Justifying Convolutional Neural Network with Argumentation for Explainability

Saung Hnin Pwint Oo^{1,*}, Nguyen Duy Hung^{1,*} and Thanaruk Theeramunkong^{1,2}
^{1}School of Information, Computer, and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Pathum Thani, Thailand
^{2}The Royal Society of Thailand, Sanam Suea Pa, Dusit, Bangkok, Thailand
E-mail: hnin.pwint172004@gmail.com, hung@siit.tu.ac.th, thanaruk@siit.tu.ac.th
*Corresponding authors

Keywords: Sentiment analysis, convolutional neural networks, similarity measure, case-based reasoning, argumentation, explainable artificial intelligence

Received: August 26, 2022

Convolutional neural network (CNN) has emerged as one of the most accurate methods for sentiment analysis, but it is largely uninterpretable, while case-based reasoning (CBR) is less accurate but offers interpretable outputs in the form of arguments from analogy. This paper presents an approach to combine these two methods, CNN for accuracy and CBR for explainability, using an assumption-based argumentation (ABA) framework. Our approach focuses on justifying CNN outputs using analogous sentences from CBR while ensuring that the combined process is argumentative and hence self-explainable. To demonstrate the proposal, we construct a CNN model $\mathcal{M}_1$ *and a CBR model* $\mathcal{M}_2$ *for sentiment analysis using different subsets of a dataset of which the remaining part is used for testing and comparing these input models with combined models. For an input sentence, if* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *predict the same sentiment, then the analogous sentence, which* $\mathcal{M}_2$ *finds, is used to explain the sentiment. If they give conflicting sentiments, a hybrid model* $\mathcal{M}_3$ *determines which one should be followed using a system of strict rules that takes into account how assertive* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *are. Another hybrid model* $\mathcal{M}_4$*, which is implemented by an ABA framework, improves on* $\mathcal{M}_3$ *by considering the probability distribution of the set of all labels from* $\mathcal{M}_1$*, and the second (or third) most similar sentences from* $\mathcal{M}_2$*.* $\mathcal{M}_3$ *and* $\mathcal{M}_4$ *preserve the accuracy of the CNN model (specifically,* 88.32% *and* 88.28% *in comparison with the* 87.59% *accuracy of the CNN). They justify* 69.95% *and* 74.53% *of CNN outputs, respectively.*

Povzetek: *Obravnavana je analiza emisij termoelektrarn z metodami mehkih množic in učenjem na osnovi primerov.*

1 Introduction

Sentiment analysis (SA) [1, 2] is an area of natural language processing (NLP) that analyzes extremely large textual datasets, such as customer reviews, to determine writers' sentiments as expressed in a text. Classical approaches to SA are mostly machine learning (ML) techniques (such as naïve bayes (NB) and support vector machine (SVM), maximum entropy (MaxEnt) and logistic regression (LR) [3]) [4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Deep learning (DL) methods [14, 15, 16, 17] (especially CNN) emerged as one of the most accurate methods [18, 19]. Unfortunately, for humans, CNN is like a black-box, offering virtually no explanation of why an input sentence should be labeled in the way it is. In the meantime, decision tree (DT) and case-based reasoning (CBR) are interpretable but cannot compete with CNN in terms of accuracy. Explainable artificial intelligence (XAI) [20] is a recent emerging AI field to address the uninterpretable problem of ML/DL models by either model-specific (intrinsic) explanation or model-agnostic (post-hoc) explanation. The intrinsic explanation mostly interprets the simple and transparent (white-box) models such as DT and rule-based systems. However, the complicated and well-trained black-box models lose their transparency while having high accuracies. These black-box models cannot be directly interpreted because it is difficult to access their internal mechanisms. Hence, the post-hoc explanation uses model-agnostic methods (*i.e.* interpretable) to justify the black-box models at either the model level (global) or the instance level (local) after the training process. The global model-agnostic methods (such as Partial Dependence Plot (PDP)) compute the measures of individual features that are critical and effective to the overall model performance. For example, PDP describes the average behavior of a trained model by computing the contribution of the individual features to the outputs. When a model has hyperparameters, the local model-agnostic methods (such as local interpretable model-agnostic explanations (LIME)) are more beneficial than the global methods for interpreting the classification decisions of the model. For example, LIME [21] figures out the relationship between input-output pairs of a trained model and takes each pair as an instance. Then it explains individual classification for each specific instance. The advantage of using local model-agnostic methods is that they can prevent sacrificing accuracy for achieving interpretability.

In this paper, we propose hybrid models of CNN and CBR that combine their strengths, *i.e.* accuracy from CNN and explainability from CBR, using the ABA framework. The goal of this paper is to justify the CNN outputs using analogous sentences from the CBR while ensuring that the combination of CNN and CBR models is argumentative. Similar to the local model-agnostic methods, our hybrid models primarily explain the sentiment classification of the CNN model at the instance level. The motivation for using the CBR for interpretability is that it provides an easier and more satisfying explanation than a chain of rules, and the model's failure can be easily diagnosed because its outcomes can be traced back to prior analogous sentences. Our approach is graphically depicted in Figure 1.

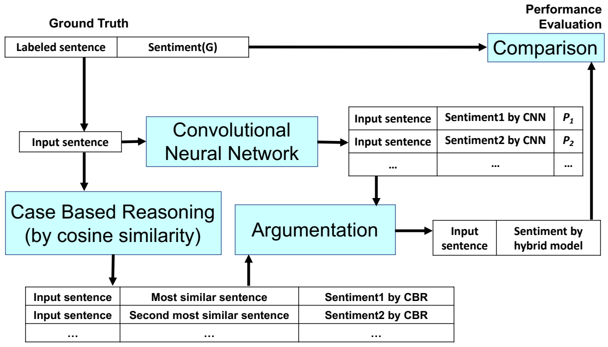

Figure 1: Overview of our hybrid approach where P_i refers to the conditional probability of i^{th} sentiment given the input sentence.

In order to test our model, we construct CNN and CBR models using a dataset containing 25,104 annotated sentences (*i.e.* ground truth), which are customer reviews written in the Myanmar language. Specifically, the entire dataset $\mathcal{D}$ is divided into three subsets: 61% of $\mathcal{D}$ (denoted by $\mathcal{D}_1$) is used to train the CNN model (called $\mathcal{M}_1$), 18% of $\mathcal{D}$ (denoted by $\mathcal{D}_2$) is used to build the CBR model (called $\mathcal{M}_2$), and the rest (21%) of $\mathcal{D}$ (denoted by $\mathcal{D}_3$) is used to test the models, including our hybrid models. For an input sentence n, the CNN model $\mathcal{M}_1$ using *softmax* as the squashing function in the output layer produces a probability distribution of sentiments. The CBR model $\mathcal{M}_2$ searches in $\mathcal{D}_2$ for similar sentences to the input sentence denoted by n and ascribes the labels of those sentences to n (often, the most similar sentence is used). If $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on the label of n, then it is clear that the hybrid models use that label as well. Moreover, the most similar sentence that $\mathcal{M}_2$ finds can be used to explain the label in this case. This means that the explanation takes the form of an argument from analogy. In the case of disagreement, the hybrid model needs to decide whether to follow $\mathcal{M}_1$ or $\mathcal{M}_2$. In this paper, we first present a rule-based hybrid model called $\mathcal{M}_3$ using a system of strict rules to make this decision. We then revise $\mathcal{M}_3$ by adding exceptions to some strict rules of $\mathcal{M}_3$, to make finer decisions, taking into account the labels of the second-most, third-most (*etc.*) similar sentences of the input sentence. The result is a new model $\mathcal{M}_4$, which is implemented by a structured argumentation framework called ABA. Thus, both hybrid models, $\mathcal{M}_3$ and $\mathcal{M}_4$, justify $\mathcal{M}_1$'s outputs via analogous sentences from $\mathcal{M}_2$. Hence, the presence of sentiment words (*i.e.* positive, negative) in the sentences derives the model's classification. We compare the performance of all models using the same dataset $\mathcal{D}_3$. The results show that our hybrid models are 88.32% and 88.28% accurate, respectively, while the CNN model and the CBR model achieve 87.59% and 71.57% accuracy, respectively. Hence, we can say that the hybrid models are on a par with CNN in terms of accuracy but are more interpretable. As will be seen, the hybrid models can explain around 70% and 74% of their outputs, respectively.

This paper contributes to XAI in three ways: (i) it uses CBR to justify CNN outputs; (ii) it ensures that the CNN-CBR combination is argumentative and self-explainable through argumentation; and (iii) it retains CNN accuracy while providing explainability. The interpretability of our approach is primarily post-hoc. In [22], Prakken and Ratsma describe a top-level model that explains the outputs of ML-based decision-making applications. Their work's interpretability is also post-hoc based on AI and law studies using argumentation with cases, and it can be extended with more detailed analyses of case similarities. Similar to their work, we also develop hybrid models built on top of the CNN model for justifying CNN outputs using analogous sentences from the CBR model, which is natural by considering input data to the model as cases. We use cosine similarity in the CBR model to compute the similarity between sentences and applies to a different domain (text analysis). Meanwhile, the existing local model-agnostic methods, such as LIME do not interpret the internal process of the black-box models. Instead, LIME deals with input-output pairs of an ML model, such as $f(x) = y$ for a given point (x, y) of the model f, where x is an instance and y is a target. It generates a dataset $\mathcal{D}_x = \{(x', y')|y' = f(x')$,where x' is a point near $x\}$, and then uses $\mathcal{D}_x$ to train an interpretable model. Like LIME, our approach also takes input-output pairs of the CNN model for each sentence. The CBR model is built using the different dataset $\mathcal{D}_2$, which computes the similarity between the input and analogous sentences. These analogous sentences then justify the output of the input sentence (*i.e.* the CNN's sentiment classification).

Let us briefly review computational argumentation and the recent line of work on XAI that leverages the wide array of reasoning abstractions and explanation delivery methods of argumentation, hence called Argumentative XAI [23]. Computational argumentation (as it is studied in AI) is inspired by the human ability to reach conclusions via the exchange of arguments. Recent developments in the field are greatly influenced by Dung's AA framework (recalled in Section 3.3), which is defined simply as a pair (Ar, Att) of a set of arguments Ar and a binary attacking relation

Att between arguments. As mentioned above, in this paper, we use ABA, an instance of AA that treats arguments as deductive proofs from assumptions using inference rules. As discussed in [23], extension-based semantics of AA, as well as ABA, can be equivalently understood in terms of dispute trees [24] which have formed the basis of several approaches to argumentation-based XAI, for example, by providing content for the so-called dialogical explanations. This paper contributes to Argumentative XAI, but we restrict ourselves to the interpretable model combination. For an illustration, consider the following input sentence: "The monthly internet package is affordable and sufficient for the average user." Although both the CNN and CBR models predict that the sentence has a positive sentiment, the CNN gives no justification while the CBR basically says that the sentence is similar to another sentence "I recommend a monthly unlimited internet package with a reasonable price and basic speed" in terms of individual words. Since the latter has been assigned a positive sentiment, the former should be assigned the same sentiment. Hence, we could "borrow" this argument from analogy to explain the decision of the CNN. When the CNN and CBR models predict different sentiments for the same input sentence, we need to decide which model we shall follow. If we follow the CBR, we can still use the argument from analogy but bear a greater risk of the wrong prediction since in general, the CBR is less accurate than the CNN. In this paper, we model this decision process by a rule-based system and then by an ABA framework.

It is worth noting that the problem of classification combination is not new, but the existing literature does not include an argumentation-based approach as in our paper. Common approaches deploy information fusion techniques, notably the Dempster-Shafer theory (DST), as done in [25, 26, 27, 28, 29, 30, 31]. The basic idea here is to view the output of a classification system as a DST mass function (*aka* a basic probability assignment) and use different DST combination rules to combine the outputs of a classification ensemble. One of the early known works in this direction was conducted by Xu et al. [26]. But these common approaches do not include an argumentation-based approach as in our paper. More recent works using DST include information fusion [32], a combination of SVM and Bayesian density model [33], and hybrid approaches [34, 35, 36, 37].

The remaining part of this paper is structured as follows. Section 2 discusses existing approaches to SA. Section 3 recalls the formal theories used in this paper. In Section 4, we describe our hybrid method applied to sentiment analysis (SA). Section 5 provides technical details and comparisons. We discuss in Section 6 and conclude in Section 7.

2 Existing approaches to Sentiment Analysis

Current approaches to SA make a distinction between three levels: document level [7, 8, 9, 10, 13, 38], sentence level [11, 12, 39, 40, 41], and aspect level [5, 6, 42]. They are based on lexicons [10, 12, 43, 44, 45, 46, 47], ML [5, 48, 49, 50], hybrid of lexicons and ML [50, 51, 52], and deep learning [18, 53, 54, 55, 56]. In [57], the paper surveyed multiple-word representation models with their power of expression using ML algorithms for NLP-related tasks. For example, in [43], the authors proposed a lexicons-based SA approach for mining food and restaurant reviews written in the Myanmar language. For a recent review of approaches based on DL, readers can refer to studies [18, 58]. In [4, 59, 60, 61, 62, 63, 64, 65, 66], the authors demonstrated that CNN provided better results than the other approaches.

In [59], the authors described an SA approach based on a CNN model, in which the parameter weights of the CNN were initialized for classifying tweets at both the message and phrase levels. Initially, they trained a word2vec model that was refined by the CNN model on a distantly supervised corpus. Finally, the pre-trained parameters from the word2vec model were used to initialize the CNN model, which was trained on the supervised training corpus from Semeval-2015. In [60], the authors proposed a deep CNN model that performed character- to sentence-level sentiment analysis of short texts. The network involved two convolutional layers to extract relevant features from words and sentences. The network was evaluated using the Stanford Sentiment Treebank (*i.e.* movie reviews) and the Stanford Twitter Sentiment corpus (*i.e.* Twitter messages). In [61], the author applied two DL models, long short-term memory (LSTM) and dynamic CNN, to classify Thai Twitter data by investigating the effect of word order in Thai tweets. They compared both LSTM and dynamic CNN models to the classical techniques, such as NB and SVM, and MaxEnt, using bag-of-words. They found that LSTM and dynamic CNN outperformed NB and SVM but not MaxEnt. In [62], the paper described the experimental results of four kinds of CNNs, in which each CNN was implemented on top of word2vec for sentence-level sentiment classification on seven datasets. In [63], the authors designed and experimented with CNNs, which had consecutive convolutional layers for classifying long and complex texts, with three datasets. They showed that using consecutive convolutional layers provided better performance for longer texts. In [64], the authors proposed a classifier based on 2-layer CNN for classifying Italian Twitter messages. They trained CNN in the form of multi-tasking. In [65], a CNN model was trained on the top of a word embedding model, fastText, to perform SA. Three movies reviews datasets were applied in the experiments. The results of the CNN model with fastText were compared with ML techniques and the other CNN model with word2vec. In [66], the authors designed a classifier for SA using multiple CNNs with different configurations of hyper-parameters. The hyper-parameters included types of word embedding, activation functions, filter sizes that were used in convolution, the number of feature maps, and the pooling methods that were used for data reduction. They analyzed the performance of the classifier based on these different configurations. In [67], the authors proposed a hybrid approach, called NOD-CC, that combined CNN with CBR for discovering and classifying object types in images. The hybrid approach outperformed CNN when the training dataset had insufficient data and CNN had low confidence in its prediction. Hence, the level of confidence was measured by comparing it with a threshold. The final image classification from CNN or CBR was determined by an algorithm (called controller) according to their hypotheses when CNN incorrectly classified object type and the queried image did not appear in the training dataset at run time. In [68], to address the impact of higher-level abstraction avoidance on sentiment feature learning in texts and sentiment classification performance, the authors developed a model called AEC-LSTM that built an LSTM network combining with emotional intelligence (EI) and attention mechanisms. An emotion-enhanced LSTM, called ELSTM, was developed using EI to enhance the ability of LSTM networks in features learning. In [69], the author developed a SAKG-BERT model combining sentiment analysis knowledge with the Bidirectional Encoder Representations from Transformers (BERT) language representation model to provide the interpretability to the deep learning algorithm. In [70], the paper focused on the combination of rule-based reasoning and CBR to be used for sentiment analysis.

Meanwhile, argumentation provided successful results in improving the performance of classification tasks using ML techniques [71, 72, 73]. In [72], the paper surveyed the existing seven approaches that improved the performance of ML techniques using argumentation. These approaches differed in the use of argumentation with its semantics and ML techniques. The paper provided a comparative analysis of these approaches. In [73], the authors proposed a methodology for mining bipolar argumentation frameworks (BAFs) from natural language texts, which were treated as arguments by defining attack and support relations between them. They applied ML classifiers to determine relations between arguments and illustrated their methodology on a dataset of hotel reviews. In [74], the paper presented an argument mining framework that automatically detected argumentative sentences and argument components using CNN. The framework was applied to both a specific domain (*e.g.* essays, web comments) and cross-domain data. The performance of the framework outperformed traditional ML techniques such as NB and SVM. However, the framework did not solve the conflicts between the sentences through argumentation.

In recent years, argumentation has become popular in sentiment analysis [75, 76, 77, 78, 79] to improve the performance of SA in classification problems. In [76], the authors developed a classification methodology, called classification enhanced with Arguments (CleAr) that combined argumentation and supervised learning. CleAr applied cross-domain sentiment polarity (positive/negative) classification and relation-based argumentation mining to improve the performance of sentiment classification. In classification, classifiers were trained from one corpus (Tweets) and predicted another corpus (movie reviews). They argued that class labels, which are resulted from the trained classifiers, were able to correct misclassifications through argumentation. In argumentation, sentences were taken as arguments to determine whether they attacked or supported an argument or neither attacked nor supported it. In [77], the author presented a framework to mine opinions from Twitter-based given queries. Twitter-based arguments for the queries were generated from the tweets, which were collected relating to the query. Using SA tools, a sentiment was defined for each argument. Given a query, they built an opinion tree with sentiments (positive, negative, and neutral), whose root node was the query. For defining attack relations between arguments, conflict trees were generated if conflict elements with different sentiments were contained in the opinion tree. In [79], the authors proposed a deep learning method based on the LSTM model to construct bipolar argumentation frameworks and detect deceptive reviews. The LSTM model was used to determine reviews as argumentative relations (*i.e.* attack, support, and neither attack nor support) between reviews. This method outperformed standard supervised classifiers on small datasets by integrating deep learning with argumentative reasoning. The method also improved performance varying from 1 percent to 3 percent compared with the results without argumentative features. In [80], the authors presented an architecture (so-called ANNA) that combined ANNs and the abstract argumentation (AA) framework for effective prediction with explanation, dialectically and logically. Autoencoder and ANN-based feature selection methods were used to select the highest-ranked features from the training examples where these features were coherent[1]. Then, a case base was created using the coherent features together with their outcomes, and the case base was structured by an AA driven and case-based reasoning (AA-CBR) inspired model using the AA framework. With the help of AA's semantics, AA-CBR predicted new cases through argumentation.

Moreover, several approaches have been investigated to provide solutions to AI challenges to allow XAI to become interpretable [23, 81, 82, 83, 84, 85]. In [81], the paper surveyed multiple ANN-CBR twin-systems that integrate ANN and CBR to analyze a solution to the (XAI) problem while maintaining ANN accuracy and CBR interpretability, using post-hoc explanation-by-examples. The paper defined a future direction for the XAI solution (especially from CBR). In [23], the paper surveyed several argumentation-based XAI approaches, including intrinsic and post-hoc explanations using different argumentation frameworks. It also discussed the future directions of the XAI problems. In [82], the authors developed a deep neural network architecture that includes an autoencoder and a special prototype layer to explain its predictions. They merged the network with CBR to ensure that the network was accurate and interpretable. In [83], the authors described an interpretable framework for machine learning-based mammography that includes a CBR-based interpretable neural network algorithm. Although only a small dataset of images is used,

[1] There is no same cases having different outcomes.

Table 1: Summary of related works on argumentative XAI and hybrid models in SA

Reference #	Proposed	Result
[73]	The paper proposed a methodology for producing BAFs by establishing attack and support relations between arguments (*i.e.* texts) regarding a specific topic.	The root note in the BAFs graph was a widely accepted argument, and as applied to a dataset of hotel reviews, the methodology determined that the hotel rooms were either nice or bad.
[76]	The paper proposed the CleAr methodology for classifying tweets that combined argumentation with supervised classifiers such as SVM and NB.	The baselines' F1 measure is improved by a maximum of 0.07 points using CleArs with 84.3% accuracy for the Discontinuity-Free Quantitative Argumentation Debate.
[77]	A twitter-based argumentation framework for SA using incrementally generated queries from a set of tweets was proposed in the paper, constructing opinion trees and conflict trees.	To demonstrate how the framework might be applied in practice, the paper included a user case.
[79]	The paper proposed a method for extracting BAFs from tweets and detecting whether news articles supported tweets using BiLSTMs.	When combined with traditional supervised classifiers, the argumentative feature outperformed these classifiers on small data sets, such as the hotel data set, with increases ranging from 1% to 3% and 76.38% accuracy.
[80]	The paper proposed an architecture that combined ANNs for feature selection and an instance of AA and CBR for justifying the predictions in order to produce accurate and explainable predictions.	Having 96.2% accuracy, the architecture outperformed the two ANN techniques with improvements of 10% in F1 when utilizing a size 30 hidden layer.
[82]	In order for a deep neural network to explain its own reasoning for each prediction based on similar cases, the paper proposed a network architecture using an autoencoder and a decoder.	On the standard MNIST test set, which included grayscale images of handwritten digits, it achieved 99.22% accuracy.
[83]	The paper proposed a framework that used CBR in order to understand and explain mammogram predictions made by the deep learning network ProtoNet.	1136 digital screening mammograms were used in the experiment, and the overall accuracy was 83%.

data with whole image labeling is combined with data with pixel-wise annotations, resulting in improved accuracy and interpretability. In [84], giving end-to-end solutions to DL challenges, the paper presented two perspectives on concerns to be addressed to improve deep learning and AI by solving DL challenges via integration and applying CBR with deep network components.

Having higher accuracy, DL models perform well in classification and prediction, becoming state-of-the-art models. However, they are viewed as black-boxes without any clues on how predictions are made so. In this study, CNN is used to obtain high accuracy for SA. However, it does not provide interpretability. To address this problem, we use post-hoc explanation-by-examples to justify CNN outputs.

Table 1 summarizes related approaches[2]. These approaches focused on argumentative XAI, which combined ML/DL techniques with argumentation, and hybrid models in SA, which combined black-box models with interpretable models. Among these approaches, the method in [80] is the most similar to our study, providing not only classification but also explanation. In contrast to [80], ANNs behaved feature selection but not classification, while the AA-CBR model performed classification based on a case base. In our work, we study the combination of CNN and CBR through an assumption-based argumentation (ABA) framework. Both the CNN and the CBR models predict the outcomes of new inputs (*i.e.* sentences) in their own classification ways. However, CNN and CBR models may produce conflicting outcomes for the same input. We solve this conflict via a system of strict rules and argumentation by considering not only the probability distribution on the set of all outcomes from the CNN model but also different similar sentences from the CBR model (using cosine similarity). Thus, our approach efficiently and quickly interprets CNN outputs in sentence sentiment classification.

3 Theoretical backgrounds

This section briefly explains the main theoretical backgrounds on which our proposal is based: convolutional neural network (CNN), case-based reasoning (CBR), abstract argumentation (AA) and assumption-based argumentation (ABA).

[2] More related SA approaches are summarized in Table 14 (Appendix C).

3.1 Convolutional neural network

Convolutional neural network (CNN), a kind of neural network, has been very successful in image recognition and classification problems. Following Kim [62], many authors have explored CNN in text classification achieving great results. In the following we briefly recall CNN operations, then we discuss "some customization or specific points" that are specific to our proposal.

The main operations of CNN are convolution, non-linearity (*i.e.* activation), pooling, and classification. In the text classification, input to CNN is an $n \times d$ matrix where n is the length of the sentence, d is the dimension and each row of the matrix is a word vector $(1 \times d)$ produced by a word embedding model (such as word2vec). In the convolutional layer, the input matrix is modified by filters[3] with stride[4]. Convolution extracts local features and produces convolved features matrix (*i.e.* output feature map) by using an activation function. In general, the convolution creates $(n-i(h-1)) \times (n-i(h-1))$ matrix as an output feature map from the input matrix by applying $h \times h$ filter and i stride. In the pooling[5] layer, the max pooling selects the maximum value from features of the convolved feature map. When multiple filters with different region sizes are applied, the max pooling features from different filters are concatenated. In the fully connected layer, each feature from the previous layer fully connects to all features of the current layer. Finally, CNN classifies these features using either softmax function for multiple classes or sigmoid function for binary classes. Moreover, CNN applies dropout (between 0 and 1) to reduce overfitting for regularization. Readers can refer to [15] for details.

3.2 Case-based reasoning

Case-based reasoning (CBR) is a methodology for solving problems based on past cases about similar problems [86]. CBR has been applied to solve different kinds of decision-making problems, such as classification, diagnosis, prediction, planning, and configuration. The general architecture of CBR is depicted in Figure 2 which is adapted from [87].

When a new problem is to be solved, CBR retrieves similar cases from the case base based on similarity measures. In the simplest scenario, the solution of the most similar case is reused for the current case. Popular similarity measures include cosine similarity, Euclidean distance and *etc.* The solution is evaluated with the problem, and if necessary, then it is revised. After solving the problem, CBR updates the case base by retaining the new case and its solution for future problems to be solved.

3.3 Abstract argumentation

The study of argumentation in AI is inspired by humans' ability to reach conclusion via exchange of arguments. Recent developments in the field are greatly influenced by Dung's AA framework [88].

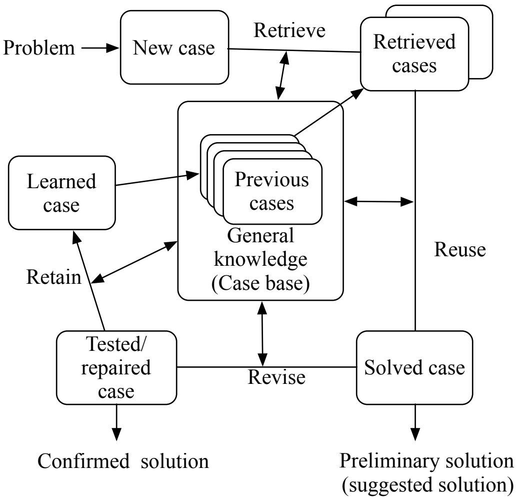

Figure 2: Typical CBR cycle adapted from [87]

Definition 1. Abstract argumentation (AA) [88] framework is a pair, $\mathcal{AF} = (Ar, Att)$ where Ar is a set of arguments and $Att \subseteq Ar \times Ar$ is a set of attacks.

An AA framework can be visualized as a directed graph, as demonstrated by the following example.

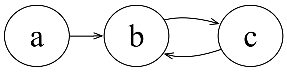

Figure 3: An abstract argumentation framework

Example 1. In the AA $\mathcal{AF}$ depicted in Figure 3, it contains a set of arguments $Ar = \{a, b, c\}$ and a set of attacks $Att = \{(a, b), (b, c), (c, b)\}$.

AA has several semantics specifying when an argument is acceptable. Given an AA framework $\mathcal{AF} = (Ar, Att)$, a set of arguments $S \subseteq Ar$ is conflict-free if each argument of S does not attack itself and other arguments of S. Argument A is acceptable with respect to S if S attacks any arguments attacking A. S is admissible if it is conflict-free and each argument of S is acceptable with respect to S. S is a complete extension if it is admissible in $\mathcal{AF}$ and contains every argument acceptable with respect to S. S is a preferred extension of $\mathcal{AF}$ if it is a maximal (with respect to set inclusion) complete extension. Let $f(S) = \{A \in Ar | A$ is acceptable with respect to $S\}$ be a characteristic function. S is a grounded extension if it is the least fix-point of the characteristic function $f(S)$. Argument A is credulously (*resp.* groundedly) accepted with respect to $\mathcal{AF}$ (*i.e.* $\mathcal{AF} \vdash_{x \in \{cr,gr\}} A$) if it is contained in a preferred (*resp.* grounded) extension.

Example 2. (Continue Example 1.) $\emptyset, \{a\}, \{b\}, \{c\}$, and $\{a, c\}$ are conflict-free. $\emptyset, \{a\}, \{c\}$ and $\{a, c\}$ are admissible. $\{a\}, \{c\}$ and $\{a, c\}$ are complete extensions. $\{a, c\}$ is a preferred and grounded extension.

[3] Filter is a matrix as known as kernel.

[4] Stride is the amount for shifting the filter across the matrix.

[5] Pooling methods are max, min, average, and sum.

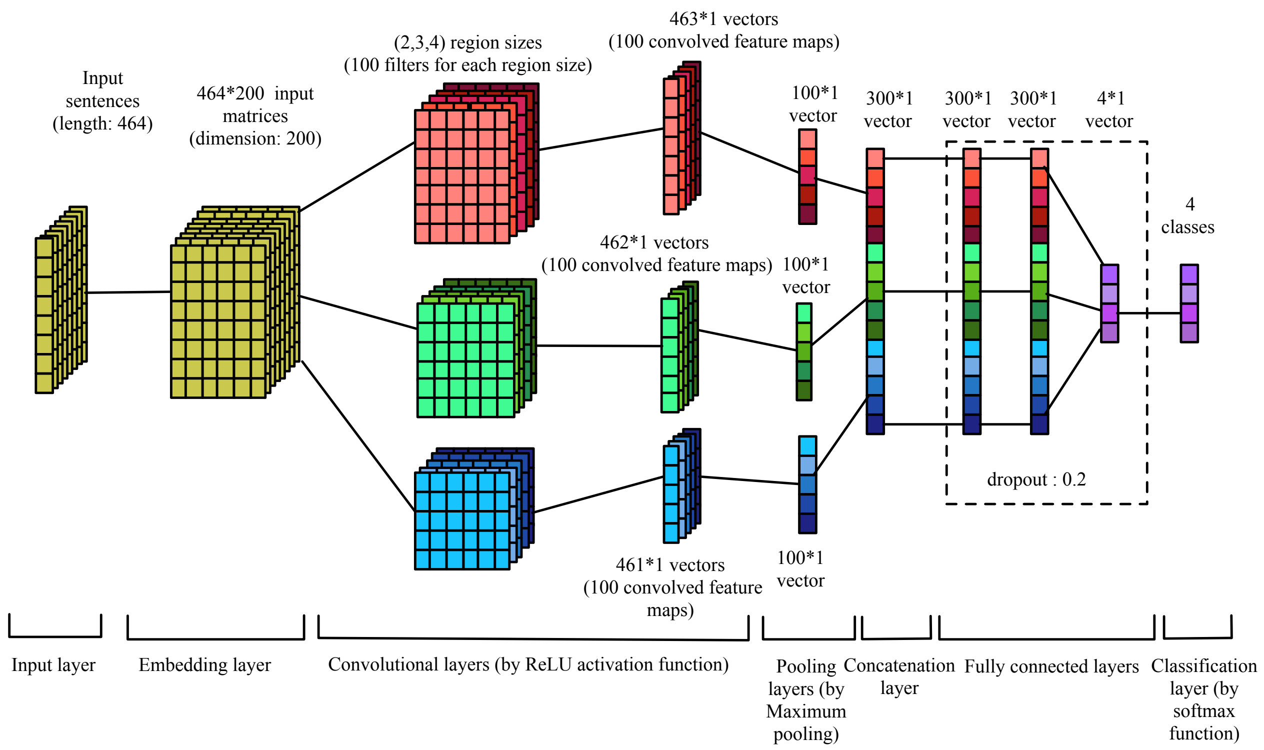
Input sentences (length: 464)
464*200 input matrices (dimension: 200)
(2,3,4) region sizes (100 filters for each region size)
463*1 vectors (100 convolved feature maps)
462*1 vectors (100 convolved feature maps)
461*1 vectors (100 convolved feature maps)
100*1 vector
100*1 vector
100*1 vector
300*1 vector
300*1 vector
300*1 vector
4*1 vector
4 classes
dropout : 0.2
Input layer
Embedding layer
Convolutional layers (by ReLU activation function)
Pooling layers (by Maximum pooling)
Concatenation layer
Fully connected layers
Classification layer (by softmax function)

Figure 4: CNN architecture of the proposed approach

3.4 Assumption-based argumentation

Assumption-based argumentation [24] is an instance of AA that constructs arguments from inference rules supported by assumptions.

Definition 2. An ABA framework is a triple $\mathcal{F} = (\mathcal{A}, \mathcal{R}, \overline{})$ where $\mathcal{A}$ is a set of assumptions, $\mathcal{R}$ is a set of inference rules of the form $\alpha_0 \leftarrow \alpha_1, \ldots, \alpha_n$ ($n \geq 0$) and $\overline{}$[6] is a total mapping from each assumption to its contrary.

We shall refer to an inference rule with some assumptions in its body as a defeasible rule; and a strict rule otherwise. A strict rule of the form $\alpha \leftarrow$ is called a fact. An argument for a proposition π supported by a set of assumptions Q, denoted (Q, π) is basically a proof tree with π labeling the root, Q is the set of labels of leave nodes, and for each internal node labeled by α, there is such an inference rule $\alpha \leftarrow \alpha_1, \ldots, \alpha_n$ that the children of the nodes are labeled by $\alpha_1, \ldots, \alpha_n$. An argument (Q, π) attacks another argument (Q', π') if π is the contrary of some assumption in Q'. The semantics of ABA $\mathcal{F}$ is defined by the semantics of the $\mathcal{AF}$ consisting of the above defined arguments and attacks. A proposition π credulously/groundedly accepted in ABA $\mathcal{F}$, denoted $\mathcal{F} \vdash_x \pi$ ($x \in \{cr, gr\}$), if there is an argument for π that is credulously/groundedly in the corresponding $\mathcal{AF}$.

4 Proposed approach

In this section, we present our two-step hybrid approach that combines a CNN model and a CB model using argumentation as shown in Figure 1. The dataset $\mathcal{D} = \{(s_i, l_i)\}_1^n$ consists of a set of sentences $\mathcal{S} = \{s_1, \ldots, s_n\}$ (n=25104 in our case) where each sentence s_i is associated with a sentiment label $l_i \in \mathcal{L} = \{+1, -1, +0, -0\}$. The interpretation of these labels is as follows: +1 =“positive”, −1 =“negative”, +0 =“neutral”, and −0 =“unrelated”. We divide $\mathcal{D}$ into three disjoint subsets $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3$ with sizes of 15413, 4510, and 5181, respectively. $\mathcal{D}_1$ is used to train the CNN model; $\mathcal{D}_2$ is used to build the CBR model, and $\mathcal{D}_3$ is used to test how well-built models perform. Now let us describe each model in detail.

4.1 The CNN model ($\mathcal{M}_1$)

As we mentioned in Section 3.1, we use some specific architecture of CNN for our $\mathcal{M}_1$ model. This section describes 1) our process for producing the model and 2) the performance of the model on the test dataset $\mathcal{D}_3$. Our CNN model is built on top of a word embedding model (*i.e.* word2vec[7]) as shown in Figure 4.

The model contains an input layer, an embedding layer, three convolution layers, three maximum pooling layers, a concatenate layer, two dense layers, a dropout layer, and a classification layer. The maximum length of all input sentences is 464 words and the dimension of the word vector is 200[8]. Then a 464 × 200 matrix for each input sentence is

[6] $\overline{x}$ is the contrary of the assumption x.

[7] Word2vec represents each word as a row vector, having similar vector representations for words with similar meanings, whereas doc2vec is an extension of word2vec for evaluating document relationships (not only words). Instead of word2vec, doc2vec might be used. However, because we focus on sentence sentiment classification, we train the CNN model on word2vec.

[8] Based on the dataset we used, we set the vector space of word2vec to

taken as an input into the CNN model. We apply $(2, 3, 4)$ different region sizes in three convolutional layers respectively, using the $ReLU$ activation function to generate convolved features maps from all filters. 100 filters are used for each region size so there are a total of 300 filters. Every convolved features map is globally maximized in each pooling layer, which produces a 100×1 vector. The vectors from all pooling layers are connected in the concatenation layer. The two dense layers and dropout layer work as fully connected layers. Then 0.2 is assigned as the rate of dropout for reducing network overfitting. The second dense layer works as regularization from outputs of the dropout layer. Finally, the $softmax$ function is applied in the classification layer to classify the input sentences into their respective sentiments.

For accuracy assessment, Table 2 shows the confusion matrix of $\mathcal{M}_1$. The CNN model achieves 87.59% accuracy and it misclassifies 643 out of 5,181 sentences. As $\mathcal{M}_1$ shall be used as a black-box component of the hybrid model developed later on, we need to establish some notations regarding the usage of $\mathcal{M}_1$. By $P_{\mathcal{M}_1}(l|n)$, we mean the probabilistic value that $\mathcal{M}_1$ produces for a given input n. By $\mathcal{M}_1(n) = l$, we mean $l = argmax_{l \in \mathcal{L}} P_{\mathcal{M}_1}(l|n)$.

Table 2: Confusion matrix of $\mathcal{M}_1$ (sentiments are denoted as follows: POS stands for positive; NEG for negative; NEU for neutral; URE for unrelated)

$\mathcal{M}_1$ \ $\mathcal{D}_3$	POS	NEG	NEU	URE	Total
POS	311	18	41	6	376
NEG	73	1,970	128	36	2,207
NEU	17	70	646	45	778
URE	10	46	153	1611	1,820
Total	411	2,104	968	1,698	5,181

4.2 The CBR model ($\mathcal{M}_2$)

As we mentioned in Section 3.2, the model $\mathcal{M}_2$ is based on CBR. This section describes 1) our process to produce the model and 2) the performance of the model on the test dataset $\mathcal{D}_3$. The model computes the similarity between new and existing sentences using cosine similarity. The model then predicts a possible sentiment for the new sentence regarding sentiments of similar sentences.

Example 3. Customer reviews written in Myanmar language are collected from a telecommunication company's Facebook page and annotated with their respective sentiments. These annotated sentences are then kept in the dataset $\mathcal{D}$. From the dataset, three sentences are extracted and translated into the following English sentences: s_1 and s_2 as existing sentences, and n_1 as a new sentence.

s_1 : I recommend a monthly unlimited internet package with a reasonable price and basic speed.

s_2 : The speed of low-price monthly unlimited internet package is too slow.

n_1 : The price of monthly unlimited internet package is reasonable but the speed is weak for HD streams.

Table 3 shows similar sentences with their respective similarity values.

Table 3: Similarities between sentences

New sentence	Sentence in case base	Similarity
n_1	s_1	0.97
n_1	s_2	0.96

Example 4. (Continue Ex.3.) For the sentence n_1, s_1 and s_2 are the first two most similar sentences and annotated as positive sentiment (+1) and negative sentiment (-1) respectively. On the basis of the closest similarity between sentences, the model predicts n_1 as positive sentiment (+1).

For accuracy assessment, Table 4 shows the confusion matrix of $\mathcal{M}_2$. The CBR model achieves 71.57% accuracy and it misclassifies 1,473 out of 5,181 sentences. As $\mathcal{M}_2$ shall be used as a component of the hybrid model developed later on, we need to establish some notations regarding the usage of $\mathcal{M}_2$. $sim(s, n)$ denotes the probabilistic value (between 0 and 1) that $\mathcal{M}_2$ produces for given input n and s, where n stands for new sentence and s stands for the existing sentence in the case database that matches the most with the sentence n. The fact that $\mathcal{M}_2(n) = \mathcal{D}_2(s)$ where $s = argmax_{m \in \mathcal{D}_2}(sim(m, n))$ means $\mathcal{M}_2$ labels sentiment of n as the same sentiment of s.

Table 4: Confusion matrix of $\mathcal{M}_2$

$\mathcal{M}_2$ \ $\mathcal{D}_3$	POS	NEG	NEU	URE	Total
POS	365	88	63	177	693
NEG	24	1780	228	301	2,333
NEU	14	207	595	252	1,068
URE	8	29	82	968	1,087
Total	411	2,104	968	1,698	5,181

4.3 The rule-based hybrid model ($\mathcal{M}_3$)

To justify sentiment classification of the CNN model $\mathcal{M}_1$ via the CBR model $\mathcal{M}_2$, we develop a series of different hybrid models combining $\mathcal{M}_1$ and $\mathcal{M}_2$ in different ways. This section presents the first model in the series, called the rule-based hybrid model $\mathcal{M}_3$. We describe the construction of $\mathcal{M}_3$ and then present a logic program implementing $\mathcal{M}_3$. Finally, we report the performance of $\mathcal{M}_3$ using the test dataset $\mathcal{D}_3$.

For a new sentence n, $\mathcal{M}_3$ assigns a label $\mathcal{M}_3^{\alpha,\beta}(n)$ to n according to the following system of equations taking $\alpha, \beta \in [0, 1]$ as parameters.

200-dimensions for a maximum of 2 million Myanmar words because 300-dimensions is a standard vector space for Google News having 3 million words.

$$\mathcal{M}_3^{\alpha,\beta}(n) = \begin{cases} \mathcal{M}_1(n) & \text{if } \mathcal{M}_1(n) = \mathcal{M}_2(n) \quad (i) \\ \mathcal{M}_2(n) & \begin{array}{l}\text{if condition } C \text{ (defined in 2) and} \\ \text{either condition below holds:} \\ \bullet\ sim(s,n) \geq \beta, \text{ or } \quad (ii.1) \\ \bullet\ \alpha < sim(s,n) < \beta \text{ and} \\ \quad * \ \mathcal{M}_1(n) = -1 \text{ and} \\ \quad\ \ \mathcal{M}_2(n) = +1, \text{ or } \quad (ii.2) \\ \quad * \ \mathcal{M}_1(n) \neq -1 \text{ and} \\ \quad\ \ \mathcal{M}_2(n) = +0 \quad (ii.3)\end{array} \\ \mathcal{M}_1(n) & \text{otherwise} \quad (iii) \end{cases} \tag{1}$$

Let us recall the notations: n refers to a new sentence to be labeled; s refers to the most similar sentence of n in the dataset $\mathcal{D}_2$, that is $s = argmax_{m \in \mathcal{D}_2} sim(m,n)$ where $sim(m,n)$ measures the similarity between m and n (we use the cosine similarity); $\mathcal{M}_1(n)$ refers to the sentiment that $\mathcal{M}_1$ assigns to n; $\mathcal{M}_2(n)$ refers to the sentiment that $\mathcal{M}_2$ assigns to n (*i.e.* $\mathcal{M}_2(n) = \mathcal{D}_2(s)$); $P_{\mathcal{M}_1}(\_ \mid \_)$ refers to the probability distribution on the set of labels that $\mathcal{M}_1$ produces condition to the input sentence. Different values of numeric parameters α, β give rise to different versions of $\mathcal{M}_3$, where attention needs not to be paid to these parameters, we may write $\mathcal{M}_3^{\alpha,\beta}(n)$ for simplicity.

Now let us elaborate on different cases of the above system in Equation (1)[9].

- **Case i**: $\mathcal{M}_1$ and $\mathcal{M}_2$ assign the same sentiment to n and hence $\mathcal{M}_3^{\alpha,\beta}$ should do the same thing (*i.e.* $\mathcal{M}_3^{\alpha,\beta}(n) = \mathcal{M}_1(n) = \mathcal{M}_2(n)$).
- **Case ii**: Basically $\mathcal{M}_1$ and $\mathcal{M}_2$ give conflicting labels and $\mathcal{M}_3^{\alpha,\beta}$ follows $\mathcal{M}_2$. There are three sub-cases ii.1, ii.2, ii.3 detailed in the following which share a common condition C defined as follows.

$$C = \begin{array}{l} \mathcal{M}_1(n) \neq \mathcal{M}_2(n) \text{ and} \\ \mathcal{M}_1(s) = \mathcal{M}_2(n) \text{ and} \\ P_{\mathcal{M}_1}(l|n) < P_{\mathcal{M}_1}(l'|s) \\ \text{where } l = \mathcal{M}_1(n), l' = \mathcal{M}_1(s) \end{array} \tag{2}$$

The first part $\mathcal{M}_1(s) = \mathcal{M}_2(n)$ of C says that $\mathcal{M}_1$ would agree with $\mathcal{M}_2$ if $\mathcal{M}_1$ is given s as input. The second part $P_{\mathcal{M}_1}(l|n) < P_{\mathcal{M}_1}(l'|s)$ says that $\mathcal{M}_1$ is more assertive in its label assignment for s than for n. Besides C, case ii.1-3 each requires a specific condition:

  - **Case ii.1**: $sim(s,n) \geq \beta$ says that s and n are similar enough and hence $\mathcal{M}_2$ should be confident in transferring the observed label of s to n.
  - **Case ii.2** and **Case ii.3**: $\alpha < sim(s,n) < \beta$ says that n and s are similar but probably not close enough and hence $\mathcal{M}_2$ should not be too conclusive in its decision $\mathcal{M}_2(n) = \mathcal{D}_2(s)$. Hence to call $\mathcal{M}_3^{\alpha,\beta}$ to follow $\mathcal{M}_2$, we should introduce an extra condition: $\mathcal{M}_1(n) = -1$ and $\mathcal{M}_2(n) = +1$ (condition ii.2); or $\mathcal{M}_1(n) \neq -1$ and $\mathcal{M}_2(n) = +0$ (condition ii.3). Here condition ii.2 says that $\mathcal{M}_1$ assigns negative sentiment to n and $\mathcal{M}_2$ assigns positive sentiment to n while condition ii.3 says that the label $\mathcal{M}_1$ assigns to n is not negative sentiment and the label $\mathcal{M}_2$ assigns to n is neutral sentiment.

- **Case iii**: Otherwise, $\mathcal{M}_3^{\alpha,\beta}$ assigns the same sentiment that $\mathcal{M}_1$ assigns to n (*i.e.* $\mathcal{M}_3^{\alpha,\beta}(n) = \mathcal{M}_1(n)$).

It is worth noting that $\alpha < \beta$, which should be close to 1. By changing the values of α and β, we can control the range of case ii and case iii. In particular, if $\alpha \sim \beta \sim 1$, case ii will never fire and hence $\mathcal{M}_3$ will degenerate to the CNN model $\mathcal{M}_1$. On the other hand, if $\alpha = 0$, case iii will never fire, and hence $\mathcal{M}_3$ will depart from $\mathcal{M}_1$ to the greatest extent. We report the performance of $\mathcal{M}_3^{0.8,0.96}$ (*i.e.* $\alpha = 0.8, \beta = 0.96$).

Now let us switch our attention to the implementation of $\mathcal{M}_3$ by structured argumentation. No conditions of the above three cases contain an exception, and hence the whole system of equations (1) can be implemented by a set of strict inference rules $\mathcal{R}_1$ which can be wrapped by an ABA framework $\mathcal{F}_1 = (\mathcal{A}_1, \mathcal{R}_1, \overline{})$ with $\mathcal{A}_1 = \emptyset$. The rules in $\mathcal{R}_1$ deploy several self-describing predicates: $m_1(N,L)$ means that the model $\mathcal{M}_1$ assigns label L to input sentence N; $m_2(N,S,L,R)$ says that $\mathcal{M}_2$ finds a sentence S with label L to be R^{th}-most similar sentence to the input sentence N[10] (for convenience, let $m_2(N,S,L)$ stand for $m_2(N,S,L,1)$); for simplicity let $p(N,L,X)$ stand for $P_{\mathcal{M}_1}(L|N) = X$; $sim(N,S,Z)$ says that Z is the similarity value between sentences N and S. Appendix A presents all rules of $\mathcal{R}_1 = \{r_1, r_2, .., r_7\}$, for example

$$\begin{array}{lrl} r_1: & case(i,N,L) \leftarrow & m_1(N,L), m_2(N,S,L). \\ r_7: & m_3(N,L) \leftarrow & case(i,N,L); \\ & & case_{\mathcal{M}_3^{\alpha,\beta}}(ii,N,L); \\ & & case(iii,N,L). \end{array}$$

Note that predicate $case(\_,N,L)$ says that $\mathcal{M}_3$ assigns label L to N according to the specific cases mentioned in the above. For case ii, we add subscript $\mathcal{M}_3^{\alpha,\beta}$ to differentiate it with $case_{\mathcal{M}_4^{\alpha,\beta}}(ii,N,L)$ of model $\mathcal{M}_4$ developed in the next section. Note that in case i and iii, $\mathcal{M}_3^{\alpha,\beta}$ and $\mathcal{M}_4$ behave exactly the same.

Given an input sentence N described by a set of facts $\mathcal{R}_N$, the label that $\mathcal{M}_3$ assigns to N is computed by ABA framework $(\mathcal{A}_1, \mathcal{R}_1 \cup \mathcal{R}_N, \overline{})$ obtained from the above ABA framework $\mathcal{F}_1$ by adding $\mathcal{R}_N$ into the set of inference rules.

Example 5. Consider a sample input sentence n_1 where $\mathcal{R}_{n_1} = \{r_8, ..., r_{13}\}$ consists of:

$$\begin{array}{ll} r_8: m_1(n_1,+0) \leftarrow & r_9: m_2(n_1,s_1,+1) \leftarrow \\ r_{10}: m_1(s_1,+1) \leftarrow & r_{11}: p(n_1,+0,0.55) \leftarrow \\ r_{12}: p(s_1,+1,0.9) \leftarrow & r_{13}: sim(n_1,s_1,0.97) \leftarrow \end{array}$$

[9] The algorithmic form of $\mathcal{M}_3^{0.8,0.96}$ can be found in Appendix B.

[10] $m_2(N,S,L,R)$ can be defined by a rule below

$$\begin{array}{ll} m_2(N,S,L,R) \leftarrow & S = arg\_rank_{S_1 \in \mathcal{D}_2}(R, sim(S_1,N)), \\ & L = \mathcal{D}_2(S). \end{array}$$

where $arg\_rank_{S_1 \in \mathcal{D}_2}(R, sim(S_1,N))$ means that S is the R^{th}-most similar sentence to N.

$\mathcal{R}_{n_1}$ says that s_1 is the most similar sentence. Facts r_8 and r_9 say that $\mathcal{M}_1$ and $\mathcal{M}_2$ assign conflict labels to n_1. Hence rules r_2 and r_5 fire (refer to Appendix A), resulting in the grounded acceptance of $m_3(n_1, +1)$ in the ABA $(\mathcal{A}_1, \mathcal{R}_1 \cup \mathcal{R}_{n_1}, \overline{})$.

Table 5: Confusion matrix of $\mathcal{M}_3^{0.8,0.96}$

$\mathcal{M}_3$ \ $\mathcal{D}_3$	POS	NEG	NEU	URE	Total
POS	357	33	39	6	435
NEG	30	1,958	123	35	2,146
NEU	14	69	656	52	791
URE	10	44	150	1,605	1,809
Total	411	2,104	968	1,698	5,181

For the accuracy assessment, Table 5 shows the confusion matrix of $\mathcal{M}_3^{0.8,0.96}$ (*i.e.* $\alpha = 0.8, \beta = 0.96$). With respect to the test dataset $\mathcal{D}_3$, $\mathcal{M}_3^{0.8,0.96}$ achieves 88.32% accuracy and it misclassifies 605 out of 5,181 sentences. The table says that $\mathcal{M}_3^{0.8,0.96}$ performs a bit better than the CNN model $\mathcal{M}_1$ (0.73%) and the CBR model $\mathcal{M}_2$ (16.75%). Table 6 shows the number of sentences labeled by $\mathcal{M}_3^{0.8,0.96}$ per case. We observe that 3624/5181 of tested sentences fall into case i, suggesting that around 70% of the outputs of $\mathcal{M}_3$ are explained by the sentiments of similar sentences. Table 7 shows each case's accuracy. Here we observe that 93.87% of case i is the most accurate, meaning that in this case $\mathcal{M}_3$ is not only interpretable but also highly accurate.

Table 6: Numbers of sentences of $\mathcal{M}_3^{0.8,0.96}$ per case

Sentiments	Cases		
	case i	case ii	case iii
POS	315	73	47
NEG	1,769	13	364
NEU	515	21	255
URE	1,025	0	784
Total	3,624	107	1,450

Table 7: Accuracy of $\mathcal{M}_3^{0.8,0.96}$ per case

	Cases		
	case i	case ii	case iii
No. of correctly labeled sentences	3,402	67	1,103
No. of sentences	3,624	107	1,450
Accuracy	93.87%	62.61%	76.07%

4.4 The argumentation-based hybrid model ($\mathcal{M}_4$)

In this section, we revise $\mathcal{M}_3$ to obtain an argumentation-based model called $\mathcal{M}_4$. Basically, $\mathcal{M}_4$ refines case ii of $\mathcal{M}_3$ by considering not only the most similar sentence but also the second most similar sentence, the third most similar sentence, and so on. The idea here is quite obvious: more information can help us to refine the decision rules. For simplicity of presentation, we consider only the second most similar sentence though. $\mathcal{M}_4$ shares the same case i and iii with $\mathcal{M}_3$. If case ii of $\mathcal{M}_3$ occurs then case ii of $\mathcal{M}_4$ occurs and vice versa but two models may assign different labels to the input sentence. Concretely, suppose that $case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L_1)$ holds. Note that here L_1 is the label of the most similar sentence to the input sentence N, *i.e.* $m_2(N, S_1, L_1, 1)$ holds. In the following, L_i refers to the label of the i^{th} most similar sentence to N, *i.e.* $m_2(N, S_2, L_2, 2)$, $m_2(N, S_3, L_3, 3)$, and so on. As will be seen, $\mathcal{M}_4$ may assign a different label to N, *i.e.* $case_{\mathcal{M}_4^{\alpha,\beta}}(ii, N, L)$ holds for some L probably different from L_1. The rules that determine whether L is the same as L_1 (the label that $\mathcal{M}_3$ assigns to the given input sentence N) or a different label (*e.g.* L_2, L_3) are as follows;

a. $\mathcal{M}_4$ assigns $L = L_2$ if $m_1(S_2, L_2) \wedge m_2(N, S_2, L_2, 2)$ and either of the conditions holds:

   - $L_1 = +0$ (*i.e.* neutral sentiment) or $L_1 = -0$ (*i.e.* unrelated sentiment), or
   - $L_1 = +1$ (*i.e.* positive sentiment) and $P_{\mathcal{M}_1}(L_2|S_2)$[11] ≥ 0.98.

b. Otherwise, $\mathcal{M}_4$ follows $\mathcal{M}_3$ to assign L_1 to N.

$\mathcal{M}_4$ is implemented by an ABA framework $(\mathcal{A}_2, \mathcal{R}_2, \overline{})$ obtained from the ABA framework $(\mathcal{A}_1, \mathcal{R}_1, \overline{})$ by the replacing $\mathcal{R}_1$ with $\mathcal{R}_2 = \mathcal{R}_1 \setminus \{r_7\} \cup \{r'_7\} \cup \{r_{14}, ..., r_{16}\}$. Concretely:

- The inference rules r'_7 replaces the inference rule r_7 where

$$\begin{aligned} r'_7 : m_4(N, L) \leftarrow \quad & case(i, N, L); \\ & case_{\mathcal{M}_4^{\alpha,\beta}}(ii, N, L); \\ & case(iii, N, L). \end{aligned}$$

- $\mathcal{R}_2$ also contains the following additional inference rules $r_{14.1}$, $r_{14.2}$, r_{15} and r_{16} where

$$\begin{aligned} r_{14.1} : case_{\mathcal{M}_4^{\alpha,\beta}}(ii_a, N, L) \leftarrow \quad & case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L_1), \\ & m_2(N, S_1, L_1, 1), \\ & m_1(S_2, L_2), \\ & m_2(N, S_2, L_2, 2), \\ & (L_1 = +0; L_1 = -0; \\ & (L_1 = +1, \\ & p(S_2, L_2, X), \\ & X >= 0.98)), L = L_2. \end{aligned}$$

  implementing the condition *a* described in the above rules that determine label assignment of $\mathcal{M}_4$.

$$\begin{aligned} r_{14.2} : case_{\mathcal{M}_4^{\alpha,\beta}}(ii_a, N, L) \leftarrow \quad & case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L_1), \\ & m_2(N, S_1, L_1, 1), \\ & m_1(S_2, L_2), \\ & m_2(N, S_2, L_2, 2), \\ & (L_1 = -1; (L_1 = +1, \\ & p(S_2, L_2, X), \\ & X < 0.98)), L = L_1. \end{aligned}$$

  stating that $\mathcal{M}_4$ assigns label L_1 to the input sentence N if these conditions hold: either $L_1 = -1$ or $L_1 = +1$ and $P_{\mathcal{M}_1}(L_2|S_2) < 0.98$.

$$\begin{aligned} r_{15} : case_{\mathcal{M}_4^{\alpha,\beta}}(ii_b, N, L) \leftarrow \quad & case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L), \\ & \sim case_{\mathcal{M}_4^{\alpha,\beta}}(ii_a, N, \_). \end{aligned}$$

  stating that $\mathcal{M}_4$ follows $\mathcal{M}_3$ as default condition.

[11] Recall that $P_{\mathcal{M}_1}(L_i|S_i)$ means the conditional probability for label L_i according to $\mathcal{M}_1$ given S_i.

$$r_{16} : case_{\mathcal{M}_4^{\alpha,\beta}}(ii, N, L) \leftarrow \quad case_{\mathcal{M}_4^{\alpha,\beta}}(ii_a, N, L); \\ case_{\mathcal{M}_4^{\alpha,\beta}}(ii_b, N, L).$$

stating that one of two conditions occurs assigning label L to input sentence N, then $\mathcal{M}_4$ should follow.

- $\mathcal{A}_2$ contains a set of assumptions $\{\sim case(i, N, L), \sim case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L), \sim case_{\mathcal{M}_4^{\alpha,\beta}}(ii_a, N, L)\}$. The contraries are as follows.
  $\overline{\sim case(i, N, L)} = case(i, N, L)$
  $\overline{\sim case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L)} = case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L)$
  $\overline{\sim case_{\mathcal{M}_4^{\alpha,\beta}}(ii_a, N, L)} = case_{\mathcal{M}_4^{\alpha,\beta}}(ii_a, N, L)$

Given facts of $\mathcal{R}_N$ for input sentence N, the label that $\mathcal{M}_4$ assigns to N is computed by ABA framework $(\mathcal{A}_2, \mathcal{R}_2 \cup \mathcal{R}_N, \overline{})$ obtained from the above ABA framework $\mathcal{F}_2$ by adding $\mathcal{R}_N$ into the set of inference rules.

Example 6. (Continue Ex. 5.) Now the set of facts $\mathcal{R}_{n_1}$ about the input sentence n_1 adds the following facts.

$$r_{17} : m_1(s_2, -1) \leftarrow \qquad r_{18} : m_2(n_1, s_2, -1, 2) \leftarrow$$
$$r_{19} : p(s_2, -1, 0.98) \leftarrow \qquad r_{20} : sim(n_1, s_2, 0.96) \leftarrow$$

According to facts (from r_8 to r_{13}) of Ex.5, rule r_5 fires which concludes $case_{\mathcal{M}_3^{\alpha,\beta}}(ii, n_1, +1)$. By the facts (from r_{17} to r_{20}), rule $r_{14.1}$ fires which concludes $case_{\mathcal{M}_4^{\alpha,\beta}}(ii, n_1, -1)$. When rule r'_7 fires, this ABA groundedly accepts $m_4(n_1, -1)$ so $\mathcal{M}_4$ assigns negative sentiment to n_1.

Table 8: Confusion matrix of $\mathcal{M}_4^{0.8,0.96}$

$\mathcal{M}_4$ \ $\mathcal{D}_3$	POS	NEG	NEU	URE	Total
POS	362	43	41	8	454
NEG	24	1,949	121	37	2,131
NEU	15	68	658	48	789
URE	10	44	148	1,605	1,807
Total	411	2,104	968	1,698	5,181

For accuracy assessment, Table 8 shows the confusion matrix of $\mathcal{M}_4^{0.8,0.96}$ (*i.e.* $\alpha = 0.8, \beta = 0.96$). Using the test dataset $\mathcal{D}_3$, $\mathcal{M}_4^{0.8,0.96}$ achieves 88.28% accuracy and it misclassifies 607 out of 5181 sentences. The table says that $\mathcal{M}_4^{0.8,0.96}$ performs a bit better than the CNN model $\mathcal{M}_1$ (0.69%) and the CBR model $\mathcal{M}_2$ (16.71%).

Table 9: Numbers of sentiments of $\mathcal{M}_4^{0.8,0.96}$ per case

Sentiments	Cases		
	case i	case ii	case iii
POS	342	90	22
NEG	1,967	16	148
NEU	528	19	242
URE	1,025	0	782
Total	3,862	125	1,194

Table 9 shows the number of sentences labeled by $\mathcal{M}_4^{0.8,0.96}$ per case. We observe that 3862/5181 of tested sentences fall into case i, suggesting that around 74% of the outputs of $\mathcal{M}_4$ are explained by the sentiments of similar sentences. Table 10 shows each case's accuracy. Here it concludes that 92.52% of case i is the most accurate.

Table 10: Accuracy of $\mathcal{M}_4^{0.8,0.96}$ per case

	Cases		
	case i	case ii	case iii
No. of correctly labeled sentences	3,573	75	922
No. of sentences	3,862	125	1,194
Accuracy	92.52%	60.00%	77.22%

5 Some Technical Details and Comparisons

In the following, we describe some technical details that are intentionally left out in the previous section. Initially, the whole dataset contains 25,104 sentences (in the Myanmar language) that are customer reviews about products and services of a telecommunication company from Facebook. All the sentences are labeled with one of the following sentiments: positive (+1), negative (−1), neutral (+0), and unrelated (−0). The sentences are identified as unrelated if they are not concerned with the company's products/services, whereas positive, negative, and neutral sentences are concerned with the company's products/services. In the preprocessing stage, the font of these sentences is converted into the Myanmar Unicode for implementation in Python. These sentences are segmented using "—" between single words since spaces are occasionally used in the Myanmar language. Then a word2vec model is built for all the unique words in the dataset for implementing CNN model.

Let us specify how sentences in the dataset are labeled in detail. The labeling process is done in three steps. Firstly, we create three dictionaries: one for positive words, one for negative words, and one for product words. Based on these dictionaries, four conditions apply to the labeling scheme: if a sentence contains product words and negative words, we annotate it with a negative sentiment; if the sentence contains product words and positive words but no negative words, we annotate it with a positive sentiment; if the sentence contains product words but no negative words and no positive words, we annotate it with a neutral sentiment; and otherwise, we annotate the sentence with an unrelated sentiment. Meanwhile, we build a lexicon-based rule-based system (RBS) according to these hypotheses. The system automatically assigns sentiments to all sentences in the dataset. Finally, although the manual labeling process is done by a person, we modify the dataset by verifying the manually labeled sentences with the automatically labeled sentences to obtain the ground truth dataset.

In this paper, we partitioned the entire dataset into 79.36% for training the CNN and CBR models and 20.64% for testing the models. In the CNN model, the larger the training dataset, the higher the model performance for predicting unknown data. A CNN trained from a larger dataset usually obtains a better learning hyper-parameters and then higher performance. A CBR model, on the other hand, handles a new case by reusing successful solutions from previously solved similar problems. Every labeled sentence of a case base has the potential to be utilized as an example for explaining outputs. Hence, the dataset used to implement the CBR model should be clean and coherent (*e.g.* duplicate sentences must be removed). We should allow the CBR model to be implemented with relatively small datasets, due to the high cost of constructing completely clean datasets. Thus, we use three-quarters of the training dataset to train the CNN model and one-quarter of the training dataset to implement the CBR model, rather than dividing the training dataset equally for both models.

The purpose of this study is to use argumentation in the combination of CNN and CBR models. We integrate CNN and CBR to produce hybrid models that are both interpretable and more accurate. Thus, argumentation is required during the combination process to interpret the combination result. For demonstration purposes, we generated one CNN and one CBR using separate datasets. We may build them using the same dataset because it is trivial how input models are created depending on dataset in this paper. In the testing process, however, we use the same dataset to evaluate CNN and CBR. Table 11 shows the data partition for implementing our approach.

Table 11: Data partition

Sentiments	Number of sentences		
	Training		Testing
	CNN	CBR	
POS	421	789	411
NEG	6,894	1,114	2,104
NEU	2,567	1,499	968
URE	5,531	1,108	1,698
Total	15,413	4,510	5,181

The CNN model is built using TensorFlow with some python packages: numpy, keras, pandas, sklearn, tqdm, genism and utils. In our previous work [89], the CNN model was trained using 1,152 annotated sentences and tested using 495 unlabeled sentences. In the current work, the CNN model is implemented with the similar architecture of our previous work. In this paper, the CNN model is trained using 15,413 annotated sentences and tested using 5,181 unlabeled sentences. The CNN model results in 97.61% accuracy in the training process, and 87.59% accuracy in the testing process. Meanwhile, we build the CBR model based on cosine similarity using 4,510 sentences. Using the test dataset, the CBR model achieves 71.57% accuracy. Moreover, the CBR model can explain label assignment to input sentence in form of argument from analogy. Although the CNN model provides better accuracy than the CBR model, it is uninterpretable.

The rule-based hybrid (RBH) model, $\mathcal{M}_3$, combines both CNN and CBR models using the system of equations to determine whether to follow one of them. In the system of equations, there are two variables α and β that can change the range of case ii and case iii. In the implementation of the RBH model, we assign 0.8 to α and 0.96 to β and structure the model by a structured argumentation (using strict rules). Using the test dataset, the model achieves 88.32% accuracy that is a bit more accurate than the CNN model but it can interpret around 70% of their outputs while CNN cannot give any explanation. Then, the argumentation-based hybrid (ABH) model, $\mathcal{M}_4$, revises the RBH model by considering different similarities between input and existing sentences. We structure the ABH model using an ABA framework by adding assumptions to some strict rules of the RBH model. Using the test dataset, the ABH model results in 88.28% accuracy that is also a bit better performance than the CNN model and it can explain around 74% of their outputs (*i.e.* the ABH model provides more explainable outputs than the RBH model). Therefore, our hybrid models do not sacrifice accuracy to achieve interpretability.

Additionally, let us describe another potential solution capable of providing explanations for CNN model outputs based on a rule-based system (RBS). We may also include a lexicon-based RBS (*i.e.* named as $\mathcal{B}_1$ and constructed using positive and negative lexicons) in our second hybrid model to explain the outputs. Let $b(N, L)$ be a lexicon-based predicate stating that the RBS assigns label L to a given input sentence N. Assume that $\mathcal{M}'_4$ is another ABH model implemented using an ABA framework $\mathcal{F}'_2 = (\mathcal{A}'_2, \mathcal{R}'_2, \overline{})$[12] derived from the ABA framework $(\mathcal{A}_1, \mathcal{R}_1, \overline{})$ by the replacing $\mathcal{R}_1$ with $\mathcal{R}'_2 = \mathcal{R}_1 \setminus \{r_7\} \cup \{r'_7\} \cup \{r'_{14.1}, r'_{14.2}, r_{15}, r_{16}\}$ and having the same assumptions and their contraries as $\mathcal{A}_2$. $r'_{14.1}$ and $r'_{14.2}$ are inference rules that replace $r_{14.1}$, and $r_{14.2}$, respectively, where

$$\begin{aligned} r'_{14.1} : case_{\mathcal{M}'_4{}^{\alpha,\beta}}(ii_a, N, L) \leftarrow \quad & case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L_1), \\ & m_2(N, S_1, L_1, 1), \\ & m_1(S_2, L_2), \\ & m_2(N, S_2, L_2, 2), \\ & b(N, L_2), L = L_2. \end{aligned}$$

means that $\mathcal{M}'_4$ assigns label L_2 to the input sentence N if $\mathcal{M}_2$ assigns L_1 and L_2 into N according to S_1 and S_2, respectively, and $\mathcal{B}_1$ assigns L_2 into N.

$$\begin{aligned} r'_{14.2} : case_{\mathcal{M}'_4{}^{\alpha,\beta}}(ii_a, N, L) \leftarrow \quad & case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L_1), \\ & m_2(N, S_1, L_1, 1), \\ & m_1(S_2, L_2), \\ & m_2(N, S_2, L_2, 2), \\ & b(N, L_1), L = L_1. \end{aligned}$$

means that $\mathcal{M}'_4$ assigns label L_1 to the input sentence N if $\mathcal{M}_2$ assigns L_1 and L_2 into N according to S_1 and S_2, respectively, and $\mathcal{B}_1$ assigns L_1 into N.

Example 7. (Continue Ex. 6.) After adding the fact $r_{21} : b(n_1, -1) \leftarrow$ into the set of facts $\mathcal{R}_{n_1}$, and $\mathcal{R}_{n_1}$ into the set of inference rules of ABA $\mathcal{F}'_2$, this ABA groundedly accepts $m'_4(n_1, -1)$. Consequently, $\mathcal{M}'_4$ assigns negative sentiment to n_1.

As a result, our second hybrid model, which employs the RBS, may produce the same explainable outputs in a more comprehensive manner. Table 12 shows the confusion matrix of $\mathcal{M}'^{0.8,0.96}_4$.

Finally, we compare our hybrid models to the CNN model, the CBR model, and various baseline models such as the lexicon-based RBS, LSTM, BERT, LR and SVM

[12] Assume that ABA $\mathcal{F}'_2$ uses same predicates as $\mathcal{F}_2$ for simplicity, with the exception that $m'_4(\_, \_)$ and $case_{\mathcal{M}'_4{}^{\alpha,\beta}}(\_, \_, \_)$ are used instead of $m_4(\_, \_)$ and $case_{\mathcal{M}_4^{\alpha,\beta}}(\_, \_, \_)$ respectively.

Table 12: Confusion matrix of $\mathcal{M}_4'^{0.8,0.96}$

$\mathcal{D}_3$ $\mathcal{M}'_4$	POS	NEG	NEU	URE	Total
POS	362	44	41	8	455
NEG	24	1,948	120	36	2,128
NEU	15	68	659	49	791
URE	10	44	148	1,605	1,807
Total	411	2,104	968	1,698	5,181

that are denoted as $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, $\mathcal{B}_4$ and $\mathcal{B}_5$ respectively. Table 13 shows the comparison of all models in terms of recall, precision, f-measure, accuracy, average f-measure and explainability rate (ER), where ER is computed by the following equation.

$$\text{ER} = \frac{\begin{array}{c}\text{the number of sentences having the same}\\ \text{sentiments predicted by an explainable model}\end{array}}{\text{the total number of predicted sentences}} \tag{3}$$

In Table 13, $\mathcal{M}_3$ uses Equation 1 to combine $\mathcal{M}_1$ and $\mathcal{M}_2$ based on the most similar sentence, $\mathcal{M}_4$ takes into account the second most similar sentence to improve $\mathcal{M}_3$ using the ABA framework, and $\mathcal{M}'_4$ is an alternative to $\mathcal{M}_4$ using the lexicon-based RBS for enhancing $\mathcal{M}_3$.

Example 8. (Continue Ex. 7.) According to previous examples, given the input sentence n_1, the CNN model classifies n_1 as neutral sentiment while the CBR model labels positive sentiment to n_1. However, both CNN and CBR models assign conflict sentiments to n_1 (for accuracy assessment, n_1 is annotated as negative sentiment). According to Ex. 5, the RBH model still assigns positive sentiment to n_1 using case ii of the system of equations. Basically, if case ii of the RBH model occurs, then the case ii of the ABH model occurs and vice versa. According to Ex. 6 the ABA $\mathcal{F}_2$ that structures the ABH model, groundedly accepts $m_4(n_1, -1)$, meaning that the ABH model assigns negative sentiment to n_1, by inference rule $r_{14.1}$ fires. Finally, the ABH model produces the final label to the input sentence.

6 Discussion

This study describes two hybrid models, RBH and ABH, which combine CNN and CBR through argumentation to justify CNN outputs in terms of CBR's analogous sentences. Thus, this study examines post-hoc explanation-by-examples for explaining CNN outputs at the instance level. Now, let us compare our approach to the related approaches (listed in Tables 1 and 14). In [82, 83], they also investigate hybrid systems that combine DL with CBR for interpreting DL outputs using post-hoc explanation-by-examples in XAI. However, they do not consider that DL and CBR may produce different outputs. In contrast, our approach addresses the conflicts between CNN and CBR using the ABA framework when they produce different sentiments for the same input sentence.

Concerning argumentative XAI, in [73, 79], they focus on merging DL with argumentation and constructing attacks and supports between arguments from contradictory texts. Through argumentation, [76] determines whether an input text is labeled with a sentiment produced by supervised classifiers or a different sentiment, and then it corrects the classifiers' misclassifications. In [77], it detects sentiment of an input query (*i.e.* word), with attacks and supports defined between queries generated from texts. All of these approaches, including ours, can provide explanations through argumentation. Our work, however, constructs arguments from inference rules supported by assumptions, with attacks arising from the contraries of assumptions. In [80], ANN was combined with CBR using the AA framework, in which the ANN selects features, and the AA-CBR takes these features as inputs for predicting new cases. However, in our approach, CNN and CBR label their respective sentiments for an input sentence. If they agree with the sentiment, our hybrid models will follow. Otherwise, our hybrid models use argumentation to determine which one to follow. Then, we compare to the other existing SA approaches and summarize it in Table 14 (Appendix C).

An ideal explanation for CNN should truly describe how CNN reaches a classification decision for a given input. However, to the best of our knowledge, no explanation systems so far are able to truly go after this ideal definition without extra assumptions about the internal working of the CNN. Indeed several researchers including [90] argued forcefully that at least for high-stake decisions, the aim should not be to explain black-box machine learning models like CNN but to design interpretable models instead. This is because if there is such an ideal explanation system, it would be less accurate than the (CNN) model to be explained (since otherwise, the explanation system would make the explained model redundant). But if the explanation system is not accurate, it might distract the user from following correct predictions of the black-box, and hence it raises a lot of doubt about the value of the explanation system. Hence, several recent studies on explainable AI (XAI) do not go after the ideal definition of black-box AI models. Instead, these studies try to justify the outcomes of CNN by another interpretable model without claiming any relationships between the produced justifications and the internal inference process of the CNN. For example, in [22] the authors propose to use the training dataset of the CNN to construct an argumentation framework which is then used to justify the outcomes of CNN. Indeed, one can also say that none of the existing local model-agnostic methods such as LIME [21] and LORE [91] truly explain the internal inference process of CNN, because they view CNN as a black-box. Sharing the same view, this paper does not aim to truly explain CNN's internal inference process. Instead, we propose to justify CNN outcomes by CBR model and combine two models by ABA frameworks. Though falling in the same line of work as Prakken and Ratsma's work [22], we do not assume that CBR has a known set of influential features as they do because we compute similarity measures between sentences using cosine similarity. Finally, we enhance our results by a lexicon-based model

Table 13: Comparison of Models' Performance: Recall, precision, f-measure, accuracy and average f-measure, and explainability rate.

Models	Sentiments	Recall	Precision	f-measure (F)	Accuracy & average F & ER
CNN ($\mathcal{M}_1$)	POS	75.67%	82.71%	79.03%	87.59% 84.00% N/A
	NEG	93.63%	89.26%	91.39%	
	NEU	66.74%	83.03%	74.00%	
	URE	94.88%	88.52%	91.59%	
CBR ($\mathcal{M}_2$)	POS	88.81%	52.67%	66.12%	71.57% 68.58% 100%
	NEG	84.60%	76.30%	80.23%	
	NEU	61.47%	55.71%	58.45%	
	URE	57.01%	89.05%	69.52%	
RBH ($\mathcal{M}_3$) ($\mathcal{M}_1 + \mathcal{M}_2 + Eq(1)$)	POS	86.86%	82.07%	84.40%	88.32% 85.66% 69.95%
	NEG	93.06%	91.24%	92.14%	
	NEU	67.77%	82.93%	74.59%	
	URE	94.52%	88.72%	91.53%	
ABH ($\mathcal{M}_4$) ($\mathcal{M}_3 + ABA$)	POS	88.08%	79.74%	83.70%	88.28% 85.56% 74.53%
	NEG	92.63%	91.46%	92.04%	
	NEU	67.98%	83.40%	74.90%	
	URE	94.52%	88.82%	91.58%	
ABH$^+$ ($\mathcal{M}_4'$) ($\mathcal{M}_3 + ABA + \mathcal{B}_1$)	POS	88.08%	79.56%	83.60%	88.28% 85.54% 74.53%
	NEG	92.59%	91.54%	92.06%	
	NEU	68.08%	83.31%	74.93%	
	URE	94.52%	88.82%	91.58%	
RBS ($\mathcal{B}_1$)	POS	15.33%	37.72%	21.80%	80.78% 67.30% 100%
	NEG	86.93%	84.21%	85.55%	
	NEU	75.93%	68.12%	71.81%	
	URE	91.76%	88.37%	90.03%	
LSTM ($\mathcal{B}_2$)	POS	77.13%	91.62%	83.75%	86.82% 84.41% N/A
	NEG	90.64%	89.83%	90.23%	
	NEU	69.11%	78.15%	73.36%	
	URE	94.52%	86.48%	90.32%	
BERT ($\mathcal{B}_3$)	POS	46.47%	71.80%	56.43%	72.71% 66.15% N/A
	NEG	78.61%	78.24%	78.43%	
	NEU	45.76%	51.57%	48.49%	
	URE	87.10%	76.16%	81.26%	
LR ($\mathcal{B}_4$)	POS	46.47%	82.33%	59.41%	68.48% 63.47% N/A
	NEG	68.25%	71.98%	70.07%	
	NEU	39.88%	59.38%	47.71%	
	URE	90.40%	66.62%	76.71%	
SVM ($\mathcal{B}_5$)	POS	2.19%	81.82%	4.27%	62.57% 42.81% N/A
	NEG	71.58%	63.60%	67.35%	
	NEU	16.12%	66.95%	25.98%	
	URE	92.50%	61.15%	73.63%	

which detects influential features (sentimental words).

In this paper, we cannot compare the accuracy of our approach with that of the above approaches because the accuracy is achieved using a different dataset. Instead, as shown in Table 13, we use the same testing dataset to compare the accuracies of our hybrid models to those of state-of-the-art models such as CNN, LSTM, and BERT. Our hybrid models outperform them slightly better. None of these state-of-the-art models explain their outputs. However, our hybrid models justify 69.95% and 74.53% of the outputs, based on the sentiments of similar sentences, respectively. Thus, our hybrid models provide interpretability without sacrificing accuracy.

In summary, assuming that CNN is a black-box, we justify its outcomes by an interpretable model. Providing justifications is not as good as providing genuine explanations but many studies [22, 67, 81, 82, 83, 84, 90, 92, 93] including Prakken and Ratsma's work have proved that the former is beneficial. As we do not assume any knowledge about the internals of CNN, our approach can be readily applied to other ML formalisms. Note that providing justifications by past cases as what we do is not uncommon in daily life. For example, in the legal domain, judges often use precedents to justify their decisions in the current case (the doctrine of stare decisis). Hence we believe that our approach is not unnatural for human users.

7 Conclusion

CNN lends itself to one of the most accurate methods for sentiment analysis (SA) [18, 19] but suffers from the interpretability problem due to its black-box nature. In the so-called XAI research, to tackle this problem one often uses a trained CNN to generate data for training an interpretable model such as DT, which is then used to explain how the original CNN works [94, 95].

In this paper, we take another route: combining CNN with CBR, which is constructed using a different dataset. In particular, we develop several hybrid models which in the ideal case take the output of CNN for a given unlabeled sentence and use the similar sentence from CBR to explain this output - a kind of explanation by analogy. Since we focus on the post-hoc explanation of XAI, our hybrid models take each input-output pair of the CNN model as a case. Similar to our work, other studies in the literature have combined CNN and CBR to produce twin systems, and LIME and LORE also provide post-hoc explanations for ML models. Because of the trade-off between interpretability and accuracy in XAI, the more interpretable a model is, the less accurate it is. In demonstration, our hybrid models balance this trade-off issue by interpreting CNN outputs without sacrificing CNN accuracy for achieving interpretability.

Let us close with some technical details on the models' development. We train the CNN model built on the top of word2vec using 15,413 sentences while we build the CBR model by cosine similarity using 4,510 sentences. Both models produce labels for an input sentence but their label assignments may either agree or disagree. In the case of agreement, our first hybrid model assigns the same label produced by both CNN and CBR models to the input sentence. In the case of disagreement, the first hybrid model combines them via the system of equations to determine which model should be followed. Using the test dataset, the first hybrid model can interpret 69.95% of the outputs according to the sentiments of similar sentences. Since the first model produces outputs based on the most similar sentence to the input sentence, our second hybrid model modifies it by considering labels of the second (third, *etc.*) similar sentences. Using the test dataset, the second hybrid model can interpret 74.53% of the outputs according to the sentiments of similar sentences. In overall accuracy, our hybrid models achieve 88.32% and 88.28% respectively while the CNN model and the CBR model get 87.59% and 71.57% respectively.

In general, the accuracy of a hybrid model mostly depends on the accuracy of its input models. This phenomenon has been reported quantitatively in, for example [20, 96]. Hence, the most reliable way to improve the accuracies of our hybrid models is to train the input CNN with more data. Note that our hybrid models have achieved the accuracy of the input CNN, and one might ask whether we can still increase their accuracies without improving the input CNN. Clearly, a positive answer here means that we can provide explanation models that are more accurate than the CNN models that call for explanation. And as argued by [90] (see the previous section), such explanation models would make CNN models redundant. The main objective of our work is to provide justifications for CNN outputs via hybrid interpretable models that retain the accuracy of CNN. We do not focus on significantly increasing CNN accuracy. In the future, we shall experiment with different CNN models such as BERT. As argued, more accurate input CNNs lead to more accurate hybrid models but the replacement of input CNN does not change the basics of our approach since we assume CNNs as black-boxes.

As limitations, cases ii.2 and ii.3 of the system of equations are domain specific conditions for our approach, which has been developed in the dataset with an unbalanced number of positive and negative sentences. In the future, we will solve the limitation using a data sampling method to balance the number of sentences in the dataset and figure out other combination ways of different models rather than CNN and CBR. Using multiple datasets, we will advance our contribution by interpreting multiple state-of-the-art models such as LSTM and BERT rather than CNN. Furthermore, by excluding one of the datasets from $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$, we will evaluate the performance of our hybrid models based on ablation studies. Then, probabilistic argumentation, or DST, would be used in order to determine the probability of the explanations produced by our hybrid models.

Acknowledgments

This research is supported by Center of Excellence in Intelligent Informatics, Speech and Language Technology and Service Innovation (CILS), Intelligent Informatics and Service Innovation (IISI) Research Center of Sirindhorn International Institute of Technology and the TRF Research Team Promotion Grant (RTA), the Thailand Research Fund under the grant number RTA6280015.

References

[1] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[2] Subhabrata Mukherjee and Pushpak Bhattacharyya. Sentiment analysis: A literature survey. *arXiv preprint arXiv:1304.4520*, 2013.

[3] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[4] Kitsuchart Pasupa and Thititorn Seneewong Na Ayutthaya. Thai sentiment analysis with deep learning techniques: A comparative study based on word embedding, pos-tag, and sentic features. *Sustainable Cities and Society*, 2019.

[5] Neha Nandal, Rohit Tanwar, and Jyoti Pruthi. Machine learning based aspect level sentiment analysis for amazon products. *Spatial Information Research*, pages 1–7, 2020.

[6] D Shubham, P Mithil, Meesala Shobharani, and S Sumathy. Aspect level sentiment analysis using machine learning. In *Materials Science and Engineering Conference Series*, volume 263, page 042009, 2017.

[7] Duyu Tang, Bing Qin, and Ting Liu. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432, 2015.

[8] Richa Sharma, Shweta Nigam, and Rekha Jain. Opinion mining of movie reviews at document level. *arXiv preprint arXiv:1408.3829*, 2014.

[9] Zhifei Zhang, Duoqian Miao, Zhihua Wei, and Lei Wang. Document-level sentiment classification based on behavior-knowledge space method. In *International Conference on Advanced Data Mining and Applications*, pages 330–339. Springer, 2012.

[10] John Rothfels and Julie Tibshirani. Unsupervised sentiment classification of english movie reviews using automatic selection of positive and negative sentiment items. *CS224N-Final Project*, 43(2):52–56, 2010.

[11] VS Jagtap and Karishma Pawar. Analysis of different approaches to sentence-level sentiment classification. *International Journal of Scientific Engineering and Technology*, 2(3):164–170, 2013.

[12] Efstratios Kontopoulos, Christos Berberidis, Theologos Dergiades, and Nick Bassiliades. Ontology-based sentiment analysis of twitter posts. *Expert systems with applications*, 40(10):4065–4074, 2013.

[13] S Behdenna, Fatiha Barigou, and Ghalem Belalem. Document level sentiment analysis: A survey. *EAI Endorsed Transactions on Context-Aware Systems and Applications*, 4(13), 2018.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[15] Jianxin Wu. Introduction to convolutional neural networks. *National Key Lab for Novel Software Technology. Nanjing University. China*, 5:23, 2017.

[16] Larry Medsker and Lakhmi C Jain. *Recurrent neural networks: design and applications*. CRC press, 1999.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[18] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

[19] Qurat Tul Ain, Mubashir Ali, Amna Riaz, Amna Noureen, Muhammad Kamran, Babar Hayat, and A Rehman. Sentiment analysis using deep learning techniques: a review. *Int J Adv Comput Sci Appl*, 8(6):424, 2017.

[20] David Gunning, Mark Stefik, Jaesik Choi, Timothy Miller, Simone Stumpf, and Guang-Zhong Yang. Xai-explainable artificial intelligence. *Science Robotics*, 4(37), 2019.

[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, 2016.

[22] Henry Prakken and Rosa Ratsma. A top-level model of case-based argumentation for explanation: formalisation and experiments. *Argument & Computation*, (Preprint):1–36.

[23] Kristijonas Čyras, Antonio Rago, Emanuele Albini, Pietro Baroni, and Francesca Toni. Argumentative xai: A survey. *arXiv preprint arXiv:2105.11266*, 2021.

[24] Phan Minh Dung, Robert A Kowalski, and Francesca Toni. Assumption-based argumentation. In *Argumentation in artificial intelligence*, pages 199–218. Springer, 2009.

[25] Ahmed Al-Ani and Mohamed Deriche. A new technique for combining multiple classifiers using the dempster-shafer theory of evidence. *Journal of Artificial Intelligence Research*, 17:333–361, 2002.

[26] Lei Xu, Adam Krzyzak, and Ching Y Suen. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, 22(3):418–435, 1992.

[27] Galina Rogova. Combining the results of several neural network classifiers. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*, pages 683–692. Springer, 2008.

[28] David A Bell, Ji-wen W Guan, and Yaxin Bi. On combining classifier mass functions for text categorization. *IEEE transactions on knowledge and data engineering*, 17(10):1307–1319, 2005.

[29] Yaxin Bi. The impact of diversity on the accuracy of evidential classifier ensembles. *International Journal of Approximate Reasoning*, 53(4):584–607, 2012.

[30] Cuong Anh Le, Van-Nam Huynh, Akira Shimazu, and Yoshiteru Nakamori. Combining classifiers for word sense disambiguation based on dempster–shafer theory and owa operators. *Data & Knowledge Engineering*, 63(2):381–396, 2007.

[31] Van-Nam Huynh, Tri Thanh Nguyen, and Cuong Anh Le. Adaptively entropy-based weighting classifiers in combination using dempster–shafer theory for word sense disambiguation. *Computer Speech & Language*, 24(3):461–473, 2010.

[32] Vahid Yaghoubi, Liangliang Cheng, Wim Van Paepegem, and Mathias Kersemans. A novel multi-classifier information fusion based on dempster–shafer theory: application to vibration-based fault detection. *Structural Health Monitoring*, page 14759217211007130, 2020.

[33] Chenbin Zhang, Ningning Qin, and Le Yang. Optimal combination of svm and bayesian density model using dempster-shafer theory. In *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, pages 505–509, 2020.

[34] Sajjad Talesh Hosseini, Omid Asghari, and Parham Pahlavani. A hybrid approach to model the dykes in sungun porphyry copper deposit using dempster–shafer theory. *Arabian Journal of Geosciences*, 13(24):1–20, 2020.

[35] Sergio Peñafiel, Nelson Baloian, Horacio Sanson, and José A Pino. Applying dempster–shafer theory for developing a flexible, accurate and interpretable classifier. *Expert Systems with Applications*, 148:113262, 2020.

[36] Amalendu Si, Sujit Das, and Samarjit Kar. Picture fuzzy set-based decision-making approach using dempster–shafer theory of evidence and grey relation analysis and its application in covid-19 medicine selection. *Soft Computing*, pages 1–15, 2021.

[37] Thimmaiah Gudiyangada Nachappa, Sepideh Tavakkoli Piralilou, Khalil Gholamnia, Omid Ghorbanzadeh, Omid Rahmati, and Thomas Blaschke. Flood susceptibility mapping with machine learning, multi-criteria decision analysis and ensemble using dempster shafer theory. *Journal of Hydrology*, page 125275, 2020.

[38] Dai Quoc Nguyen, Dat Quoc Nguyen, Thanh Vu, and Son Bao Pham. Sentiment classification on polarity reviews: an empirical study using rating-based features. 2014.

[39] Bernhard Lutz, Nicolas Pröllochs, and Dirk Neumann. Sentence-level sentiment analysis of financial news using distributed text representations and multi-instance learning. *arXiv preprint arXiv:1901.00400*, 2018.

[40] Bishan Yang and Claire Cardie. Context-aware learning for sentence-level sentiment analysis with posterior regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 325–335, 2014.

[41] Vrushali K Bongirwar. A survey on sentence level sentiment analysis. *International Journal of Computer Science Trends and Technology (IJCST)*, 3(3), 2015.

[42] Kim Schouten and Flavius Frasincar. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830, 2015.

[43] Yu Mon Aye and Sint Sint Aung. Senti-lexicon and analysis for restaurant reviews of myanmar text. *International Journal of Advanced Engineering, Management and Science*, 4(5), 2018.

[44] Anna Jurek, Maurice D Mulvenna, and Yaxin Bi. Improved lexicon-based sentiment analysis for social media analytics. *Security Informatics*, 4(1):1–13, 2015.

[45] Cataldo Musto, Giovanni Semeraro, and Marco Polignano. A comparison of lexicon-based approaches for sentiment analysis of microblog posts. In *DART@ AI* IA*, pages 59–68, 2014.

[46] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307, 2011.

[47] Nitika Nigam and Divakar Yadav. Lexicon-based approach to sentiment analysis of tweets using r language. In *International Conference on Advances in Computing and Data Sciences*, pages 154–164. Springer, 2018.

[48] Ali Hasan, Sana Moin, Ahmad Karim, and Shahaboddin Shamshirband. Machine learning-based sentiment analysis for twitter accounts. *Mathematical and Computational Applications*, 23(1):11, 2018.

[49] Deepak Kumar Gupta and Asif Ekbal. Iitp: supervised machine learning for aspect based sentiment analysis. 2014.

[50] Zhang Hailong, Gan Wenyan, and Jiang Bo. Machine learning and lexicon based methods for sentiment classification: A survey. In *2014 11th Web Information System and Application Conference*, pages 262–265. IEEE, 2014.

[51] Lei Zhang, Riddhiman Ghosh, Mohamed Dekhil, Meichun Hsu, and Bing Liu. Combining lexicon-based and learning-based methods for twitter sentiment analysis. *HP Laboratories, Technical Report HPL-2011*, 89, 2011.

[52] Nipuna Upeka Pannala, Chamira Priyamanthi Nawarathna, JTK Jayakody, Lakmal Rupasinghe, and Kesavan Krishnadeva. Supervised learning based approach to aspect based sentiment analysis. In *2016 IEEE International Conference on Computer and Information Technology (CIT)*, pages 662–666. IEEE, 2016.

[53] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

[54] Asad Abdi, Siti Mariyam Shamsuddin, Shafaatunnur Hasan, and Jalil Piran. Deep learning-based sentiment classification of evaluative text based on multi-feature fusion. *Information Processing & Management*, 56(4):1245–1259, 2019.

[55] Sujata Rani and Parteek Kumar. Deep learning based sentiment analysis using convolution neural network. *Arabian Journal for Science and Engineering*, 44(4):3305–3314, 2019.

[56] Ashish Kumar and Aditi Sharan. Deep learning-based frameworks for aspect-based sentiment analysis. In *Deep Learning-Based Approaches for Sentiment Analysis*, pages 139–158. Springer, 2020.

[57] Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–35, 2021.

[58] Olivier Habimana, Yuhua Li, Ruixuan Li, Xiwu Gu, and Ge Yu. Sentiment analysis using deep learning approaches: an overview. *Science China Information Sciences*, 63(1):1–36, 2020.

[59] Aliaksei Severyn and Alessandro Moschitti. Twitter sentiment analysis with deep convolutional neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 959–962. ACM, 2015.

[60] Cicero Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[61] Peerapon Vateekul and Thanabhat Koomsubha. A study of sentiment analysis using deep learning techniques on thai twitter data. In *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6. IEEE, 2016.

[62] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[63] Hannah Kim and Young-Seob Jeong. Sentiment classification using convolutional neural networks. *Applied Sciences*, 9(11):2347, 2019.

[64] Jan Milan Deriu and Mark Cieliebak. Sentiment analysis using convolutional neural networks with multi-task training and distant supervision on italian tweets. In *Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Napoli, Italy, December 5-7, 2016*. Italian Journal of Computational Linguistics, 2016.

[65] Igor Santos, Nadia Nedjah, and Luiza de Macedo Mourelle. Sentiment analysis using convolutional neural network with fasttext embeddings. In *2017 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–5. IEEE, 2017.

[66] Nadia Nedjah, Igor Santos, and Luiza de Macedo Mourelle. Sentiment analysis using convolutional neural network via word embeddings. *Evolutionary Intelligence*, pages 1–25, 2019.

[67] JT Turner, Michael W Floyd, Kalyan Gupta, and Tim Oates. Nod-cc: A hybrid cbr-cnn architecture for novel object discovery. In *International Conference on Case-Based Reasoning*, pages 373–387. Springer, 2019.

[68] Faliang Huang, Xuelong Li, Changan Yuan, Shichao Zhang, Jilian Zhang, and Shaojie Qiao. Attention-emotion-enhanced convolutional lstm for sentiment analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[69] Xiaoyan Yan, Fanghong Jian, and Bo Sun. Sakg-bert: Enabling language representation with knowledge graphs for chinese sentiment analysis. *IEEE Access*, 9:101695–101701, 2021.

[70] Petr Berka. Sentiment analysis using rule-based and case-based reasoning. *Journal of Intelligent Information Systems*, pages 1–16, 2020.

[71] Marco Lippi and Paolo Torroni. Argument mining: A machine learning perspective. In *International Workshop on Theory and Applications of Formal Argumentation*, pages 163–176. Springer, 2015.

[72] Oana Cocarascu and Francesca Toni. Argumentation for machine learning: A survey. In *COMMA*, pages 219–230, 2016.

[73] Oana Cocarascu and Francesca Toni. Mining bipolar argumentation frameworks from natural language text. 2017.

[74] Rihab Bouslama, Raouia Ayachi, and Nahla Ben Amor. Using convolutional neural network in cross-domain argumentation mining framework. In *International Conference on Scalable Uncertainty Management*, pages 355–367. Springer, 2019.

[75] Vlad Niculae, Joonsuk Park, and Claire Cardie. Argument mining with structured svms and rnns. *arXiv preprint arXiv:1704.06869*, 2017.

[76] Lucas Carstens and Francesca Toni. Using argumentation to improve classification in natural language problems. *ACM Transactions on Internet Technology (TOIT)*, 17(3):30, 2017.

[77] Kathrin Grosse, María P González, Carlos I Chesnevar, and Ana G Maguitman. Integrating argumentation and sentiment analysis for mining opinions from twitter. *AI Communications*, 28(3):387–401, 2015.

[78] Oana Cocarascu and Francesca Toni. Identifying attack and support argumentative relations using deep learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1374–1379, 2017.

[79] Oana Cocarascu and Francesca Toni. Combining deep learning and argumentative reasoning for the analysis of social media textual content using small data sets. *Computational Linguistics*, 44(4):833–858, 2018.

[80] Oana Cocarascu, Kristijonas Cyras, and Francesca Toni. Explanatory predictions with artificial neural networks and argumentation. 2018.

[81] Mark T Keane and Eoin M Kenny. How case-based reasoning explains neural networks: A theoretical analysis of xai using post-hoc explanation-by-example from a survey of ann-cbr twin-systems. In *International Conference on Case-Based Reasoning*, pages 155–171. Springer, 2019.

[82] Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[83] Alina Jade Barnett, Fides Regina Schwartz, Chaofan Tao, Chaofan Chen, Yinhao Ren, Joseph Y Lo, and Cynthia Rudin. Iaia-bl: A case-based interpretable deep learning model for classification of mass lesions in digital mammography. *arXiv preprint arXiv:2103.12308*, 2021.

[84] David Leake and David Crandall. On bringing case-based reasoning methodology to deep learning. In *International Conference on Case-Based Reasoning*, pages 343–348. Springer, 2020.

[85] David Leake, Xiaomeng Ye, and David J Crandall. Supporting case-based reasoning with neural networks: An illustration for case adaptation. In *AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering*, 2021.

[86] Michael M Richter and Rosina O Weber. *Case-based reasoning*. Springer, 2016.

[87] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[88] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial intelligence*, 77(2):321–357, 1995.

[89] Saung Hnin Pwint Oo, Thanaruk Theeramunkong, and Nguyen Duy Hung. Sentence sentiment classification using convolutional neural network in myanmar texts. In *Proceedings of the 2020 2nd International Conference on Image, Video and Signal Processing*, pages 144–149, 2020.

[90] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. nat mach intell. 2019; 1: 206–215.

[91] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*, 2018.

[92] Mark T Keane and Eoin M Kenny. The twin-system approach as one generic solution for xai: An overview of ann-cbr twins for explaining deep learning. *arXiv preprint arXiv:1905.08069*, 2019.

[93] Jérémie Clos, Nirmalie Wiratunga, and Stewart Massie. Towards explainable text classification by jointly learning lexicon and modifier terms. 2017.

[94] Dino Pedreschi, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Luca Pappalardo, Salvatore Ruggieri, and Franco Turini. Open the black box data-driven explanation of black box decision systems. *arXiv preprint arXiv:1806.09936*, 2018.

[95] Dino Pedreschi, Fosca Giannotti, Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, and Franco

Turini. Meaningful explanations of black box ai decision systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 9780–9784, 2019.

[96] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.

Appendix A

The model $\mathcal{M}_3^{\alpha,\beta}$ can be represented by the ABA framework $\mathcal{F}_1 = (\mathcal{A}_1, \mathcal{R}_1, \overline{})$ where $\mathcal{A}_1$ is empty and $\mathcal{R}_1 = \{r_1, ..., r_7\}$ is a set of inference rules containing the following rules.

- Each inference rule (from r_1 to r_6) represents each case of the system of equations 1 on page 8.

$$
\begin{array}{lll}
r_1: & case(i, N, L) \leftarrow & m_1(N, L), m_2(N, S, L). \\
r_2: & case(ii_1, N, L_2) \leftarrow & m_1(N, L_1), m_1(S, L_2), \\
 & & m_2(N, S, L_2), L_1 \neq L_2, \\
 & & p(N, L_1, X), p(S, L_2, Y), \\
 & & Y > X, sim(N, S, Z), \\
 & & Z \geq \beta. \\
r_3: & case(ii_2, N, L_2) \leftarrow & m_1(N, L_1), m_1(S, L_2), \\
 & & m_2(N, S, L_2), L_1 \neq L_2, \\
 & & L_1 = -1, L_2 = +1, \\
 & & p(N, L_1, X), p(S, L_2, Y), \\
 & & Y > X, sim(N, S, Z), \\
 & & \alpha < Z < \beta. \\
r_4: & case(ii_3, N, L_2) \leftarrow & m_1(N, L_1), m_1(S, L_2), \\
 & & m_2(N, S, L_2), L_1 \neq L_2, \\
 & & L_1 \neq -1, L_2 = +0, \\
 & & p(N, L_1, X), p(S, L_2, Y), \\
 & & Y > X, sim(N, S, Z), \\
 & & \alpha < Z < \beta. \\
r_5: & case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L) \leftarrow & \sim cae(i, N, \_), \\
 & & case(ii_1, N, L); \\
 & & case(ii_2, N, L); \\
 & & case(ii_3, N, L). \\
r_6: & case(iii, N, L) \leftarrow & m_1(N, L), \\
 & & \sim case(i, N, \_), \\
 & & \sim case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, \_).
\end{array}
$$

- The following rule r_7 represents the outcome of the model $\mathcal{M}_3$ to assign a label to a input sentence.

$$
\begin{array}{lll}
r_7: & m_3(N, L) \leftarrow & case(i, N, L); \\
 & & case_{\mathcal{M}_3^{\alpha,\beta}}(ii, N, L); \\
 & & case(iii, N, L).
\end{array}
$$

Given an input sentence N described by a set of facts $\mathcal{R}_N$, the label that $\mathcal{M}_3$ assigns to N is computed by ABA framework $(\mathcal{A}_1, \mathcal{R}_1 \cup \mathcal{R}_N, \overline{})$ obtained from the above ABA framework $\mathcal{F}_1$ by adding $\mathcal{R}_N$ into the set of inference rules (see Ex. 5).

Appendix B

The algorithmic form of $\mathcal{M}_3^{0.8,0.96}$ is expressed as the following procedure.

```
procedure $m_3\_label\_assignment(n$ : new sentence)
    $s$ := a similar sentence
    $\mathcal{L} := \{+1, -1, +0, -0\}$//i.e. set of labels
    $l_1$ := label that is assigned by $\mathcal{M}_1$ to $n$
    $l_2$ := label that is assigned by $\mathcal{M}_2$ to $n$
    $l_3$ := label that is assigned by $\mathcal{M}_1$ to $s$
    $p_{l_1}$ := conditional probability of $l_1$ given $n$
    $p_{l_3}$ := conditional probability of $l_3$ given $s$
    $sim$ := similarity measure between $s$ and $n$
    $\alpha := 0.8$
    $\beta := 0.96$
    if $l_1 = l_2$ then
        assign $l_1$ into $n$
    else
        if $l_3 = l_2$ and $p_{l_1} < p_{l_3}$then
            if $sim >= \beta$ then
                assign $l_2$ into $n$
            if $\alpha < sim < \beta$ then
                if ($l_1 = -1$ and $l_2 = +1$) or
                    ($l_1 \neq -1$ and $l_2 = +0$) then
                    assign $l_2$ into $n$
                else  assign $l_1$ into $n$
            else  assign $l_1$ into $n$
        else  assign $l_1$ into $n$
```

Appendix C

The following tables summarize existing approaches to SA.

Table 14: Summary of existing approaches to SA

Reference #	Proposed	Result	Comparison to our work
[23]	Argumentative-based explanations for intrinsic and post-hoc explanations in XAI approaches were surveyed.	The paper highlighted some gaps in the state-of-the-art models for argumentation-based XAI and suggested future research directions.	Our approach also uses argumentation in XAI where the combined process of CNN and CBR is argumentative and hence self-explainable.
[59]	A three-step process was proposed to train the CNN model using the SemEval-2015 corpus after initializing the parameter weights from a CNN with a pre-trained word2vec model.	The constructed CNN model performed the phrase-level and message-level subtasks using the official test sets provided by the SemEval-2015 campaign, achieving 84.79% accuracy and 64.59% accuracy, respectively.	They remain limitation of black-box models to provide interpretability that cannot be directly derived from the models and provided for their outputs. Our approach overcomes this limitation via post-hoc explanation-by examples, classifying sentiments for sentences by CNN while justifying the CNN outputs by similar sentences from CBR.
[60]	The paper proposed a deep CNN to classify the sentiments of short texts using character- to sentence-level information.	The deep CNN achieved 86.4% accuracy for the Stanford twitter sentiment corpus and 85.7% accuracy for the Stanford sentiment treebank corpus.	
[61]	The study proposed to classify the sentiment of Thai twitter data using two DL techniques: LSTM and dynamic CNN, while considering the impact of word order in tweets.	With the exception of MaxEnt, the results showed that LSTM and dynamic CNN outperformed NB and SVM with an accuracy of 75.30% and 75.35%, respectively.	
[62]	A series of architectural modifications to CNN were proposed to train for sentence-level classification tasks on top of pre-trained word vectors, enhancing the use of both task-specific and static vectors.	In a series of experiments, these CNNs with fine-tuned hyperparameters outperformed other state-of-the-art models on four out of seven tasks, where CNN-static obtained 89.6% accuracy on a dataset about opinion polarity detection.	

Reference #	Proposed	Result	Comparison to our work
[63]	On three datasets, the paper offered several fine-tuned CNN models for sentence sentiment classification experiments.	The experimental results showed that CNN with consecutive convolutional layers performed well with long texts, achieving a weighted-F1 score of 81% for binary classification and a weighted-F1 score of 68% for ternary classification, respectively.	They also remain same limitation of black-box models to provide interpretability.
[64]	A 2-layer CNN classifier employing multi-task training was proposed, using a huge amount of weakly labeled data to predict the sentiment of Italian tweets.	In an experiment using test sets from the EvalItalia-2016 competition, it obtained 65.2% accuracy and 66% accuracy using a single-task training approach and a multi-task training approach after using cross-validation, respectively.	
[65]	The paper proposed a CNN with fastText word embeddings in order to perform sentence sentiment classification on three datasets.	The performance of CNN with fastText outperformed traditional ML techniques such as NB, SVM, and LR, achieving 85.2% accuracy on the Movie review dataset, which was comparable to the 85% accuracy of CNN with word2vec.	
[66]	The paper proposed CNN classifiers for SA that highlighted how the hyper-parameters affected the classifier's performance.	When CNNs with different configurations are compared, the setting of key parameters yields the best classification accuracy, with 86.6%, tested on the Stanford sentiment treebank dataset.	
[68]	By combining emotional intelligence and attention mechanisms, the author proposed an LSTM model, known as AEC-LSTM, to improve LSTM's capacity to identify emotion modulation (*i.e.* abstraction level) for textual data.	The model was tested on four datasets, including the IMDB dataset, where it outperformed other conventional ML algorithms with an accuracy of 96.3%.	

Reference #	Proposed	Result	Comparison to our work
[67]	The paper proposed an approach that switched classification between CNN and CBR in order to detect distinct types of objects in images from the PASCAL-Part dataset.	Always-CNN's classification accuracy was 61.27%, always-CBR's was 68.93%. When there were data-poor situations, CNN's classification was switched to CBR, which got 64.12% accuracy.	They did not solve conflicts that occurs in the combination process. Our approach solves such conflict and build CNN and CBR using distinct datasets: CNN for accuracy and CBR for interpretability.
[69]	The SAKG-BERT model was proposed, which combined the language representation model BERT and SA knowledge to increase the DL algorithm's interpretability.	With 95% accuracy on the Car review dataset and 95.3% accuracy on the Chnsenticorp dataset, SAKG-BERT surpassed the two BERT baselines in comparison.	
[70]	The paper proposed a technique for performing SA that combined rule-based system (RBS) and CBR, with their strengths and limitations.	The decision-making process can be insightful because it is based on rules or cases.	
[81]	This paper proposed a theoretical survey of twin systems that combined ANNs with CBRs in order to solve the XAI problem via post-hoc explanation.	In the paper, further directions for this XAI solution were outlined, including feature-weighting techniques.	
[84]	This paper proposed integrating CBR with those DL methods in order to address DL challenges, such as learning from a few samples.	The paper concentrated on how the CBR can assist in addressing the DL challenges.	
[74]	Using three different corpora, the paper proposed a framework that takes the effectiveness of character-level and word-based CNNs for argument mining based on in-domain and cross-domain cases.	In the cross-domain case-on-essays of the Test-on-Essays corpora, word-based CNN surpasses char-based CNN, SVM, and NB with an accuracy of 98%.	Unlike to our approach, although arguments and the components that make up arguments are detected, the resolution of conflicts between arguments was not addressed.