**
if $(t_k \in W) \bigcap (t_k \in P)$ **then**
$w_{pos}(k) = Pw(t_k)$
else if $(t_k \in W) \bigcap (t_k \in N)$ **then**
$w_{neg}(k) = Nw(t_k)$
else if $(t_k \in W) \bigcap (t_k \notin N) \bigcap (t_k \notin N)$ **then**
$w_{neu}(k) = 0$
end if
end for

$$W_s = \sum_{j=1}^{m} w_{pos}(j) \pm \sum_{j=1}^{m} w_{neg}(j) \qquad (7)$$

end for

Application of Distributed Web Crawlers in Information Management System

Bo Wen
School of Computer Science and Technology, Huaibei Normal University, Huaibei, 235000, China
E-mail: bowen1983@yeah.net

Technical paper

Keywords: web crawlers, Hadoop, information management system

Received: February 7, 2018

In the Internet era, cloud data and big data constantly develop, and Internet has become the main platform for enterprises and individuals to release information. As a result, a large amount of data generates, and people spend more energy on finding information that they want. The desire for accurately acquiring information needed becomes increasingly stronger. This study designed a distributed web crawlers system based on Hadoop and used it to do large-scale information management. The simulation experiment verified that the system could operate stably in information management system, which offers a reference for the application of distributed web crawlers in information management systems.

Povzetek: Razvit je distribuirani spletni preiskovalnik na osnovi Hadoopa za upravljanje informacij.

1 Introduction

Internet rapidly develops in the 21st century, accompanied by data volume increasing in exponential form on Internet. With the diversification of information, the management of information has become more and more difficult. How to timely and accurately search information through search engine and manage the information becomes crucial. Requirements on relevant technologies are also being improved constantly. With the development of computer, information management system has emerged. More efficient and simple information management systems are being developed. Qin [1] designed a SG-UAP development tool based on Eclipse development environment which was applicable to Windows operation system; a database platform was developed based on Oracle to provide Tomcat network information management service; the system managed network information through service-oriented architecture. Gupta et al. [2] analyzed management information service and proposed to manage network information with management information service and found that management information system could optimize network information and accurately collect and manage data. Zhao et al. [3] established a topic-focused crawler based scientific research information system to improve the information management level. Web crawlers can capture webpage information from the network and extracted and stored the key information to solve the urgent problem of information acquisition. But information collection based on web crawlers is facing with difficulties such as information repetition and existence of dynamic pages. Therefore distributed technologies are needed to solve the problems and enhance crawling efficiency. In the study of Su et al. [4], single-thread and multi-thread web crawlers were implanted into a distributed system to capture and store data with diversified and personalized operations, which enhanced the capturing speed. In the study of Zhang et al. [5], Hadoop based distributed web crawler system was optimized. The parameters were optimized through analysis on factors influencing crawling efficiency. Distributed web crawlers have great advantages in collecting and storing information; hence it can help establish a practical and high-efficient information management system. In this study, web crawlers were analyzed, and then a Hadoop based distributed web crawlers system was designed to manage network information. The simulation experiment suggested that the system could effectively collect and store network information and enhance the performance of single-node web crawlers, which provides a reference for the application of distributed network crawlers in information management system.

2 System related technologies

2.1 Web crawlers

Distributed web crawler is a program which crawls Web resources on the Internet according to some rules and provides the obtained network information to search engine. Therefore it is an indispensable part of search engine [6]. To achieve a high crawling ability, a web crawler should have the five characteristics [7].

(1) High performance

A large amount of information involves mass Uniform Resource Locator (URL). Distributed web

crawlers should timely and effectively capture useful information in webpage. The more the information in unit time is, the better the performance of web crawlers is.

(2) Expandability
Expandability should be improved to achieve a high performance of web crawlers. Expandability means that the whole crawler system will not be affect when the current web crawlers are being updated or doing other operations. Better expandability is needed in efficient crawling of information in different sites as the programming language and code editor are different in different websites.

(3) Robustness. Facing with a large number of servers, web crawlers may encounter emergencies such as crawler trap in the process of work. Reasonable processing of these conditions is a character of an excellent web crawler. Only when web crawlers have favorable robustness can they get back to work after interruption. Moreover the previously crawled content should be restored after setup.

(4) Friendliness: Web crawlers should protect relevant information of websites as per robots protocols. The crawling scope of web crawlers should be defined. Moreover additional burdens to websites should be avoided when web crawlers capture information.

(5) Updatability. Web crawlers should be able to perceive the alternation of websites and timely acquire new website content to replace the old one.

Information management system needs to collect and store diversified data on the Internet. With the explosive growth of data, the traditional stand-alone web crawlers have gradually been not as good as before. Hence stronger and more comprehensive information management systems are needed.

2.2 Hadoop

Hadoop, a basic framework of distributed system developed by Apache Software Foundation, is composed of many ordinary, low-cost single computers. It can rapidly and flexibly process mass data. It has the following advantages.

(1) High reliability. Its ability in processing data is highly reliable.

(2) Strong fault tolerance. Hadoop can automatically replicate many copes and allocate failed tasks.

(3) High scalability. Hadoop can process and allocate data between hundreds of servers and easily expand to thousands of nodes.

(4) High efficiency. Hadoop can efficiently transfer data between different nodes.

(5) Low cost. Compared to other commercial data warehouse, Hadoop is open-source.

Hadoop has two core parts. One is distributed file system, i.e. Hadoop Distributed File System (HDFS). HDFS is capable of storing large files, for example, files in a size of more than 100 TB. HDFS is also featured by strong fault tolerance. It can operate on low-cost hardware. The other core is MapReduce computational model which can concurrently calculate mass data and have favorable extensibility and fault tolerance. It has a huge advantage in data processing.

2.3 Application values of distributed web crawlers in information management system

In view of the advantages of distributed system and the properties of web crawlers, distributed web crawler is feasible. Distributed web crawler is composed of web crawler and distributed system, which is capable of fulfill different tasks by making the best use of information on the Internet. It effectively makes up the defects of the stand-alone web crawler. It can capture more websites and collect and store more data. Therefore Hadoop based distributed web crawler has high application values in information management system.

3 Design of information management system

3.1 Design of distributed web crawler system architecture

3.1.1 Design of physical architecture

To satisfy the aforementioned characteristics, cost of PC server should be saved, and moreover Hadoop based distributed architecture should be extensible [8]. The system should allocate the crawled page data on different nodes using its ability of distributed storage capacity. Moreover a strong fault tolerance was needed to set the number of data copies and reallocate the failed tasks on other nodes. The distributed architecture could enhance the overall performance of crawlers to the large extent.

The physical architecture of web crawlers in this study included Hadoop cluster and Storm cluster [9]. To reduce the pressure on Hadoop cluster during operation, separate deployment was adopted. Crawler tasks were divided into multiple tasks and operated on multiple Slave nodes based on the distributed storage and calculation abilities of distributed architecture. The collected data were stored in clusters. Then the data generated when crawlers crawled and analyzed webpage were written into Kafka, and Storm was used to calculate index results in real time. The physical architecture is shown in Figure 1.

3.1.2 Design of logic structure

The logic structure of distributed web crawlers is shown in Figure 2. It included batch processing part and real-time calculation part. Batch processing was mainly realized based on Hadoop platform, and it was responsible for achieving crawling tasks and storing data in Hbase. Real-time calculation was realized based on Storm platform, and it was responsible for calculating relevant data generated in system operation and storing the results in iRedis.

3.2 Modules of distributed web crawlers

The system module of the distributed web crawler was

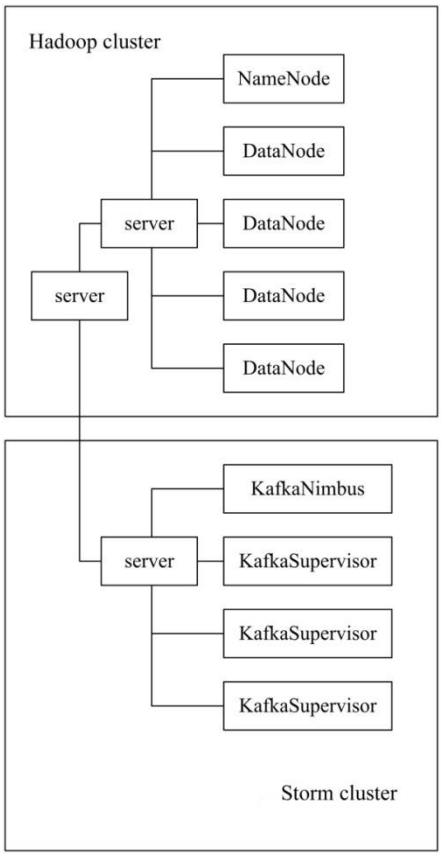


Figure 1: Design of the physical architecture of distributed web crawlers.

composed of the following parts.

(1) URL splitting and injection module: firstly read the URL path of user, then obtain URL list, and split it into several parts and allocate to TaskTracker.
(2) Webpage access module: acquire webpage according to URL links and download and save it locally.
(3) Webpage analysis module: analyze the captured webpage in aspects of structure and content.
(4) Link filtering module: filter the acquired URL and eliminate ineffective and repeated links.
(5) Data storage module: Save data in the database of HDFS.

3.3 Design of key technology

3.3.1 URL standardization

URL is a kind of character which can show information resources on www, and information resource has one and only has one URL [10]. URL standardization meant standardizing URL and transforming a URL to a qualified equivalent URL. Its transformation was realized by replacing /xx/../ with /, /../ with /, /./ with / and xx//yy with /.

3.3.2 Allocation of crawler tasks

Before crawling based on the distributed architecture, tasks were allocated to the distributed clusters [11]. When some node failed, tasks should be reallocated. For Hadoop cluster with n nodes, a URL was selected from URL set, Topn URLs were divided into N sets, and the sets were allocated to different nodes of Hadoop set to do crawling tasks. If some node failed, Master would allocate the failed task to other nodes without affecting the crawling speed of the current nodes. The network pressure of websites should be considered in the process of crawling.

3.3.3 Balance politeness

Crawling of the same website should follow the principle of balance politeness [12]. URLs were ranked according to score rules; then URL was taken out one by one from the URL set and allocated to N subsets; the number of URLs in one set and the number of URLs from the same Host in one set should be limited. In this way, the pressure of webpage could be reduced when web crawlers were crawling information.

3.3.4 Webpage revisit

Network usually has favorable dynamic property. When web crawlers fulfilled a crawling task, then the webpage might change. Therefore web crawlers should update website content at a certain time interval and the content which needed to be crawled.

3.3.5 Data deduplication

There are many same data on the network. Therefore network data should be processed by deduplication.

(1) Webpage content was separated into words, i.e.

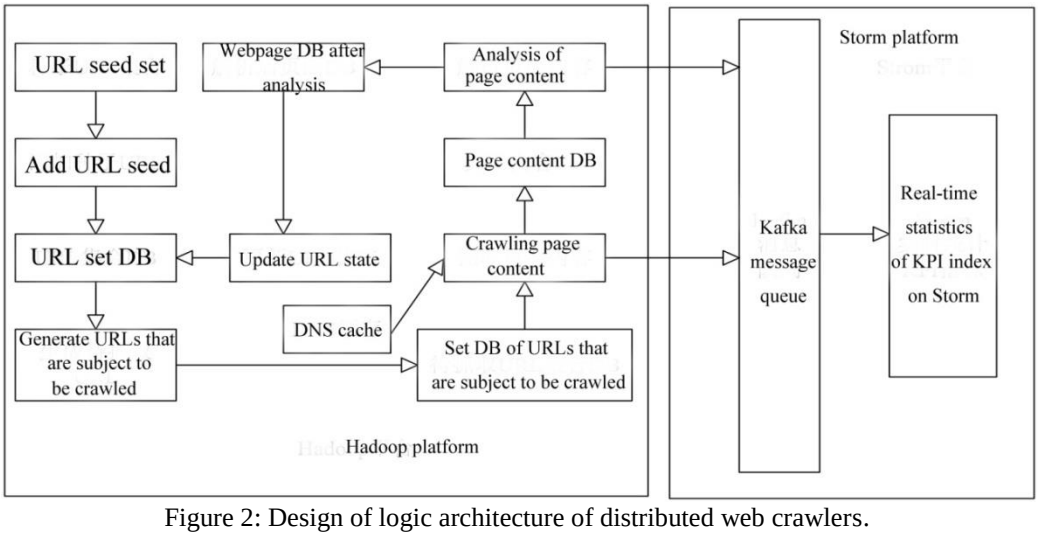


Figure 2: Design of logic architecture of distributed web crawlers.

characteristic vectors. The occurrence frequency of every word in documents was taken as weight.

(2) The Hash value of every characteristic vector was calculated [13], and moreover those vectors were processed by weighed accumulation.

(3) The result larger than 0 was denoted as 1 and otherwise as 0, and the final results were Simhash signature values [14].

(4) The similarity of data was determined according to different Simhash signature values.

4 Concrete implementation of distributed crawler

URL initial module was combined with parallel circulation model to analyze the procedures of URL insertion, URL list generation, web crawling and data update in the data crawling experiment of distributed crawler. A module circulation formed from link update in link library, crawl list generation, URL crawling execution, key information analysis to link update in link library. The module composition and flow circulation can benefit the concrete implementation of distributed crawler. The concrete implementation flow is shown in Figure 3.

5 System test and results analysis

Before testing of the network management, the test environment should be adjusted. VMware Workstation was installed in window host. Then Hadoop cluster was established in the virtual machine. In the development of the system, Java codes written by Eclipse IDE were installed in the host. Hadoop Eclipse plug-in units was installed and connected to Hadoop clusters. Data were processed using Hadoop Distributed File System (HDFS) and MapReduce calculation model.

5.1 Functional test

5.1.1 Test content and scheme

Functional test included the following content.

(1) Webpage crawling test
In the initial URL set, 0, 1 and 4 URL link seeds were added. Then three conditions, i.e. effective crawling, partially effective crawling and ineffective crawling, were considered. After crawling, whether the downloaded target data satisfied standards or not were checked.

(2) Filter test on URL link
The URL link log sheet which was subject to be crawled was checked to determine whether link standardization and deduplication operations should be performed or not.

(3) Webpage data extraction test
Whether the analysis module was corrected and could effectively extract data on webpage and store the data in relevant documents or not was checked.

(4) Test on webpage category classification
The system classified webpage into different categories and checked whether the classification was corrected or not.

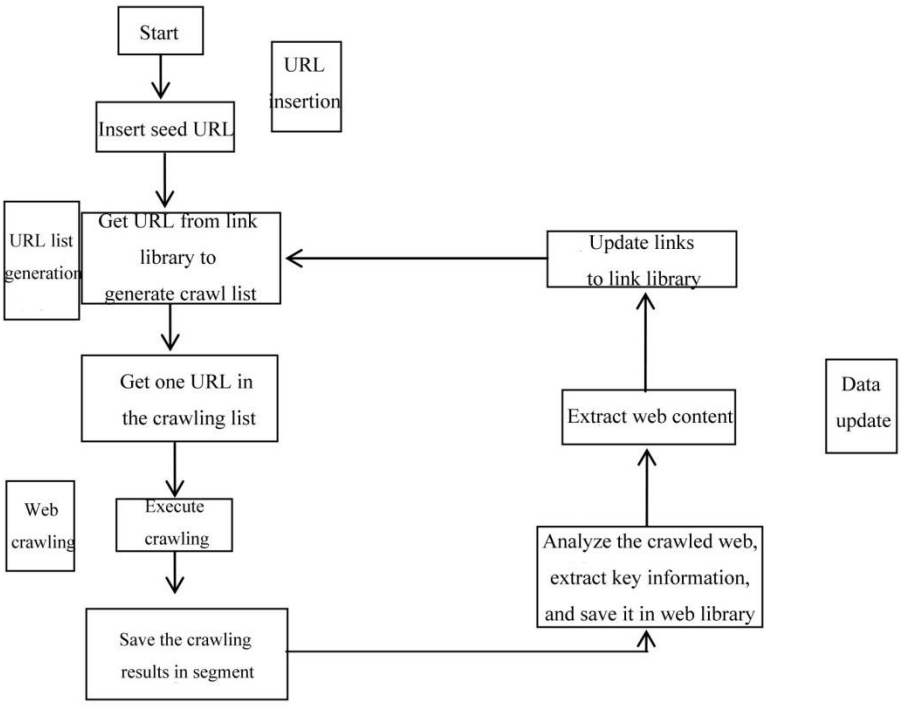


Figure 3: The implementation flow of the distributed crawler.

5.1.2 Test results

The system could do webpage crawling according to the prescribed initial URL set and added the crawled URLs into URLs which were subject to be crawled. Standardization and deduplication were performed before addition. The extracted data were stored in relevant documents. Moreover it could rapidly classify webpage.

5.2 Performance test

5.2.1 Test content and scheme

(1) Test on collection scale
After a period of webpage crawling, the size of the collected webpage data was calculated to measure the collection scale.

(2) Test on operation speed
During crawling, the size of the collected web data, i.e. x, was calculated after n hours of movement. The computational formula for crawling speed v was v = x/n.

5.2.2 Test results

Table 1 shows the data collection speed of the clusters based on four nodes. The operation of the system included webpage downloading, web analysis, extraction of record information on the network and classification of web text. This study could basically satisfy the requirements according to the data in Table 1.

5.3 Test on expandability

5.3.1 Test content and scheme

Test on expandability: the number of nodes on Hadoop platform was changed. Then test was performed when the number of coordinated nodes was 1, 2 and 3 to determine whether the operation was normal and what were the effects on the performance of the system.

5.3.2 Test results

Figure 4 demonstrated the data collection and analysis of the system when the time and number of nodes were different. It could be noted that the operation speed was the highest when there was only one node; the operation speed had remarkable improvement with the increase of nodes, but the speed of each node had no significant changes. Through test, it was concluded that the expandability could satisfy the predetermined requirements.

Internet plays an increasingly important role in the production and life of people and has been the main source of information. Distributed web crawlers can grab key data among mass data, which is greatly helpful to information acquisition. Bal et al. [15] put forward intelligent distributed crawler crawling network based on client-server architecture. In the architecture, load is managed by server. Every time when crawlers were loaded, URLs were dynamically allocated to allocate load to others, which enhanced the ability of information crawling. Kumar et al. [16] developed distributed semantic web crawlers and successfully crawled and

Table 1: The operation results of the information management system.

Number	1	2	3	4	5
Segment name	Segment20171002093417	Segment2017100213672	Segment2017100360349	Segment2017100413725	Segment2017100547436
Size (MB)	39.21	82.61	180.44	305.14	400.62
Operation time (h)	0.6	1.1	2.4	4.5	5.7

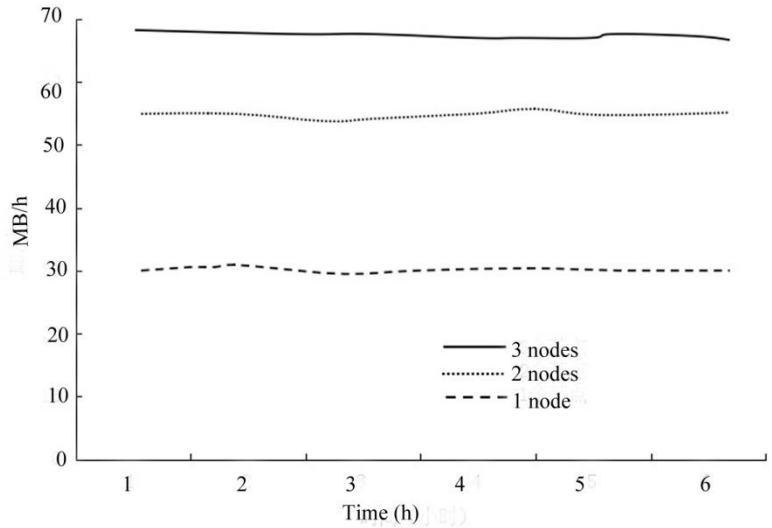


Figure 4: The test results of system expandability.

utilized HTML compiled by owl/RDf and semantic web. In information management system, distributed web crawlers can give full play to its advantages because it can effectively crawl information needed among mass data and efficiently collect and manage them. The application of distributed web crawlers can achieve efficient and safe management of information and has high practicability.

6 Conclusion

In conclusion, distributed network crawlers based information management system could precisely satisfy the requirements of web crawling, with a high performance and expandability. Moreover it can effectively reduce repeated visit and download of resources to improve efficiency of information searching. It can also reduce the time and money spent on resource acquisition because of the low cost. Therefore it can be applied for extracting network information. This work provides a reference for the application of distributed network crawlers based information management system in data extraction.

7 References

[1] Qin Y., Xuan H., Zhang B. (2016). Intelligent Management System of Power Network Information Collection Under Big Data Storage. *13th Global Congress on Manufacturing and Management (GCMM 2016), MATEC Web of Conferences*, Zhengzhou.

[2] Gupta C. L. P., Sharma S., Tripathi S. (2015). Importance of Management Information System in Electronic-Information Era. *East Carolina University*, 1(2).

[3] Zhao Q. A. (2016). Research and Implementation of Scientific Research Information Management System Based on the Topic Web Crawler. *Anhui: Anhui University*, pp. 1-46.

[4] Su L., Wang F. (2017). Web crawler model of fetching data speedily based on Hadoop distributed system. *IEEE International Conference on Software Engineering and Service Science*, Beijing, pp. 927-931.

[5] Zhang X., Xian M. (2015). Optimization of Distributed Crawler under Hadoop. *International Conference on Engineering Technology and Application*, 22:02029.

[6] Qu X., Hu R., Zhou L., Wang L., Zhu Q. (2015). Expert Achievements Model for Scientific and Technological Based on Association Mining. *International Symposium on Distributed Computing and Applications for Business Engineering and Science*, Guiyang, pp. 272-275.

[7] Bahrami M., Singhal M., Zhuang Z. (2015). A cloud-based web crawler architecture. *International Conference on Intelligence in Next Generation Networks*, Paris, pp. 2016-223.

[8] Pu Q. (2016). The Design and Implementation of a High-Efficiency Distributed Web Crawler. Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress, Auckland, pp. 100-104.

[9] Kim M., Han S., Cui Y., Lee, H. Cho H., S. Hwang. (2014). CloudDMSS: robust Hadoop-based multimedia streaming service architecture for a cloud computing environment. *Cluster Computing*, 17(3): 605-628.

[10] Bhagyashree E., Tanuja K. (2015). Phishing URL Detection: A Machine Learning and Web Mining-based Approach. *International Journal of Computer Applications*, 123.

[11] Santhosh K. D. K., Kamath M. (2014). Design and implementation of competent web crawler and indexer using web services. *International Conference on Advanced Communication Control and Computing Technologies*, Ramanathapuram, pp. 1672-1677.

[12] Dąbek Osb T. M. (2012). Strengthen the faith as the task of the Pastors of the Church. The Apostles Peter and Paul as examples for the Pastors of the Church for proclaim and, *Scriptura Sacra*, (16): 19.

[13] Dong C. (2015). Asymmetric color image encryption scheme using discrete-time map and hash value. *Optik - International Journal for Light and Electron Optics*, 126(20): 2571-2575.

[14] Qiao Y., Yun X., Zhang Y. (2016). Fast Reused Function Retrieval Method Based on Simhash and Inverted Index. *Trustcom/BigDatase/ISPA*, Tianjin, PP. 937-944.

[15] Bal S. K., Geetha G. (2016). Smart distributed web crawler. *International Conference on Information Communication and Embedded Systems*, Chennai, pp. 1-5.

[16] Kumar N. and Singh M. (2016). Framework for Distributed Semantic Web Crawler. *International Conference on Computational Intelligence and Communication Networks*, Jabalpur, pp. 1403-1407.

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of S**love**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please register as an author and submit a manuscript at: http://www.informatica.si. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica LaTeX format and figures in .eps format. Style and examples of papers can be obtained from http://www.informatica.si. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

SUBSCRIPTION

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: drago.torkar@ijs.si

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than twentyfour years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica web edition is free of charge and accessible at http://www.informatica.si.
Informatica print edition is free of charge for major scientific, educational and governmental institutions. Others should subscribe.

Informatica WWW:

http://www.informatica.si/

Referees from 2008 on:

A. Abraham, S. Abraham, R. Accornero, A. Adhikari, R. Ahmad, G. Alvarez, N. Anciaux, R. Arora, I. Awan, J. Azimi, C. Badica, Z. Balogh, S. Banerjee, G. Barbier, A. Baruzzo, B. Batagelj, T. Beaubouef, N. Beaulieu, M. ter Beek, P. Bellavista, K. Bilal, S. Bishop, J. Bodlaj, M. Bohanec, D. Bolme, Z. Bonikowski, B. Bokovi, M. Botta, P. Brazdil, J. Brest, J. Brichau, A. Brodnik, D. Brown, I. Bruha, M. Bruynooghe, W. Buntine, D.D. Burdescu, J. Buys, X. Cai, Y. Cai, J.C. Cano, T. Cao, J.-V. Capella-Hernndez, N. Carver, M. Cavazza, R. Ceylan, A. Chebotko, I. Chekalov, J. Chen, L.-M. Cheng, G. Chiola, Y.-C. Chiou, I. Chorbev, S.R. Choudhary, S.S.M. Chow, K.R. Chowdhury, V. Christlein, W. Chu, L. Chung, M. Ciglari, J.-N. Colin, V. Cortellessa, J. Cui, P. Cui, Z. Cui, D. Cutting, A. Cuzzocrea, V. Cvjetkovic, J. Cypryjanski, L. ehovin, D. erepnalkoski, I. osi, G. Daniele, G. Danoy, M. Dash, S. Datt, A. Datta, M.-Y. Day, F. Debili, C.J. Debono, J. Dedi, P. Degano, A. Dekdouk, H. Demirel, B. Demoen, S. Dendamrongvit, T. Deng, A. Derezinska, J. Dezert, G. Dias, I. Dimitrovski, S. Dobriek, Q. Dou, J. Doumen, E. Dovgan, B. Dragovich, D. Drajic, O. Drbohlav, M. Drole, J. Dujmovi, O. Ebers, J. Eder, S. Elaluf-Calderwood, E. Engstrm, U. riza Erturk, A. Farago, C. Fei, L. Feng, Y.X. Feng, B. Filipi, I. Fister, I. Fister Jr., D. Fier, A. Flores, V.A. Fomichov, S. Forli, A. Freitas, J. Fridrich, S. Friedman, C. Fu, X. Fu, T. Fujimoto, G. Fung, S. Gabrielli, D. Galindo, A. Gambarara, M. Gams, M. Ganzha, J. Garbajosa, R. Gennari, G. Georgeson, N. Gligori, S. Goel, G.H. Gonnet, D.S. Goodsell, S. Gordillo, J. Gore, M. Grar, M. Grgurovi, D. Grosse, Z.-H. Guan, D. Gubiani, M. Guid, C. Guo, B. Gupta, M. Gusev, M. Hahsler, Z. Haiping, A. Hameed, C. Hamzaebi, Q.-L. Han, H. Hanping, T. Hrder, J.N. Hatzopoulos, S. Hazelhurst, K. Hempstalk, J.M.G. Hidalgo, J. Hodgson, M. Holbl, M.P. Hong, G. Howells, M. Hu, J. Hyvrinen, D. Ienco, B. Ionescu, R. Irfan, N. Jaisankar, D. Jakobovi, K. Jassem, I. Jawhar, Y. Jia, T. Jin, I. Jureta, . Jurii, S. K, S. Kalajdziski, Y. Kalantidis, B. Kalua, D. Kanellopoulos, R. Kapoor, D. Karapetyan, A. Kassler, D.S. Katz, A. Kaveh, S.U. Khan, M. Khattak, V. Khomenko, E.S. Khorasani, I. Kitanovski, D. Kocev, J. Kocijan, J. Kollr, A. Kontostathis, P. Koroec, A. Koschmider, D. Koir, J. Kova, A. Krajnc, M. Krevs, J. Krogstie, P. Krsek, M. Kubat, M. Kukar, A. Kulis, A.P.S. Kumar, H. Kwanicka, W.K. Lai, C.-S. Laih, K.-Y. Lam, N. Landwehr, J. Lanir, A. Lavrov, M. Layouni, G. Leban, A. Lee, Y.-C. Lee, U. Legat, A. Leonardis, G. Li, G.-Z. Li, J. Li, X. Li, X. Li, Y. Li, Y. Li, S. Lian, L. Liao, C. Lim, J.-C. Lin, H. Liu, J. Liu, P. Liu, X. Liu, X. Liu, F. Logist, S. Loskovska, H. Lu, Z. Lu, X. Luo, M. Lutrek, I.V. Lyustig, S.A. Madani, M. Mahoney, S.U.R. Malik, Y. Marinakis, D. Marini, J. Marques-Silva, A. Martin, D. Marwede, M. Matijaevi, T. Matsui, L. McMillan, A. McPherson, A. McPherson, Z. Meng, M.C. Mihaescu, V. Milea, N. Min-Allah, E. Minisci, V. Mii, A.-H. Mogos, P. Mohapatra, D.D. Monica, A. Montanari, A. Moroni, J. Mosegaard, M. Mokon, L. de M. Mourelle, H. Moustafa, M. Moina, M. Mrak, Y. Mu, J. Mula, D. Nagamalai, M. Di Natale, A. Navarra, P. Navrat, N. Nedjah, R. Nejabati, W. Ng, Z. Ni, E.S. Nielsen, O. Nouali, F. Novak, B. Novikov, P. Nurmi, D. Obrul, B. Oliboni, X. Pan, M. Panur, W. Pang, G. Papa, M. Paprzycki, M. Parali, B.-K. Park, P. Patel, T.B. Pedersen, Z. Peng, R.G. Pensa, J. Per, D. Petcu, B. Petelin, M. Petkovek, D. Pevec, M. Piulin, R. Piltaver, E. Pirogova, V. Podpean, M. Polo, V. Pomponiu, E. Popescu, D. Poshyvanyk, B. Potonik, R.J. Povinelli, S.R.M. Prasanna, K. Pripui, G. Puppis, H. Qian, Y. Qian, L. Qiao, C. Qin, J. Que, J.-J. Quisquater, C. Rafe, S. Rahimi, V. Rajkovi, D. Rakovi, J. Ramaekers, J. Ramon, R. Ravnik, Y. Reddy, W. Reimche, H. Rezankova, D. Rispoli, B. Ristevski, B. Robi, J.A. Rodriguez-Aguilar, P. Rohatgi, W. Rossak, I. Roanc, J. Rupnik, S.B. Sadkhan, K. Saeed, M. Saeki, K.S.M. Sahari, C. Sakharwade, E. Sakkopoulos, P. Sala, M.H. Samadzadeh, J.S. Sandhu, P. Scaglioso, V. Schau, W. Schempp, J. Seberry, A. Senanayake, M. Senobari, T.C. Seong, S. Shamala, c. shi, Z. Shi, L. Shiguo, N. Shilov, Z.-E.H. Slimane, F. Smith, H. Sneed, P. Sokolowski, T. Song, A. Soppera, A. Sorniotti, M. Stajdohar, L. Stanescu, D. Strnad, X. Sun, L. ajn, R. enkek, M.R. ikonja, J. ilc, I. krjanc, T. tajner, B. ter, V. truc, H. Takizawa, C. Talcott, N. Tomasev, D. Torkar, S. Torrente, M. Trampu, C. Tranoris, K. Trojacanec, M. Tschierschke, F. De Turck, J. Twycross, N. Tziritas, W. Vanhoof, P. Vateekul, L.A. Vese, A. Visconti, B. Vlaovi, V. Vojisavljevi, M. Vozalis, P. Vraar, V. Vrani, C.-H. Wang, H. Wang, H. Wang, H. Wang, S. Wang, X.-F. Wang, X. Wang, Y. Wang, A. Wasilewska, S. Wenzel, V. Wickramasinghe, J. Wong, S. Wrobel, K. Wrona, B. Wu, L. Xiang, Y. Xiang, D. Xiao, F. Xie, L. Xie, Z. Xing, H. Yang, X. Yang, N.Y. Yen, C. Yong-Sheng, J.J. You, G. Yu, X. Zabulis, A. Zainal, A. Zamuda, M. Zand, Z. Zhang, Z. Zhao, D. Zheng, J. Zheng, X. Zheng, Z.-H. Zhou, F. Zhuang, A. Zimmermann, M.J. Zuo, B. Zupan, M. Zuqiang, B. alik, J. ika,

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2018 (Volume 42) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Slovene Society for Pattern Recognition (Simon Dobrišek)
Slovenian Artificial Intelligence Society (Mitja Luštrek)
Cognitive Science Society (Olga Markič)
Slovenian Society of Mathematicians, Physicists and Astronomers (Marej Brešar)
Automatic Control Society of Slovenia (Nenad Muškinja)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Stane Pejovnik)
ACM Slovenia (Matjaž Gams)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 42 Number 1 March 2018 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **42** (2018) Number 1, pp. 1–143