Using Bagging and Boosting Techniques for Improving Coreference Resolution

Smita Vemulapalli
Center for Signal and Image Processing (CSIP),
School of Electrical and Computer Engineering, Georgia Institute of Technology,
Atlanta, GA 30332, USA
E-mail: smita@ece.gatech.edu

Xiaoqiang Luo, John F. Pitrelli and Imed Zitouni
IBM T.J. Watson Research Center,
Yorktown Heights, NY 10598, USA
E-mail: {xiaoluo,pitrelli,izitouni}@us.ibm.com

Keywords: coreference resolution, information extraction, classifier combination, bagging, boosting, entity detection and tracking, majority voting

Received: February 5, 2009

Classifier combination techniques have been applied to a number of natural language processing problems. This paper explores the use of bagging and boosting as combination approaches for coreference resolution. To the best of our knowledge, this is the first effort that examines and evaluates the applicability of such techniques to coreference resolution. In particular, we (1) outline a scheme for adapting traditional bagging and boosting techniques to address issues, like entity alignment, that are specific to coreference resolution, (2) provide experimental evidence which indicates that the accuracy of the coreference engine can potentially be increased by use of multiple classifiers, without any additional features or training data, and (3) implement and evaluate combination techniques at the mention, entity and document level.

Povzetek: Kombiniranje učnih algoritmov je uporabljeno za iskanje koreferenc.

1 Introduction

Classifier combination techniques have been applied to many problems in natural language processing (NLP). Popular examples include the ROVER system [Fiscus1997] for speech recognition, the Multi-Engine Machine Translation (MEMT) system [Jayaraman and Lavie2005], and also part-of-speech tagging [Brill and Wu1998, Halteren et al.2001]. Even outside the domain of NLP, there have been numerous interesting applications for classifier combination techniques in the areas of biometrics [Tulyakov and Govindaraju2006], handwriting recognition [Xu et al.1992] and data mining [Aslandogan and Mahajani2004] to name a few. Most of these techniques have shown a considerable improvement over the performance of single-classifier baseline systems and, therefore, lead us to consider implementing such a multiple classifier system for coreference resolution as well. To the best of our knowledge, this is the first effort that utilizes classifier combination techniques for improving coreference resolution.

This study shows the potential for increasing the accuracy of the coreference resolution engine by combining multiple classifier outputs and describes the combination techniques that we have implemented to establish and tap into this potential. Unlike other domains where classifier combination has been implemented, the coreference resolution application presents a unique set of challenges that prevent us from directly using traditional combination schemes [Tulyakov et al.2008]. We, therefore, adapt some of these popular yet simple techniques to suit our application, and study the results of the implementation.

The main advantage of using combination techniques is that in cases where we have multiple classification engines, we do not merely use the classifier with highest accuracy, but instead, we combine all of the available classification engines attempting to achieve results superior to the single best engine. This is based on the assumption that the errors made by each of the classifiers are not identical and therefore if we intelligently combine multiple classifier outputs, we may be able to correct some of these errors.

The main contributions of this paper are:

- *demonstrating the potential for improvement in the baseline* – By implementing a system that behaves like an oracle, where we combine the outputs of several coreference resolution classifiers with knowledge of the truth *i.e.* the correct output generated by a human, we have shown that the output of the combination of multiple classifiers has the potential to be significantly higher in accuracy than any of the individual classifiers. This has been proven in certain other areas of NLP; here, we

have experimentally demonstrated the potential for this to be true in the area of coreference resolution.

- *adapting traditional bagging techniques for coreference resolution* – Multiple classifiers were generated from the same classification engine by subsampling the training-data set and the feature set. These classifiers were combined using entity-level sum rule and mention-level majority voting, after overcoming the problem of entity alignment between the classifier outputs.
- *implementing a document-level boosting algorithm for coreference resolution* – A document-level boosting algorithm was implemented in which a coreference resolution classifier was iteratively trained using a reweighted training set. Here, the training set is a set of documents, and since coreference resolution is performed for the entire document, the reweighting is done at the document level. This reweighting of the training set took into account the distribution of documents from different genres such as broadcast news, web logs and newswire articles.
- *addressing the problem of entity alignment* – To implement any combination technique for coreference resolution, we need to compensate for the fact that the number of entities and the number of mentions in each of the entities are different in the outputs of the coreference resolution classifiers to be combined. Therefore, there is the big challenge of aligning the entities before any of the traditional combination techniques may be implemented.

1.1 Organization of the paper

The remainder of this paper is organized as follows. In Section 2, we briefly describe the existing coreference resolution system and the data set used. Sections 3 and 4 present our adaptation of traditional bagging and boosting techniques. Section 5 contains an experimental evaluation of the proposed combination techniques. Section 6 discusses the related work. Finally, we conclude in Section 7 with suggestions for future work.

2 Coreference system and data

The terminologies used in the Automatic Content Extraction (ACE) task [NIST] are adopted in this paper: a *mention* is an instance of reference to an object, and the collection of mentions referring to the same object in a document form an *entity*. Coreference resolution is nothing but partitioning mentions into entities. For example, in the following sentence:

`John said Mary was his sister.`

there are four mentions: `John`, `Mary`, `his`, and `sister`. `John` and `his` belong to the same entity since they refer to the same person; so do `Mary` and `sister`. Furthermore, `John` and `Mary` are *named* mentions, `sister` is a *nominal* mention and `his` is a *pronominal* mention.

The basic coreference system is similar to the one described by Luo *et al.* [Luo et al.2004]. In such a system, the mentions in a document are processed sequentially, and at each step, a mention is either linked to one of existing entities, or used to create a new entity. At the end of this process, each possible partition of the mentions corresponds to a unique sequence of link or creation actions, each of which is scored by a statistical model. The one with the highest score is output as the final coreference result.

Experiments reported in the paper are done on the ACE 2005 data [NIST2005], which is available through the Linguistic Data Consortium (LDC). The dataset consists of 599 documents from rich and diversified sources (called *genres* in this paper), which include newswire articles, web logs, and Usenet posts, transcription of broadcast news, broadcast conversations and telephone conversations. We reserve the last 16% documents of each source as the test set, and use the rest of the documents as the training set. The number of documents, words, mentions and entities of this data split are tabulated in Table 1.

3 Bagging

One way to obtain multiple classifiers is via bagging or bootstrap aggregating, proposed by Breiman [Breiman1996] to improve the classification by combining outputs of classifiers that are trained using randomly-generated training sets. We have implemented bagging by using semi-randomly generated subsets of the entire training set and also subsets of the feature set.

3.1 Generation of multiple classifiers

In bagging, multiple classifiers are obtained by randomly generating subsets of the training set. Here, the training set refers to the set of documents that we use to train the system. When we subsample the training set, we do it at the document level.

We generated several classifiers by two techniques: the first is by semi-randomly sampling the training set and the second is by sampling the feature set. In the first technique, we try to sample the training set in a random fashion and generate a few classifiers by maintaining the initial distribution of the documents of different genres and a few others by not maintaining this distribution. In the second technique, we need to reduce the feature set and this is not done in a random fashion. Instead, we use our understanding of the individual features and also their relation to other features to decide which features may be dropped. In most of our experiments, we used classifiers in which either the training set or the feature set was subsampled, but not both.

Table 1: Statistics of ACE 2005 data: number of documents, words, mentions and entities in the training and test set.

DataSet	#Docs	#Words	#Mentions	#Entities
Training	499	253771	46646	16102
Test	100	45659	8178	2709
Total	599	299430	54824	18811

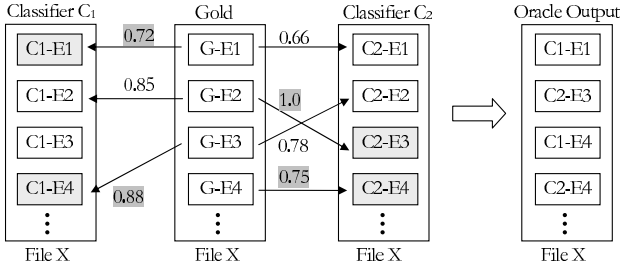

Figure 1: Working of the oracle

3.2 Oracle

In this paper, we refer to an *oracle* system which uses knowledge of the truth. In this case, truth, called *gold standard* henceforth, refers to mention detection and coreference resolution done by a human for each document. It is possible that this gold standard may have errors and is not perfect truth, but, as in most NLP systems, the human-annotated gold standard is considered the reference for purposes of evaluating computer-based coreference resolution.

To understand the oracle itself, consider an example in which we have two classifiers, and their outputs for the same input document are C_1 and C_2, as illustrated in Figure 1. The number of entities in C_1 and C_2 may not be the same and even in cases where they are, the number of mentions in corresponding entities may not be the same. In fact, even finding the corresponding entity in the other classifier or in the gold standard output G is not a trivial problem and requires us to be able to align any two classifier outputs.

The alignment between any two coreference labelings, say C_1 and G, for a document is done by finding the best one-to-one map between the entities of C_1 and the entities of G, using the algorithm explained by Luo [Luo2005]. To align the entities of C_1 with those of G, under the assumption that an entity in C_1 may be aligned with at most only one entity in G and vice versa, we need to generate a bipartite graph between the entities of C_1 and G. Now the alignment task is a maximum bipartite matching problem. This is solved by using the Kuhn-Munkres algorithm [Kuhn1955, Munkres1957]. The weights of the edges of the graph, in this case, are entity-level alignment measures. The metric we use is a relative measure of the similarity between the two entities. To compute the similarity metric $\phi(R, S)$ for the entity pair (R, S), we use the formula shown in Equation 1, where the intersection $(\cap)$ is the commonality with attribute-weighted partial scores. Attributes are things such as (ACE) entity type, subtype, entity class, etc.

$$\phi(R, S) = \frac{2\,|R \cap S|}{|R| + |S|} \tag{1}$$

The oracle output is a combination of the entities in C_1 and C_2 with the highest entity-pair alignment measures with the entities in the gold standard G. We can see in Figure 1 that the entity G-E1 is aligned with entities C1-E1 and C2-E1. We pick the entity with the highest entity-pair alignment measure (highlighted in gray in Figure 1) with the corresponding gold standard entity which, in this case, is C1-E1. This is repeated for every entity in G. The oracle output can be seen in the right-hand side of Figure 1. This technique can be scaled up to work for any number of classifiers.

3.3 Preliminary classifier combination approaches

Imitating the oracle. Making use of the existing framework of the oracle, we implement a combination technique that imitates the oracle except that in this case, we do not have the gold standard. If we have N classifiers C_i, $i = 1$ to N that we plan to combine, then we replace the gold standard by each of the N classifiers in succession, to get N outputs $Comb_i$, $i = 1$ to N.

The task of generating multiple classifier combination outputs that are of a significantly higher accuracy than the original classifiers is often considered to be easier than the task of finding out which of the output classifiers is highest-accuracy to pick as the final output. We used the formulas in Equations 2, 3 and 4 to assign a score S_i to each of the N combination outputs $Comb_i$ obtained, and then we pick the one with the highest score. The function Sc gives the similarity between the entities in the pair (R, S). Here, we have used the function ϕ in Equation 1 to compute the similarity between the entity-pair that forms the argument of the function Sc.

$$S_i = \frac{1}{N-1} \sum_{j \neq i} Sc(Comb_i, C_j) \tag{2}$$

$$S_i = Sc(Comb_i, C_j) \tag{3}$$

$$S_i = \frac{1}{N-1} \sum_{j \neq i} Sc(Comb_i, Comb_j) \tag{4}$$

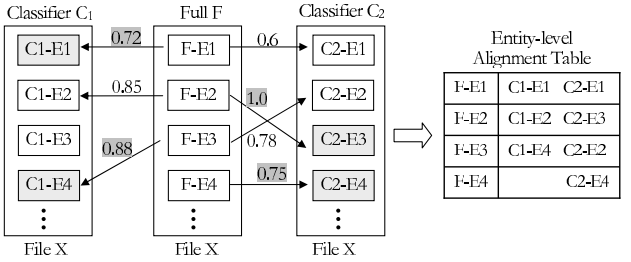

Figure 2: Entity alignment between classifier outputs

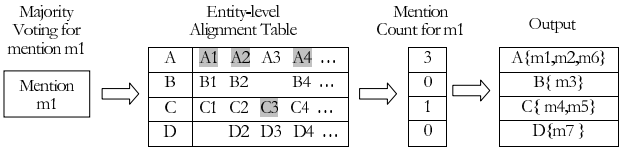

Figure 3: Mention-level majority voting

Entity-level sum-rule. We implemented a basic sum-rule at the entity level, where we generate only one combination classifier output by aligning the entities in the N classifiers and picking only one entity at each level of alignment. In the oracle, the reference for entity-alignment was the gold standard and here, since the gold standard is not available, we make use of the full system to do this. The full system is the baseline system where the entire training set and feature set have been used. The entity-level alignment has been represented as a table in Figure 2.

Let A_i, $i = 1$ to M be the aligned entities in one row of the table in Figure 2. Here, $M \leq N$ if we exclude the baseline from the combination and $M \leq N + 1$ if we include it. To pick one entity out of these M entities, we use traditional sum rule [Tulyakov et al.2008], shown in Equation 5, to compute the $S(A_i)$ for each A_i and pick the entity with the highest $S(A_i)$ value. Again, we use the function ϕ in Equation 1 to compute $Sc(A_i, A_j)$.

$$S(A_i) = \sum_{j \neq i} Sc(A_i, A_j) \tag{5}$$

3.4 Mention-level majority voting

In the previous techniques, entities are either picked or rejected as a whole but never broken down further. In the mention-level majority voting technique, we work at the mention level, so the entities created after combination may be different from the entities of all the classifiers that are being combined.

As shown in Figure 3, we have made use of the entity-level alignment table. This table is generated by aligning the entities output by the classifiers with the baseline system, as explained in the Section 3.3. In the entity-level alignment table, A, B, C and D refer to the entities in the baseline system and A1, A2, ..., D4 represent the entities of the input classifiers that are aligned with each of the baseline classifier entities. Majority voting is done by counting the number of times a mention is found in a set of aligned entities. So for every row in the table, we have a mention count. The row with the highest mention count is assigned the mention in the output. This is repeated for each mention in the document. In Figure 3, we are voting for the mention m1, which is found to have a voting count of 3 at the entity level A and a count of 1 at the entity-level of C, so the mention is assigned to the entity A as it has the majority vote. It is important to note that some entities of the classifiers may not align with any of the baseline classifier's entities as we allow only a one-to-one mapping during alignment. This leads to some entities not being a part of the alignment table. If this number is large, it may have a considerable effect on the combination.

4 Boosting

Unlike bagging techniques, the document-level boosting algorithm that we have implemented is adaptive in nature. The training set of the classifier is adaptively reweighted based on the performance of the previous classifiers. Since coreference resolution is done for a whole document, we can not split a document further. So when we reweight the training set, we are actually reweighting the documents. Figure 4 shows the overview of the document-level boosting algorithm.

The decision of which documents to boost is made using two thresholds: percentile threshold P_{thresh} and the F-measure threshold F_{thresh}. Documents in the test set that are in the lowest P_{thresh} percentile and that have a document F-measure less than F_{thresh} will be boosted in the training set for the next iteration. We shuffle the training set to create some randomness and then divide it into groups of training and test sets in a round-robin fashion such that a predetermined ratio of the number of training documents to the number of test documents is maintained. In Figure 4, the light gray regions refer to the training documents and the dark gray regions refer to the test documents. Another important consideration is that it is difficult to achieve good coreference resolution performance on documents of some genres compared to others, even if they are boosted significantly. In an iterative process, it is likely that documents of such genres will get repeatedly boosted. Also our training set has more documents of some genres and fewer of others. So we try to maintain, to some extent, the ratio of documents from different genres in the training set while splitting this training set further into groups of training and test sets.

5 Evaluation

This section describes the general setup used to conduct the experiments and presents an evaluation of the combination techniques that were implemented.

Experimental setup. The coreference resolution system used in our experiments makes use of a Maximum En-

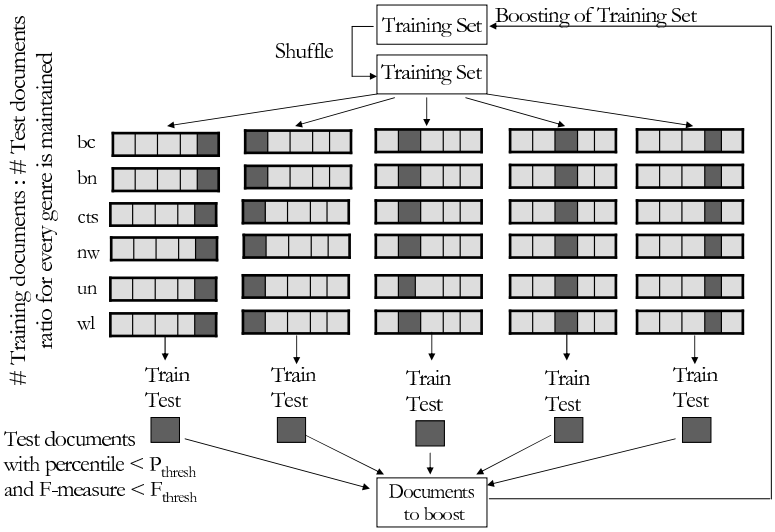

Figure 4: Document-level boosting

Table 2: Accuracy of the generated and baseline classifiers

Classifier	Accuracy (%)
$C_1 - C_{15}$ Average	77.52
Highest	79.16
Lowest	75.81
C_0 Baseline	78.53

tropy model which has lexical, syntactical, semantic and discourse features [Luo et al.2004]. The data set used here, which is split into the training and test sets, is part of the ACE 2005 data [NIST2005]. A short description of this data set may be found in Section 2 of this paper. For the purpose of evaluation, we make use of the human-annotated gold standard, described in Section 3.2, as the reference.

5.1 Bagging

The classifiers for the following experiments were generated using bagging techniques described in Section 3.1. A total of 15 classifiers (C_1 to C_{15}) were generated, 12 of which were obtained by semi-random sampling of the training set and the remaining 3 by sampling of the feature set. We also make use of the baseline classifier C_0, which was trained using the full training and feature sets. The accuracy of classifiers C_0 to C_{15} has been summarized in Table 2. The agreement between the generated classifiers' output was found to be in the range of 93% to 95%. In this paper, the metric used to compute the accuracy of the coreference resolution is the Constrained Entity-Alignment F-Measure (CEAF) [Luo2005] with the entity-pair similarity measure in Equation 1.

Oracle. To conduct the oracle experiment described in Section 3.2, we train 1 to 15 classifiers, whose output are aligned to the gold standard. For all system-generated entities aligned with a gold entity, we pick the one with the highest score as the output. We measure the performance for varying number of classifiers, and the result is plotted in Figure 5.

First, we observe a steady and significant increase in CEAF for every additional classifier. This is not surprising since an additional classifier can only improve the alignment score. Second, it is interesting to note that the oracle performance is 87.58% for a single input classifier C_1, *i.e.* an absolute gain of 9% compared to the baseline. This is because the availability of gold entities makes it possible to remove many false-alarm entities. Finally, the performance of the oracle output when all 15 classifiers (C_1 to C_{15}) are used as input is 94.59%, a 16.06% absolute improvement.

The oracle experiment is a "cheating" one since the gold standard is used. Nevertheless, it helps us understand the performance bound of combining multiple classifiers and the quantitative contribution of every additional classifier.

Preliminary classifier combination approaches. While the oracle result is encouraging, a natural question is. how much performance gain can be attained if the gold standard is not available. To answer this question, we replace the gold standard with one of the 15 classifiers C_1 to C_{15}, and align the rest classifiers. This is done in a round robin fashion as described in Section 3.3. The best performance of this procedure is 77.93%. The sum-rule combination output had an accuracy of 78.65% with a slightly different baseline of 78.81%. In other words, these techniques do not yield a statistically significant increase in CEAF relative to the baseline. This is not entirely surprising as the the 15 classifiers C_1 to C_{15} are highly correlated.

Mention-level majority voting. This experiment is conducted to evaluate and understand the mention-level majority voting technique for coreference resolution. Compared with the baseline, the results of this experiment are not statistically better, but they give us valuable insight into

the working of the combination technique. The example in Figure 6 shows the contents of a single entity-alignment level for the full system C_0 and 3 classifier outputs C_1, C_2, and C_3 and the combination output by mention-level majority voting. The mentions are denoted by the notation *'EntityID - MentionID'*, for example *7-10* is the mention with *EntityID=7* and *MentionID=10*. Here, we use the *EntityID* in the gold file. The mentions with *EntityID=7* are "correct" *i.e.* they belong in this entity, and the others are "wrong" *i.e.* they do not belong in this entity.

The aligned system mentions are of the following four types:

- *Type I mentions* – These mentions have a highest voting count of 2 or more at the same entity alignment level and therefore appear in the output.
- *Type II mentions* – These mentions have a highest voting count of 1 and are also present in more than one input classifier. So, there is a tie between the mention counts for a single mention at different entity alignments. The rule to break the tie is that mentions are included if they are also seen in the full system C_0. As can been seen, this rule brings in correct mentions such as *7-61, 7-63, 7-64*, but it also admits *20-33,20-39* and *20-62*. This is a fundamental difference between the oracle and real experiment: in the oracle, the gold standard helps to remove entities with false-alarm mentions, while the full system output itself is noisy and it is not strong enough to reliably remove undesired mentions.
- *Type III mentions* – There is only one mention *20-66* which is of this type. It is selected in the combination output since it is present in C_2 and the baseline C_0, although it has been rejected as a false-alarm in C_1 and C_3.
- *Type IV mentions* – These mentions are false-alarm mentions (relative to C_0) and are rejected in the output.

  As can be seen, this correctly rejects mentions such as *15-22* and *20-68*, but it also rejects correct mentions *7-18, 7-19* and *7-30*.

In summary, the current implementation of the mention-level majority voting technique has a limited ability to distinguish correct mentions from wrong ones due to the noisy nature of the full system C_0 which is used for alignment. We also observe that mentions spread across different alignments often have low-count and they are often tied in count. Therefore, it is important to set a minimum threshold for accepting these low-count majority votes and also investigate better tie-breaking techniques.

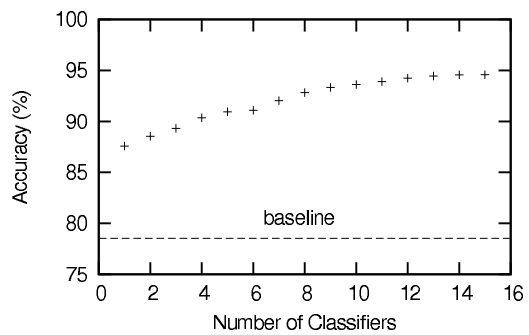

Figure 5: Oracle performance vs. number of classifiers

Table 3: Results of document-level boosting

Iteration	Accuracy (%)
1	78.53
2	78.82
3	79.08
4	78.37

5.2 Boosting.

This experiment is conducted to evaluate the document-level boosting technique for coreference resolution. Table 3 shows the results of this experiment with the ratio of the number of training documents to the number of test documents equal to 80:20, F-measure threshold F_{thresh} = 74% and percentile threshold P_{thresh} = 25%. The accuracy increases by 0.7%, relative to the baseline. Due to computational complexity considerations, we used fixed values for the parameters. Therefore, these values may be sub-optimal and may not correspond to the best possible increase in accuracy.

6 Related work

A large body of literature related to statistical methods for coreference resolution is available [Ng and Cardie2003, Yang et al.2003, Ng2008, Poon and Domingos2008, McCallum and Wellner2003]. Poon and Domingos [Poon and Domingos2008] use an unsupervised technique based on joint inference across mentions and Markov logic as a representation language for their system on both MUC and ACE data. Ng [Ng2008] proposed a generative model for unsupervised coreference resolution that views coreference as an EM clustering process. In this paper, we make use of a coreference engine similar to the one described by Luo *et al.* [Luo et al.2004], where a Bell tree representation and a Maximum entropy framework are used to provide a naturally incremental framework for coreference resolution. To the best of our knowledge, this is the first effort that utilizes classifier combination techniques to improve coreference resolution. Combination techniques have earlier been applied to various applications including machine translation [Jayaraman and Lavie2005] and part-of-speech tagging [Brill and Wu1998]. However, the use of these techniques for coreference resolution presents a unique set of challenges, such as the issue of entity alignment

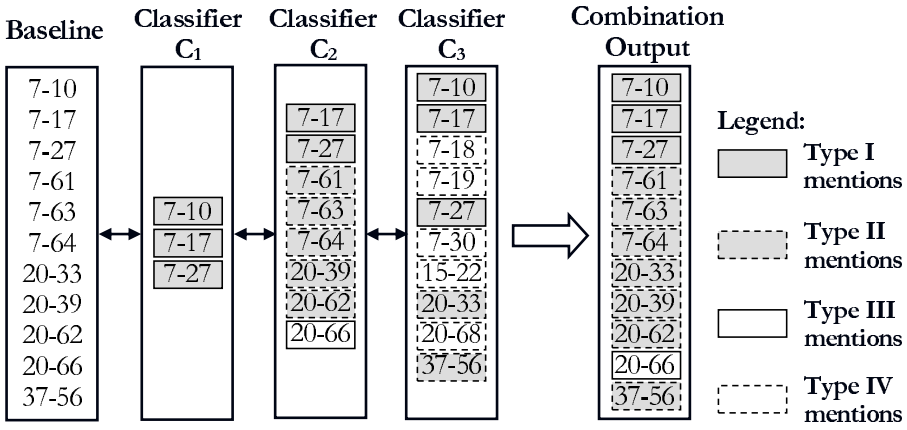

Figure 6: A real example showing the working of mention-level majority voting

between the multiple classifier outputs.

7 Conclusions and future work

This paper examined and evaluated the applicability of various bagging and boosting techniques to coreference resolution. In this paper, we also provided empirical evidence that coreference resolution accuracy can potentially be improved by making use of multiple classifiers. We proposed and evaluated new approaches to well-known classifier combination techniques that work at the mention, entity and document level. In future, we plan to work on a better alignment strategy and also explore various possibilities for improving mention-level majority voting such as setting a minimum threshold for the majority-vote and better tie-breaking. We would also like to work on further development of the document-level boosting algorithm to automatically find optimal values for the parameters that have been manually set in this paper. Another possible avenue for future work would be to test these combination techniques with other coreference resolution systems.

Acknowledgement

The authors would like to acknowledge Ganesh N. Ramaswamy for his guidance and support in conducting the research presented in this paper.

References

[Breiman1996] L. Breiman. 1996. Bagging predictors. In *Machine Learning*.

[Brill and Wu1998] E. Brill and J. Wu. 1998. Classifier combination for improved lexical disambiguation. In *Proc. of COLING*.

[Fiscus1997] J. Fiscus. 1997. A post-processing system to yield reduced word error rates: Recogniser output voting error reduction (rover). In *Proc. of ASRU*.

[Halteren et al.2001] H. Van Halteren, J. Zavrel, and W. Daelemans. 2001. Improving accuracy in word class tagging through the combination of machine learning systems. *Computational Linguistics*, 27.

[Jayaraman and Lavie2005] S. Jayaraman and A. Lavie. 2005. Multi-engine machine translation guided by explicit word matching. In *Proc. of ACL*.

[Kuhn1955] H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2.

[Luo et al.2004] X. Luo, A. Ittycheriah, H. Jing, N. Kambhatla, and S. Roukos. 2004. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proc. of ACL*.

[Luo2005] X. Luo. 2005. On coreference resolution performance metrics. In *Proc. of EMNLP*.

[McCallum and Wellner2003] A. McCallum and B. Wellner. 2003. Toward conditional models of identity uncertainty with application to proper noun coreference. In *Proc. of IJCAI/IIWeb*.

[Munkres1957] J. Munkres. 1957. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1).

[Ng and Cardie2003] V. Ng and C. Cardie. 2003. Bootstrapping coreference classifiers with multiple machine learning algorithms. In *Proc. of EMNLP*.

[Ng2008] V. Ng. 2008. Unsupervised models for coreference resolution. In *Proc. of EMNLP*.

[NIST2005] NIST. 2005. ACE'05 evaluation. `www.nist.gov/speech/tests/ace/ace05/index.html`.

[Poon and Domingos2008] H. Poon and P. Domingos. 2008. Joint unsupervised coreference resolution with Markov Logic. In *Proc. of EMNLP*.

[Schapire1999] R.E. Schapire. 1999. A brief introduction to boosting. In *Proc. of IJCAI*.

[Tulyakov et al.2008] S. Tulyakov, S. Jaeger, V. Govindaraju, and D. Doermann. 2008. Review of classifier combination methods. In *Machine Learning in Document Analysis and Recognition*.

[Yang et al.2003] X. Yang, G. Zhou, J. Su, and C. L. Tan. 2003. Coreference resolution using competition learning approach. In *Proc. of ACL*.

[Aslandogan and Mahajani2004] Y.A. Aslandogan and G.A. Mahajani. 2004. Evidence combination in medical data mining. In *Proc. of ITCC*, volume 2.

[Tulyakov and Govindaraju2006] Sergey Tulyakov and Venu Govindaraju. 2006. Classifier combination types for biometric applications. In *Proc. of CVPR Workshop*.

[Xu et al.1992] L. Xu, A. Krzyzak, and C.Y. Suen. 1992. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE Transactions on Systems, Man and Cybernetics*, 22(3), May/Jun.

[NIST] NIST. The ACE evaluation plan. `www.nist.gov/speech/tests/ace/index.html`.