IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received April 7, 2020, accepted April 17, 2020, date of publication April 27, 2020, date of current version May 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990306

Combination of Recursive and Recurrent Neural Networks for Aspect-Based Sentiment Analysis Using Inter-Aspect Relations

CEM RIFKI AYDIN AND TUNGA GÜNGÖR
Department of Computer Engineering, Boğaziçi University, 34342 Istanbul, Turkey

Corresponding author: Cem Rıfkı Aydın (cem.aydin1@boun.edu.tr)

This work was supported in part by the Boğaziçi University Research Fund (BAP) under Grant 6980D, and in part by the Turkish Directorate of Strategy and Budget under the TAM Project under Grant 2007K12-873. The work of Cem Rıfkı Aydın was supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under Grant BİDEB 2211.

ABSTRACT Sentiment analysis studies in the literature mostly use either recurrent or recursive neural network models. Recurrent models capture the effect of time and propagate the information of sentiment labels in a review throughout the word sequence. Recursive models, on the other hand, extract syntactic structures from the texts and leverage the sentiment information during training. There are only a few studies that incorporate both of these models into a single neural network for the sentiment classification task. In this paper, we propose a novel neural network framework that combines recurrent and recursive neural models for aspect-based sentiment analysis. By using constituency and dependency parsers, we first divide each review into subreviews that include the sentiment information relevant to the corresponding aspect terms. After generating and training the recursive neural trees built from the parses of the subreviews, we feed their output into the recurrent model. We evaluated our ensemble approach on two datasets in English of different genres. We achieved state-of-the-art results and outperformed the baseline study by a significant margin for both domains.

INDEX TERMS Aspect-based sentiment classification, ensemble neural network model, recurrent neural networks, recursive neural networks, sentiment analysis.

I. INTRODUCTION

Sentiment analysis is the task of identification and quantification of sentiments in reviews. This area is one of the hottest topics in natural language processing (NLP) due to its vast practical usage in social media, marketing, and political analyses. Reviews are generally collected from popular websites, such as eBay and Amazon, and are processed to detect which products are favorable and which are not. Based on the criticism in the reviews, companies supplying products can ameliorate their products by allocating more resources to development or increase the contentment of their customers and the prestige of their brand.

Although, in general, a single review expresses a sentiment towards an entity such as a product, service or political act, it is also possible to make different comments towards different aspects of the entity in a single review. For instance, in *"I found the ambiance of the restaurant great overall; however, the main dish was served a bit cold and lately."*, sentiments expressed for the aspects *"ambiance"* and *"dish"* are positive and negative, respectively. Hence, a finer analysis than considering the review as including a single opinion has to be made when performing sentiment classification.

When a review contains multiple aspects, the sentiment of an aspect is likely to have an impact on the following aspects as well. For example, in the sentence *"I liked the taste of pizza more than that of the chips."*, it is seen that the negative sentiment of the aspect *"chip"* is expressed indirectly as per the first aspect *"pizza."* That is, the polarities of aspects are likely to affect each other in a single review. Also, conjunctions like *"and"*, *"also"*, *"however"*, and *"but"* cause aspects to share their sentiments with other aspects or influence the sentiments of other aspects. In *"The quality of the display of this laptop is so sensational, so is the price thereof."*, there is a correlation between the sentiments of the aspects due to the use of the conjunction *"so."* To model this scenario, a study

The associate editor coordinating the review of this manuscript and approving it for publication was Maged Abdullah Esmail.

develops a recurrent neural network (RNN) framework using such inter-aspect relationships [1].

Recurrent models like the above make use of the sequence information in a series of objects. This helps propagate the impact of the sentiments to the preceding or succeeding words in a text. However, these models lack the information of the grammatical structures of texts. For instance, when a sentence is parsed into its constituent tokens, the words in the same subtree are expected to be semantically and syntactically more similar to each other than those in the others. The use of recursive neural networks can, therefore, be useful to assign the same or similar sentiments to the words located in the same subtrees of the parsed text. Incorporating this structural and sentiment information captured by recursive neural networks into other neural network structures, such as recurrent models, can yield a more comprehensive and robust framework. In this way, recurrent and recursive models can compensate for what the other lacks when used in an ensemble system.

In this paper, we propose a framework for the task of aspect-based sentiment analysis [2]. To capture the sentiments of aspects, we combine recurrent and recursive neural network models. For the recurrent model, we use the off-the-shelf framework [1] that uses gated recurrent units (GRU). As for the recursive module, we develop novel ways for extracting subreviews, which correspond to aspect term groups, from reviews. These subreviews are obtained in such a way that each is modified by exactly one sentiment. Each subreview is treated as a separate review and trained using recursive neural networks. The root sentiment embedding from each subreview is obtained in a distant-supervised manner, which corresponds to the embedding of the related aspect. These embeddings are then fed into the recurrent model as input.

We evaluate our novel ensemble approach on two datasets: restaurant and laptop datasets of SemEval-2014, Task 4. We outperform the baseline study [1] by a significant margin. Using only the recurrent module cannot capture sentiment information effectively. The recursive module generates the *"optimal"* sentiment root vectors of the subreviews and we merge them with the corresponding aspect component embeddings in the recurrent network. By adding a recursive model to the overall model with a distant-supervised approach in an original way, we increased the success rates by 1.6% on average for both domains. The source code is publicly available.[1]

Our research objective in this study is to enhance the success rates for aspect-based sentiment analysis (ABSA). To reach this goal, we address the following research questions. Can recurrent neural networks be enhanced by incorporating distant sentiment information? Can recurrent and recursive neural network models be merged? If yes, what is its effect? Why do such ensemble methods outperform those modeling only one of its subcomponents? Can reviews be consistently partitioned into subreviews using syntactic parsers so that each subreview holds only one relevant sentiment expressed towards the aspect(s) therein? Does utilizing dependency parsers outperform the use of constituency parsers in aspect-based sentiment analysis, and, if yes, why?

The rest of the paper is organized as follows. Section 2 presents the existing works on aspect-based sentiment analysis, recurrent and recursive neural networks, and their ensemble forms used in this classification task. We describe our models in Section 3. Section 4 explains the datasets and the experimental results, and discusses the main contributions of the proposed approach. In Section 5, we conclude the paper.

II. RELATED WORK

Sentiments can be extracted from reviews by utilizing knowledge-based techniques, statistical methods, or their hybrid combinations [3]. There is a large number of studies that use either recurrent or recursive neural network models for the sentiment classification task. However, only a few works employ ensembles of these models. There is also a body of literature which utilizes rhetorical structure for ABSA or which combines ontology-based approaches with deep learning.

In this section, we review these works in separate subsections based on the underlying neural models and structures to make them more readable. Since we evaluate the proposed approaches on SemEval-2014, Task 4 datasets, we first give an overview of the approaches used in this shared task. Before the review of the methods and the shared task approaches, we briefly touch on a few deep learning-based aspect extraction works and word representations used in sentiment analysis.

Aspects can be detected separately from sentiments or they can be learned in a joint model [4]. Poria *et al.* [5] extract aspects using a 7-layer deep convolutional neural network (CNN). They make use of features like word vectors and part-of-speech tags, and also a set of linguistic patterns. In [6], aspect representations are generated by capturing the semantic meaning of opinion targets. The authors use a long short-term memory (LSTM) model with an attention mechanism that incorporates syntactic information into the model as well. Another study based on an attentional LSTM network [7] extracts aspects and their sentiments by using both target-level and sentence-level attention. Commonsense knowledge is also incorporated into the system for the sentiment classification task.

There exist some studies that aim at building sentiment-aware word vectors and using these embeddings for sentiment analysis. A study [8] generates vectors for sentiment analysis using unsupervised and supervised methods. The unsupervised approach uses a combination of corpus-based and lexical-based features and applies singular value decomposition (SVD). The proposed approach is cross-domain and portable to other languages as well. In [9], sentiment knowledge is encoded into word embeddings using a CNN model and external sentiment lexica. Reference [10] uses emojis

[1] https://github.com/cemrifki/sentiment-recnn-rnn-ensemble-IARM

as features and builds a bi-LSTM (bi-directional LSTM) neural network to learn sentiment based word vectors. In [11], contextual information and supervised scores of words are taken into account to learn word embeddings. Another study employs a random walk algorithm in a semi-supervised manner to generate domain-specific polarities [12]. When vectors built by the SVD method are used rather than word2vec embeddings, they achieve better performance.

A. SemEval-2014, TASK 4 - ASPECT POLARITY DETECTION

SemEval-2014, Task 4 consists of four subtasks: aspect term extraction, aspect term polarity detection, aspect category extraction, and aspect category polarity detection. In this study, we focus on only aspect term polarity detection using the laptop and restaurant datasets published in the shared task [2]. 26 teams participated in this subtask. Among these, the two top ranking teams [13], [14] used a support vector machine (SVM) classifier. These studies both extract features, such as parse-trees and n-grams, and feed them into the classifier. Kiritchenko *et al.* [13] employ private sentiment lexica. One of these is an in-domain sentiment lexicon generated from the Amazon laptop reviews dataset. They also utilize out-of-domain sentiment lexicons, which are large-coverage tweet sentiment lexica and three manually curated sentiment lexica. Wagner *et al.* [14] employ public sentiment lexica. Some words are manually filtered out from the lexicons by the researchers. They also add some words manually to these lexica to adapt them into the laptop and restaurant domains. In our work, we rely on neural network models instead and not on hand-crafted features. We do not employ comprehensive feature engineering techniques. None of these 26 teams combine two types of neural networks as we do.

B. RECURRENT NEURAL NETWORK MODELS

In sentiment analysis, a large number of studies use recurrent neural networks due to their ability to model sequence data. As stated above, the sentiment of a word can affect those of the succeeding or preceding words. Two of the most widely used recurrent models are LSTM and GRUs [15].

The study we use as the baseline in this paper [1] performs aspect-based sentiment analysis using inter-aspect relations. The paper aims at finding the polarities of the given aspects and assumes that the sentiment of each aspect propagates through the text. For each aspect group, the average of aspect word embeddings is taken and a bi-directional GRU model is employed using these embeddings for each aspect group in the review. In our work, we use this recurrent model and enrich it with a recursive component for each aspect group. We, thereby, enhance the model by capturing the sentiment, syntactic, and additional semantic information contained in the reviews.

In [16], a recurrent attention mechanism is employed for aspect-based sentiment classification. The authors use position-weighted memory and recurrent attention memory to predict the sentiment of the target aspect. They thereby outperform the baseline studies by a significant margin for two domains. Arras *et al.* [17] extend the usage of layer-wise relevance propagation to recurrent neural networks. They apply a specific rule for propagation on connections of recurrent neural networks. They perform five-class sentiment classification and outperform the baseline gradient-based related approach. Another work [18] describes a model where aspect representations are merged with those of their contexts using each other's attention mechanisms in a recurrent network framework. In [19], sentiment classification is performed for Spanish tweets. They incorporate information from the sentiment lexicon into an LSTM model. In our work, we also make use of sentiment lexicon information, but in a different manner. Instead of appending sentiment scores from the lexicon to the word embeddings, we instead use this information only when we train our recursive models employing constituency and dependency parsers. Baziotis *et al.* [20] use a 2-layer bi-directional LSTM model for message-level sentiment detection. They add an attention mechanism on top of the last layer. They also employ a Siamese bi-directional model to detect topic-level sentiments. The system proposed ranked first in SemEval-2017, Task 4 *"Sentiment Analysis in Twitter."*

C. RECURSIVE NEURAL NETWORK MODELS

Different from recurrent networks, recursive neural networks can capture the syntactic structure or other types of relational structures within the text. They can be used, therefore, to generate more representative models of reviews in the sentiment analysis domain. In [21], a constituency parser is used to decompose a review into its constituent chunks using the Stanford sentiment treebank. Each node in the tree is given a sentiment label score, ranging from 0 (very negative) to 4 (very positive) in increments of 1. While training those trees, vectors at the nodes are updated with respect to sentiments. The trees model the sentiment structures of reviews in a finer way compared to the bag-of-words (BOW) or n-gram techniques, because they also take into consideration sentence components like conjunctions (*"however"*, *"although"*, etc.), while generating the parse trees. Such constructs can help shift the sentiment expressed in a subclause of the review. We use this approach in the constituency parser component in the proposed recursive model. In addition, we extract aspect term groups from the constituency parse trees in a novel way.

In contrast to the above-stated study that uses a constituency parser, [22] employs a dependency parser for the sentiment classification task. The proposed approach is especially fine-tuned for morphologically rich languages, such as Polish and Turkish. They carry out a three-class sentiment prediction task by training embeddings at each node in the dependency tree. A model described in [23] coextracts aspects and opinions. The authors employ both recursive neural network models and conditional random fields (CRF), and achieve state-of-the-art results. Their model leverages discriminative features and information is propagated between related opinions and aspects using a dependency parser. They also incorporate hand-crafted features into the framework to boost the performance. Another study [24] performs aspect-based sentiment analysis using the SemEval datasets. They combine constituency and dependency parsers by defining a subset of rules. They first convert dependency trees into phrase dependency trees. They then generate target-dependent binary phrase dependency trees. That is, they extract opinions from these original trees by training the model for target aspects. In our work, we also use constituency and dependency parsers, but we do not merge their outputs. We instead train and evaluate them separately. Also, our models extract aspect term groups and we define different and aspect-specific recursive submodules unlike them.

D. RHETORICAL STRUCTURE MODELS

While constituency and dependency parsers can capture the syntactic and semantic meanings of documents and encode this information in determining the overall sentiment, they can function only on a sentence basis. When a review consists of more than one sentence, rhetorical structures can generate the grammatical model more comprehensively and effectively. In these structures, the most important parts of text are defined as nuclei, whereas satellites contribute to the nuclei and are considered secondary. Hoogervorst *et al.* [25] use rhetorical structures in their study, in which polarity scores are assigned to words utilizing sentiment lexica. They weigh nucleus spans more heavily when determining the sentiment of the review and employ a genetic algorithm to find the optimal weights in this structure. Heerschop *et al.* [26] employ the ‘‘Sentence-level PArsing for DiscoursE’’ parser for computing rhetorical structure theory (RST) to perform sentiment analysis at document level. They state that some rhetorical relations have more importance in the sentiment classification task. They discuss that some of these relations (e.g., contrast relation) may contribute a negative weight, helping shift the overall sentiment of the document. RST is applied to sentiment classification in another study [27], in which polarity lexica are used and the propagation of sentiments across sentences and paragraphs is modeled. They make use of the ‘‘HIgh-Level Discourse Analyzer’’ in their study when carrying out the sentiment classification task at document level. Taboada *et al.* [28] determine the polarities of words using point mutual information via search engines. On- and off-topic sentences are extracted by employing an SVM classifier and these are subjected to a weighting scheme. They handle negation and take into account intensifiers and downtoners as well. They use a decision tree algorithm for sentiment classification.

E. ENSEMBLE MODELS

In ensemble approaches, two or more models are combined such that some of the models compensate for what the others lack. In [29], a CNN model over the word embedding layer is used. As the topmost layer, a recursive LSTM model relying on a constituency parser is utilized. The intuition behind this approach is that the CNN layer captures the context information per word and convolves it into a single vector. Then, these vectors are fed as input leaves into the recursive model. Feeding these embeddings into the constituency parser model's leaf nodes boosts the performance. The authors evaluate the methods on the Stanford Sentiment Treebank (SST). In our work, we instead first employ recursive neural network models to decompose a review into subreviews per aspect term. Then we feed the root embeddings of each subreview into the GRU model. In a study [30], a bi-directional LSTM model is applied before the CNN model is trained (R-CNN) and also in the other direction (C-RNN). They merge both of these architectures (i.e., R-CNN and C-RNN) with a technique named as ‘‘fusion gates.’’ Accordingly, both local contexts and temporal features are captured. In the work of Minaee *et al.* [31], LSTM and CNN models are trained separately and the average probability score of the outputs of these modules is computed. The value 0.5 is chosen as the cut-off threshold between negative and positive polarities. This is different to our work, where the output of the recursive network is fed as aspect sentiment embedding into the GRU model. Chen *et al.* [32] perform sequence prediction, by first applying a CNN model to capture the local information. Then they obtain a vector for each review and feed this embedding into every hidden node of the LSTM architecture per word. They claim this work can be expanded to be used for the sentiment classification task as well.

In addition to the ensemble models mentioned above, some works combine ontology based learning with deep learning approaches. In [33], conceptual values, which are sentiment value (positive and negative), aspect mention (e.g., ‘‘atmosphere’’ is linked to the concept ‘‘ambiance’’), and sentiment mention (e.g., ‘‘cheap’’ in ‘‘cheap price’’ has a positive connotation, whereas it expresses a negative sentiment in ‘‘cheap atmosphere’’) are taken into account. If these conceptual models do not produce enough sentiment information, bi-LSTMs are relied on. These capture the most indicative words in the left and right contexts of the target phrases. Another study [34] applies a similar approach using bi-GRU model. Meskele and Frasincar [35] enrich this hybrid approach by making use of CNN layers and regularization parameters.

We summarize the related works covered in this section in Table 1. We show only a subset of these studies in the table to make the reader get a general overview. Most of these works perform aspect-based sentiment analysis, which is also the topic in this paper. All of the studies given in the table evaluate their methods on datasets in English. We note that, although the performances of the works are included in the table, these ratios are not directly comparable since the domains are different.

III. METHODOLOGY

In this study, we perform ternary aspect-based sentiment classification, where the aspects can be positive, negative, or neutral. We first subject texts to basic tokenization and

TABLE 1. Summary of the related works. CML stands for classical machine learning algorithms.

Paper	Techniques	Domain	Performance (%)
Wagner *et al.* [14]	**CML:** n-gram, public lexica + SVM	Laptop	70.50
Kiritchenko *et al.* [13]	**CML:** n-gram, private lexica + SVM	Laptop	70.48
Taboada *et. al.* [28]	**RST:** Rhetorical structures + topic classifier + sentiment lexica	Movie	76.00
Hoogervorst *et. al.* [25]	**RST:** Rhetorical structures using sentiment lexica	Restaurant	60.00
Baziotis *et. al.* [20]	**Recurrent:** 2-layer LSTM and Siamese bi-directional LSTM	Twitter	86.30
Majumder *et al.* [1] (**baseline**)	**Recurrent:** Inter-aspect relations using LSTM	Restaurant	80.00
Wang *et al.* [23]	**Recursive** neural networks using CRF	Restaurant	84.11
Nguyen and Shirai [24]	**Recursive** neural networks combining constituency and dependency parsers	Restaurant	66.20
Van *et al.* [29]	**Ensemble** of recursive neural networks and CNN	SST	90.39
Yang *et al.* [30]	**Ensemble** of LSTM and CNN	SST	85.80
Meskele and Frasincar [34]	**Hybrid:** Lexicalized domain ontology + neural attention	Restaurant	86.31
Meskele and Frasincar [35]	**Hybrid:** Lexicalized domain ontology + a regularized neural attention model	Restaurant	87.10

other preprocessing operations (e.g., lowercasing) using the spaCy library [36]. We, thereafter, process the sentences by generating aspect term groups using both a dependency parser and a constituency parser. Lastly, we merge those two submodels into the novel proposed framework. In this section, we first explain the baseline recurrent model briefly. Then we describe the proposed recursive neural network model and the ensemble of these two models.

A. SUBMODEL 1: RECURRENT MODEL

In this model, we use the framework[2] developed by the study [1] that we use as baseline in this paper. This approach uses inter-aspect relations in such a way that aspects can modify the sentiments of the preceding or succeeding aspects. We do not change the hyperparameters of this model to perform a comparative analysis. We integrate this model into our recursive model in a novel way and obtain better performance for two domains.

We give a brief overview of the baseline study. As input, the GloVe embeddings [37] of the words in the text, including the aspect terms, are used. In the module aspect-aware sentiment representation (AASR), the context information is propagated across the text. In addition, an attention mechanism is used to determine and amplify the impacts of the sentiments which modify the relevant aspects. This process is repeated for every aspect term group so that its sentiment affects those of the others. As a final classification task, a softmax classifier is employed to determine the polarities of the given aspects. In the study, a mechanism referred to as multiple hops is also employed. Here, the hidden outputs of the processed sentence are fed again as input into the system several times, repeating the process. In this way, a finer representation of aspects is obtained. The visual summary of this approach is given in Figure 1.

B. SUBMODEL 2: RECURSIVE MODEL

Recursive neural networks are used to train models by taking the structures of texts into account. For instance, in the sentence *"I loved this movie!"*, the verb *"love"* expresses a positive sentiment and it directly modifies the object noun *"movie."* In this way, we can capture the sentiment and semantic information in the text more successfully and saliently compared to other models such as feedforward neural networks, n-grams, or BOW. In this section, we describe the constituency and dependency parser models developed in this work. These models are combined with the recurrent network model as will be explained in Section III-C.

1) CONSTITUENCY PARSER SUBMODEL

Constituency parsers build trees by breaking texts into phrases. In opinion mining, parsing texts into chunks might help organize sentiment information in a more structured way. For example, if a negation word (e.g., *"not"*) occurs in a phrase, it would shift the overall sentiment of that chunk. A similar mechanism applies when a contrastive conjunction (e.g., *"but"*) follows a phrase. Modeling these structures along with sentiment labels is reported to help obtain higher accuracies [21].

In this model, we use the Stanford CoreNLP framework [38] to parse texts into constituency chunks along with sentiment labels. An example showing the structure of this recursive neural tensor network is given in Figure 2. Here, at every node of the tree, there is a sentiment class that can be very negative (--), negative (−), neutral (0), positive (+), or very positive (++). Those sentiments are determined by the tool in a distant-supervised manner without relying on annotated sentiment labels in the training dataset. As mentioned, this model can capture the negation and scope relations in the parsed trees.

Before feeding these trees into our recursive neural network model, we extract aspect chunks. That is, for each aspect term, we generate a subtree including the aspect term and feed these subtrees separately into the model. The main intuition behind it is that, in aspect-based sentiment analysis, different sentiments may be expressed towards different aspects in the same review. Hence, splitting reviews into subreviews with respect to aspects can capture the relevant sentiments of the

[2] https://github.com/SenticNet/IARM

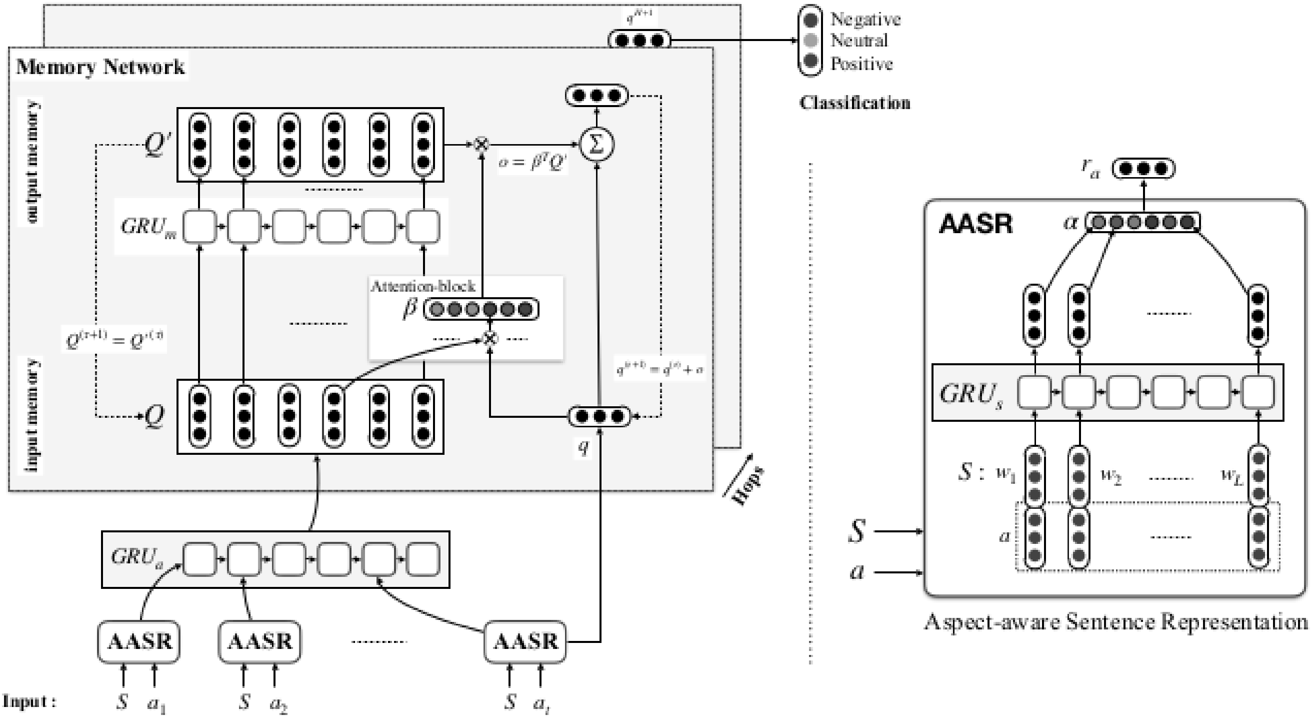

FIGURE 1. The architecture of the proposed approach in the baseline study. AASR stands for *Aspect-Aware Sentence Representation*. (Taken from [1].)

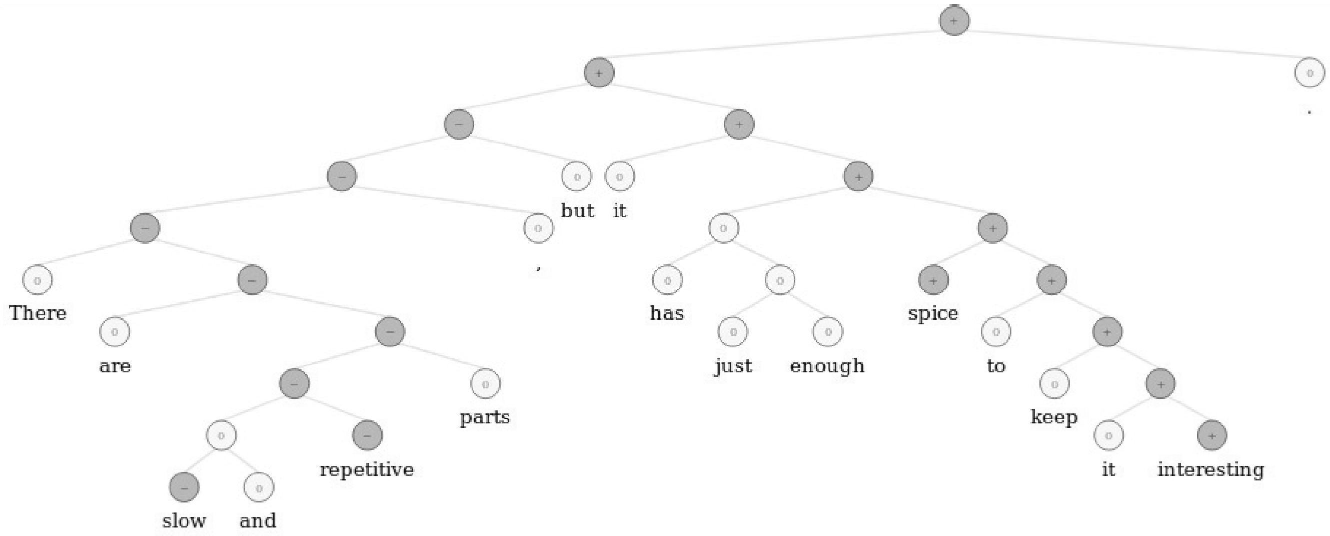

FIGURE 2. Example of a text parsed into its phrases along with their sentiment classes, ranging from very negative to very positive (--, −, 0, +, ++).

aspects and boost the performance for aspect-based opinion mining.

Our novel approach to generating a subtree per aspect term is as follows. We start scanning a tree from leaf terminals that correspond to aspect term words annotated in the training data and go upwards. When we encounter a node whose sentiment is not neutral, we assume that this word modifies the relevant leaf aspect term(s) and cut the tree at that point. That is, we take into account this node and all of its child nodes for the corresponding aspect term. This becomes the subtree representing the aspect term that lies at the bottom of it. Thus, it is possible that different aspects can be assigned to the same subreviews, as in the sentence *"I loved the ambiance and service overall."* In addition, we defined a few rules to handle the negation. For example, if a negator appears at a higher level than the node where we cut the tree as defined above, we keep expanding the subtree until the negator node and all of its children are included, and then cut the tree.

In general, this is the case that the negators are at the higher nodes than those words which are modified by them in constituency trees. Finally, as mentioned, after generating these subtrees, we train them as different subreviews in the classical recursive neural network model [21].

We obtain a higher performance than the baseline when splitting these reviews into subtrees for each aspect and training them separately for the ensemble model. However, there is a deficiency in this model. Aspects in the datasets used for ternary sentiment classification, such as the SemEval-2014 datasets we use for evaluation, have three gold sentiment labels, which are positive, neutral, and negative. However, as mentioned, we stop expanding the tree per aspect term in the bottom-up approach when we see a node of positive or negative polarity. That is, we ignore the fact that an aspect term can be neutral. Another point that should be noted related to the constituency model is that our rules sometimes fail in building meaningful subtrees. We attribute this to the general incompetence of the constituency parser which cannot perform as well as dependency parsers. Relations are much more clearly and robustly defined by dependency parsers in contrast to constituency parsers. When using dependency parsers, subreviews are generally connected to each other with specific relational features. Therefore, we are more able to distill them from the whole review. Constituency parser does not provide such capability.

After generating the subreviews using the constituency parser, we employ an open source recursive neural network framework [39] to train these separately. To combat overfitting, we use a validation set. We set the maximum epoch number as 30, since when we exceed this value, we always observe overfitting. We treat the embedding length as a hyperparameter and tried out the values of 30, 50, and 100. The vectors on the nodes encode the sentiment information at every node in the tree and are updated during the training phase. When the training phase is completed, we use the root embedding of each subreview per aspect in the ensemble model as will be explained in Section 3.3.

2) DEPENDENCY PARSER SUBMODEL

As stated above, constituency parsers can fail in generating sensible aspect subtrees to be used in the ensemble form. It is also reported in the literature that they, in general, perform worse than dependency parsers in the sentiment classification task [40]. Thus, we also make use of dependency parsers in this work to extract sentiment vectors for aspect terms and to leverage them later in our combined approach.

Dependency parsers connect words in a text by making use of the binary relationships between them. Vertices in the tree are words in the text and edges are labeled by the dependency relationships. The source of an edge is the parent modifying the child node. For example, in the sentence *"There are slow and repetitive parts."*, the words *"slow"* and *"repetitive"* modify the word *"parts."* This is visualized in Figure 3. Here, the relationship *"amod"* stands for adjectival modifier. In dependency trees, since a word can directly modify another word, models using these parsers can capture the sentiment information more accurately compared to constituency parsers. We use the spaCy library to decompose a text into its dependency relationships.

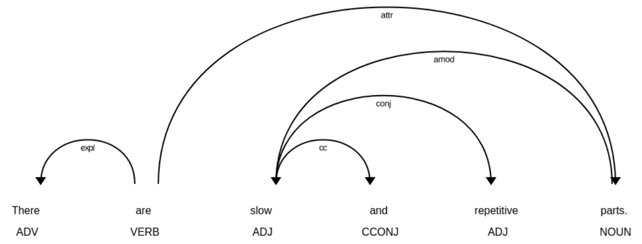

FIGURE 3. Example of a text decomposed into its relationships.

As in the case of constituency parsing, we first generate subreviews per aspect term using the dependency parser to later feed them into our ensemble model. For this purpose, we define a set of rules to decompose a dependency tree into subreview trees. If the tag of a word is verb (such as gerund, infinitive, or any type of verb) and the relationship attached to it is clausal component or conjunction, this marks the existence of a subreview. That is, all the children of a verb that are linked through relationships other than those two are recursively added to the corresponding subreview. For instance, in the sentence *"There are slow and repetitive parts, but it has just enough spice to keep it interesting."*, the word *"are"* is the main verb of the sentence. The second verb *"has"* is linked to it through the conjunction relationship. Therefore, we cut the dependency tree at this point. We merge all the children of the verb *"are"* except *"has"* (and its children) recursively and generate a subreview. We apply the same to the word *"has"* with respect to its children. The resulting subreviews are therefore *"There are slow and repetitive parts."* and *"But it has just enough spice to keep it interesting."*

After getting subcomponents per review, we filter out subtrees that do not contain aspects. Then we remove the preceding and succeeding redundant conjunctions (e.g., *"and"*), if any. We eliminate the punctuation marks at the beginning and end of the subreviews as well. We instead add the punctuation mark at the end of the whole review (e.g., *"!"*) to the end of every subreview to provide consistency. If no conjunctions or clausal components occur in the review, we assume that there are not any subreviews and the review consists of a single sentiment. Nevertheless, a single subreview (subtree) may contain more than one aspect. Accordingly, we obtain more meaningful subreviews following these rules compared to those produced by the constituency parser. An example that shows the subreviews generated by the above-stated constituency and dependency parser algorithms is given in Table 2. It can be seen that all the subreviews generated are related only to the aspects therein for the dependency parser algorithm. Also, only aspect-specific sentiments expressed towards them are included in the relevant subtrees.

TABLE 2. Sample reviews and the subreviews per aspect term generated using the dependency and constituency parsers. The underlined words are gold aspect terms.

Review	Subreviews generated	
	Dependency	Constituency (Bottom-up)
It may be a bit packed on weekends, but the vibe is good and it is the best French food you will find in the area.	**1.** It may be a bit packed on weekends. **2.** The vibe is good. **3.** It is the best French food you will find in the area.	**1.** Packed on weekends. **2.** The vibe is good. **3.** French food.
I love and I know gourmet food by excellence!	**1.** I love and I know gourmet food by excellence!	**1.** Gourmet food.
The vibe is very relaxed and cozy, service was great and the food was excellent!	**1.** The vibe is very relaxed and cozy! **2.** Service was great! **3.** The food was excellent!	**1.** The vibe is very relaxed and cozy! **2.** Service was great! **3.** The food was excellent!

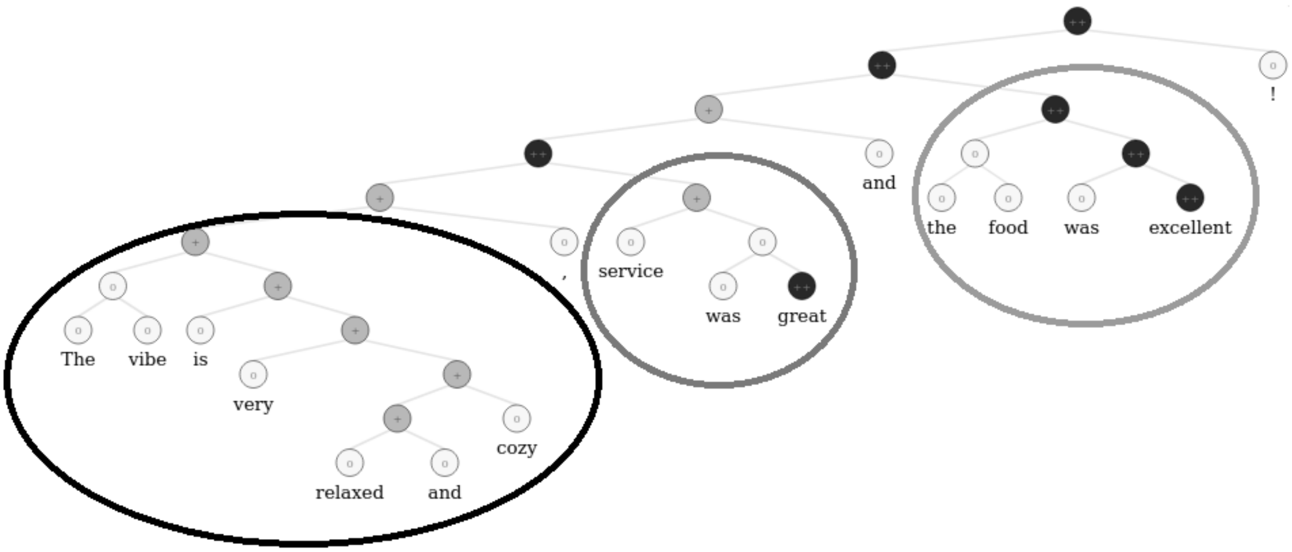

FIGURE 4. The text *"The vibe is very relaxed and cozy, service was great and the food was excellent!"* decomposed into its subreviews. Each encircled part of the figure is a subreview.

However, as for the constituency parser algorithm for decomposing a review into subreviews, examples shown in the table are not very consistent. For example, the words *"French"* and *"gourmet"* are considered positive by a polarity score of 3 by the Stanford NLP library. Therefore, the bottom-up approach for the constituency parser stops scanning the tree, when it encounters these non-neutral words. That is, the tree is cut at that point. The last review decomposed into its subreviews is visually shown in Figure 4. Each encircled part of the figure represents a subreview.

After extracting the subreviews within the reviews using the dependency parser, we train the sentiment model. In this neural model, each tree corresponding to a subreview is recursively trained separately. We use an open source code repository for this module [22]. We tweaked the parameters of this framework a bit to adapt it to our datasets. In this approach, as in the constituency parser method, there is a sentiment label at each node. We use a sentiment lexicon [12] in which the words are labeled as -1 (negative), 0 (neutral), or 1 (positive). If a corpus word does not appear in this lexicon, we assign the value 0 to it. We rely on this lexicon to assign sentiment scores to each node in the dependency parser tree. Although the labels used by the constituency parser module can belong to one of five sentiments, performing coarser analysis for the dependency parser provides us with a better performance in the final ensemble approach as compared to the model using the constituency parser. Using these labels, embeddings at the nodes are trained in a recursive way. We again test with different embedding sizes as 30, 50, and 100. We thereafter feed the trained root embeddings per aspect into our ensemble approach. We evaluated the performances of embeddings with varying sizes separately, as we empirically show in Section IV-C. As will be shown Section IV-C, the dependency parser model outperforms both the baseline and the constituency parser models by a significant margin.

In addition to this basic model, we also developed two variants of it that make use of the gold sentiment labels of the aspects in the nodes of the tree. In the first one, during the training phase, the aspect term nodes on the leaves are assigned the gold sentiment scores of the aspects, which can be −1, 0, or +1, rather than the sentiment values in the polarity lexicon. During the testing phase, for an aspect term in the text, the most frequently occurring sentiment label of the term in the training dataset is used. In the second variant, the sentiment label of the root node of the tree is changed as the gold sentiment label of the aspect in the tree. The intuition behind it is that the root embedding is one of the most effective factors in the sentiment classification task in our ensemble approach. During testing, a similar majority voting scheme is employed as defined above. However, these two variants of the basic dependency model led to a worse performance although we made use of the gold standard labels. We attribute this to the fact that the sentiment structure in the test is disarranged when lexicon and training labels are combined. For example, according to the sentiment lexicon, the polarity score of a word in the test dataset may be assigned the value 0. However, in the training data, the most frequently occurring sentiment label might be positive. Therefore, the score of +1 could be taken into account for that word in the test dataset. That is, if we utilized all the sentiment scores relying on the same source, which is polarity lexicon, results are expected to be more consistent and it would improve the performance. We observe this as will be explained in Section IV-C.

C. ENSEMBLE MODEL COMBINING RECURRENT AND RECURSIVE NETWORKS

The ensemble model combines the recurrent and recursive models in a hierarchical way such that the output of the recursive network is fed as input to the recurrent network. As stated in Section III-A, the recurrent model is the same as the baseline model [1]. On the other hand, the recursive model is based on a novel approach that makes use of subreviews and employs both constituency and dependency parsers, separately. In this framework, the recurrent model captures the temporal information by revealing the interplay of polarities. The recursive model takes into consideration the grammatical, semantic, and syntactic structure of the text at a finer scope. Each of these models compensates for what the other lacks. Hence, merging these neural models may give us more information about the text and its sentiments expressed towards its aspects. The proposed ensemble model is shown in Figure 5. After the subreviews are extracted and trained by the recursive model, the root embedding (hidden state at the root node) obtained for each subreview (aspect) is fed into the recurrent model. In the input of the recurrent network, for each word, the word embedding and the corresponding aspect and root embeddings are concatenated. We use the GloVe vectors for the words and the aspect terms. If an aspect is composed of two or more words, we take into account their average vector.

The reason to utilize root embeddings in the ensemble form is that the root represents the whole subreview and thus can capture the sentiment and semantic information more comprehensively than the intermediate nodes. Propagation within the recursive neural network is affected mostly by the polarity of the root word. When we compare the proposed model with the baseline model, we note that the proposed approach can encode the relevant sentiment information of the aspects and incorporates it into the recurrent model using inter-aspect relationships in a novel way.

TABLE 3. Details of the datasets used in this work.

Domain		Positive	Neutral	Negative
Restaurant dataset	Training	2,159	632	800
	Testing	730	196	195
Laptop dataset	Training	980	454	858
	Testing	340	171	128

IV. EXPERIMENTAL EVALUATION

In this section, we first describe the datasets used for the aspect-based ternary sentiment classification task. We then mention the hyperparameters we have chosen. Lastly, we explain and discuss our experimental settings and results.

A. DATASETS

In this study, we evaluate our approaches on the two datasets of the SemEval-2014 Task 4 competition.[3] These two corpora are composed of laptop and restaurant reviews. The data were curated for aspect-based sentiment analysis in the sense that the task is to determine the sentiment labels of the given aspects. Each aspect has a positive, neutral, or negative gold label. We summarize the datasets we used in Table 3. The figures in the table denote the numbers of aspects of different polarities in the corresponding datasets.

Since the datasets were already split into training and test datasets, we did not perform cross-validation. In order to combat overfitting in our recursive, recurrent, and ensemble methods and to choose the optimal hyperparameter values, we used 10% of our training dataset as the validation dataset.

B. HYPERPARAMETERS

We utilized three open source frameworks. The sets of hyperparameters used in these models are shown in Table 4. Two of these are recursive neural networks training models separately on the outputs of the constituency and dependency parsers. These neural models train the hidden state embeddings at each node of these recursive trees. The third one is the recurrent neural network model, which is also used as the baseline by itself in this study. We tested these models with different sizes for hidden state vectors as 30, 50, and 100. We obtained the optimal embedding size by using a validation set and fed it as input into our ensemble approach. When we choose the value 100 as the size of hidden embeddings, we observe overfitting for both domains. For the dependency and recurrent neural models, we feed the GloVe word embeddings of length 300 at the bottom-most layer. However, embeddings at intermediate nodes in the trees can be of varying sizes. For the constituency parser model, we learn leaf

[3] http://alt.qcri.org/semeval2014/task4

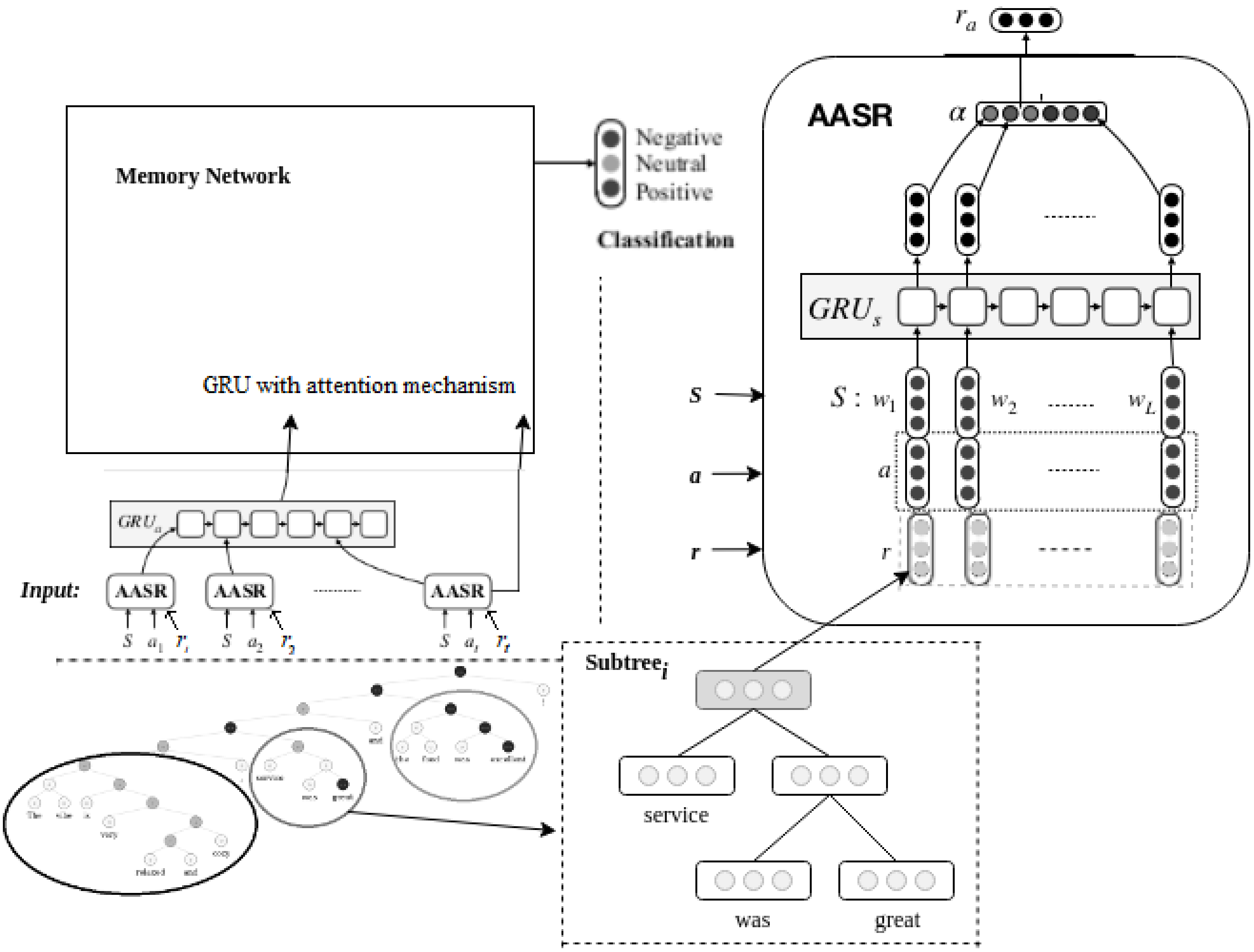

FIGURE 5. Visual summary of the ensemble framework. Our contribution is that, for each aspect term, we also employ the relevant root vector feature of the recursive trees, denoted by *r*, in addition to the baseline method. Memory network module is the same as that in the baseline study. In the given example, *Subtree$_i$* is the second subreview of the comment shown on the bottom left (minimized version of Figure 4). In this example, there are three AASR modules being run since there are three subreviews.

TABLE 4. Hyperparameters in the recursive neural network. *"Optimal size"* denotes the optimal size of embeddings at intermediary nodes in trees.

Hyperparameter		Constituency parser model	Dependency parser model	Recurrent model
Learning rate		0.001	0.05	0.001
Maximum number of epochs		30	30	30
Batch size		20	25	30
Optimal size	Restaurant	50	50	-
	Laptop	30	30	-

TABLE 5. Hyperparameters in the baseline recurrent model.

Hyperparameter	Laptop	Restaurant
Hop Count	10	3
GRU Hidden State Size	400	300
GRU Output Size	400	350

embeddings from scratch, since the corresponding framework does not allow us to employ pretrained vectors. We also show the hyperparameters used in the baseline recurrent model in Table 5.

C. RESULTS

Table 6 shows the results of the experiments for the two domains in terms of accuracy. The baseline recurrent model is given in the first row of the table. The last two rows are the ensemble forms where the embeddings trained on the recursive model are fed into the recurrent model. The training is performed on the constituency parser output in the first one, while it is performed on the dependency parser output in the second one. The three approaches explained in Section III-B2 are labeled as "root", "leaves", and "gold aspect." The "root" scheme is the basic model where the root vector of a subreview is given as input to the recurrent model. In the "leaves" scheme, the embedding vector of an aspect term corresponding to that aspect is used instead of the root of the tree. If an aspect consists of more than one word, we take

TABLE 6. Accuracies (%) of the baseline approach, the RST method, and the ensemble framework which combines the baseline recurrent submodel with different recursive models and embedding sizes.

Model	Feature	Recursive NN vector size	Accuracy (%) Restaurant	Accuracy (%) Laptop
Baseline (IARM) [1]			80.00	73.80
Constituency Parser + Recurrent NN	Leaves	30	77.98	72.23
		50	79.20	72.10
		100	78.10	72.00
	Root	30	79.48	**74.20**
		50	**80.27**	73.73
		100	78.10	72.97
Dependency Parser + Recurrent NN	Gold aspect label	30	78.30	72.90
		50	78.57	72.85
		100	77.90	72.48
	Root	30	79.90	**76.15**
		50	**80.90**	75.75
		100	79.82	73.70

its average and feed this embedding to the system. Lastly, the scheme named as "gold aspect" shown in the table is the model where we leverage the labels of aspects (i.e. positive, negative, or neutral) given in the training data for the dependency parser model. That is, for the training dataset, the label of the root in a subreview is assigned the gold sentiment label in the dependency parser tree. In addition, we define the gold sentiment label only for the corresponding aspect node (leaf) in the tree, not for the root. For the aspects in the test dataset, we take into account the most frequently occurring sentiment for these terms in the training set and change that node's sentiment accordingly in the tree. The other labels of the tree are determined by the sentiment lexicon. However, we do not include this last scheme employing gold sentiment labels for leaves in the table since this scheme does not perform better than the baseline either. These feature engineering techniques were explained in more detail in Section III-B2. The recursive neural network vector size denotes the embedding size at the nodes in the parse trees. We only examine the performance with respect to different recursive network vector sizes, not with varying recurrent network vector sizes. Its reason is that the recurrent model is the baseline study and we want to show how much we can boost the ternary classification performance without changing any of the parameters of this submodel in the ensemble framework.

The results show that the proposed ensemble approach outperforms the baseline model (IARM) for both domains when the root embeddings are used. This is the case with both the constituency parser and the dependency parser. With these results, we outperformed all the teams (26 teams) that participated in the aspect polarity detection task, SemEval-2014 for the laptop domain. As for the restaurant reviews, we rank second overall. We can construe this result as follows. The two top-ranking teams in the task use SVM classifiers by employing hand-crafted features and other feature engineering techniques. We, instead, develop an ensemble form of two deep neural network models. It is well stated in the literature that classical machine learning methods in general cannot compete with deep neural networks successfully [15].

When we compare the two parser models, we see that training the recursive network with the dependency parser yields better results than the constituency parser. This can be attributed to the property that dependency tree captures the modifier relationships directly unlike the constituency tree. A comparison between the three schemes used in feeding input to the recurrent network shows that the root form gives the best performance. In the case of constituency parser, root embeddings capture wider information related to the sentiment of the relevant aspects than the leaf embeddings. In the case of dependency parser, we think that the reason for this situation is that incorporating gold aspect labels into the trees makes the model inconsistent with the sentiments of other words whose polarities are determined by a sentiment lexicon. That is, using polarities of the opinion lexicon and training data labels alike can make the sentiments of words ambiguous and contradictory. For example, a word's label can be 0 (neutral) in the training dataset, whereas it can be +1 (positive) in the sentiment lexicon. Finally, we note that the recursive network root embedding vectors with sizes 30 and 50 give the best results in the two domains, while embedding size of 100 causes overfitting.

We also show in Table 7 how well the proposed ensemble model performs compared to the baseline when a single aspect (SA) or multiple aspects (MA) appear in reviews. We include the best results (dependency parser and the *"root"* scheme) for our ensemble approach in the table. SA and MA correspond to the cases where we take into account only the reviews having, respectively, a single aspect and more than one aspect. Every review in the datasets has either an SA or MA. We include these different scenarios to see that the interplay of sentiments between different aspects has an impact on the performance. Using reviews with multiple aspects give better success rates compared to using single-aspect reviews. The table also shows that the proposed approach outperforms the baseline approach for the two datasets in both scenarios. Our results prove to be statistically significant at $p = 0.05$ when we employ the Stuart-Maxwell test.

TABLE 7. Accuracies (%) observed for the baseline approach and our ensemble framework. SA refers to single aspect scenario, MA to multiple aspect scenario.

Model	Restaurant		Laptop	
	SA	MA	SA	MA
Baseline (IARM)	78.6	80.48	73.4	74.1
Our ensemble method	**79.5**	**81.38**	**75.75**	**76.45**

As a summary, our results indicate that incorporating a recursive network into the baseline recurrent model enriches the model by sentiment vectors relevant to the aspects. These vectors are trained via recursive neural models separately for the generated subreviews. In this way, combining grammatical (syntactical) and temporal neural models provide us with more relevant polarities for the given aspects and a better performance.

V. CONCLUSION AND FUTURE WORK

In this study, we built a framework for ternary aspect-based sentiment classification. The proposed ensemble approach is formed of a recurrent model which is also used as the baseline and a recursive model. The reviews are first divided into subreviews such that each subreview in general includes a single aspect. In this way, we hold only the relevant information and sentiment in each subreview containing the corresponding aspect term(s). The recursive model is trained using the constituency and dependency parsers of the subreviews. The root vectors of these trees are then fed into the recurrent model by concatenating them with the sentence's word and aspect embeddings.

We observed that combining recurrent and recursive neural networks provides a more comprehensive and a robust model. Recurrent neural networks capture the temporal information. However, they cannot represent the grammatical structure of texts. The intuition behind combining these models is that it captures the relevant sentiment, syntactic, and semantic information within the subreviews and thus enriches the baseline model. In this way, the recurrent and recursive approaches can model the information that the other lacks. The experiments on two datasets outperformed the baseline approach by a significant margin. When we used the dependency parser for the recursive model in the ensemble form, we achieved the best results. This indicates that dependency parsers can capture the information about which sentiment word modifies which other words more successfully compared to constituency parsers. We think that our ensemble classifier model can also be applied to other NLP tasks with minor changes.

As future work, we plan to extend our framework by (1) improving the constituency parser-based model to adapt it to capture the neutral sentiments as well, (2) incorporating a CNN model to enhance the contextual representation of aspects, (3) using sentiment lexica or semi-supervised techniques adapted to the domain to better model the sentiment components of words, (4) adapting our approach to other languages, and (5) training the recursive and recurrent models in the ensemble framework jointly to arrive at better results. We also plan to (6) evaluate our methods on other corpora (e.g., the SemEval-2015 datasets) where reviews may consist of more than one sentence and (7) compare our performances to those of RST studies and the hybrid solutions which combine ontology-based reasoning with deep learning models.

REFERENCES

[1] N. Majumder, S. Poria, A. Gelbukh, M. S. Akhtar, E. Cambria, and A. Ekbal, "IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Brussels, Belgium, 2018, pp. 3402–3411.

[2] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 27–35.

[3] E. Cambira, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, vol. 31, no. 2, pp. 102–107, Mar. 2016.

[4] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.

[5] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl.-Based Syst.*, vol. 108, pp. 42–49, Sep. 2016.

[6] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "Effective attention modeling for aspect-Level sentiment classification," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, Santa Fe, NM, USA, 2018, pp. 1121–1131.

[7] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. AAAI-*, New Orleans, LA, USA, 2018, pp. 5876–5883.

[8] C. R. Aydın, T. Güngör, and A. Erkan, "Generating word and document embeddings for sentiment analysis," presented at the 20th Int. Conf. Comput. Linguistics Intell. Text Process. (CICLing), La Rochelle, France, Apr. 2019.

[9] Z. Ye, F. Li, and T. Baldwin, "Encoding sentiment information into word vectors for sentiment analysis," in *Proc. COLING*, Santa Fe, NM, USA, 2018, pp. 997–1007.

[10] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 1615–1625.

[11] D. Tang, F. Wei, B. Qin, T. Liu, and M. Zhou, "Coooolll: A deep learning system for Twitter sentiment classification," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 208–212.

[12] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, 2016, pp. 595–605.

[13] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "NRC-Canada-2014: Detecting aspects and sentiment in customer reviews," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, Dublin, Ireland, 2014, pp. 437–442.

[14] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, and L. Tounsi, "DCU: Aspect-based polarity classification for SemEval task 4," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 223–229.

[15] Y. Goldberg and G. Hirst, "Concrete recurrent neural network architectures," in *Neural Network Methods in Natural Language Processing*, G. Hirst, Ed. San Rafael, CA, USA: Morgan & Claypool, 2017, pp. 177–183.

[16] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Copenhagen, Denmark, 2017, pp. 452–461.

[17] L. Arras, G. Montavon, K.-R. Müller, and W. Samek, "Explaining recurrent neural network predictions in sentiment analysis," in *Proc. 8th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal.*, Copenhagen, Denmark, 2017, pp. 159–168.

[18] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Melbourne, Australia, Aug. 2017, pp. 4068–4074.

[19] O. Araque, R. Barbado, J. F. Sánchez-Rada, and C. A. Iglesias, "Applying recurrent neural networks to sentiment analysis of Spanish tweets," in *Proc. TASS Workshop Semantic Anal. SEPLN*, Murcia, Spain, 2017, pp. 71–76.

[20] C. Baziotis, N. Pelekis, and C. Doulkeridis, "DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, Vancouver, BC, Canada, Aug. 2017, pp. 747–754.

[21] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. EMNLP*, Seattle, WA, USA, Oct. 2013, pp. 1631–1642.

[22] T. Korbak and P. Zak, "Fine-tuning tree-LSTM for phrase-level sentiment classification on a Polish dependency treebank. Submission to PolEval task 2," in *Proc. LTC*, Poznan, Poland, 2017, pp. 1–5.

[23] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Austin, TX, USA, 2016, pp. 616–626.

[24] T. H. Nguyen and K. Shirai, "PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, 2015, pp. 2509–2514.

[25] R. Hoogervorst, E. Essink, W. Jansen, M. Helder, K. Schouten, F. Frasincar, and M. Taboada, "Aspect-based sentiment analysis on the Web using rhetorical structure theory," in *Proc. ICWE*, Lugano, Switzerland, 2016, pp. 317–334.

[26] B. Heerschop, F. Goossen, A. Hogenboom, F. Frasincar, U. Kaymak, and F. de Jong, "Polarity analysis of texts using discourse structure," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. CIKM*, Glasgow, Scotland, UK, 2011, pp. 1061–1070.

[27] A. Hogenboom, F. Frasincar, F. de Jong, and U. Kaymak, "Using rhetorical structure in sentiment analysis," *Commun. ACM*, vol. 58, no. 7, pp. 69–77, Jun. 2015.

[28] M. Taboada, K. Voll, and J. Brooke, "Extracting sentiment as a function of discourse structure and topicality," Ph.D. dissertation, School Comput. Sci., SFU, Burnaby, BC, Canada, 2008.

[29] V. D. Van, T. Thai, and M.-Q. Nghiem, "Combining convolution and recursive neural networks for sentiment analysis," in *Proc. 8th Int. Symp. Inf. Commun. Technol. SoICT*, 2017, pp. 151–158.

[30] F. Yang, C. Du, and L. Huang, "Ensemble sentiment analysis method based on R-CNN and C-RNN with fusion gate," *Int. J. Comput. Commun. Control*, vol. 14, no. 2, pp. 272–285, 2019.

[31] S. Minaee, E. Azimi, and A. Abdolrashidi, "Deep-sentiment: Sentiment analysis using ensemble of CNN and bi-LSTM models," Apr. 2019, *arXiv:1904.04206*. [Online]. Available: https://arxiv.org/abs/1904.04206

[32] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, "Ensemble application of convolutional and recurrent neural networks for multi-label text categorization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, AK, USA, May 2017, pp. 2377–2383.

[33] O. Wallaart and F. Frasincar, "A hybrid approach for aspect-based sentiment analysis using a lexicalized domain ontology and attentional neural models," in *Proc. ESWC*, Portorož, Slovenia, 2019, pp. 363–378.

[34] D. Meškelė and F. Frasincar, "ALDONA: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalised domain ontology and a neural attention model," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 2489–2496.

[35] D. Meskele and F. Frasincar, "ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model," *IEEE Intell. Syst.*, vol. 57, no. 3, pp. 1–9, Feb. 2020.

[36] M. Honnibal and M. Johnson, "An improved non-monotonic transition system for dependency parsing," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Lisbon, Portugal, 2015, pp. 1373–1378.

[37] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, 2014, pp. 1532–1543.

[38] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford CoreNLP natural language processing toolkit," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2014, pp. 55–60.

[39] Y. Cheng. (2018). *Recursive Neural Network for Sentiment Analysis With Pytorch*. [Online]. Available: https://github.com/yc930401/RecNN-pytorch

[40] L. Dong, F. Wei, S. Liu, M. Zhou, and K. Xu, "A statistical parsing framework for sentiment classification," *Comput. Linguistics*, vol. 41, no. 2, pp. 293–336, Jun. 2015.

CEM RIFKI AYDIN received the B.Sc. degree from the Department of Computer Engineering, Bahçeşehir University, and the M.Sc. degree from the Department of Computer Engineering, Boğaziçi University, where he is currently pursuing the Ph.D. degree. He has been supported by the TÜBİTAK BIDEB 2211 Scholarship during his Ph.D. studies. He is a Researcher at the TETAM Research Center to complete his doctoral research. He has participated in several NLP projects and published several scientific articles. His research interests include natural language processing, machine learning, deep learning, and sentiment analysis.

TUNGA GÜNGÖR received the M.S. and Ph.D. degrees from the Department of Computer Engineering, Boğaziçi University.

He was a Visiting Professor at the Center for Language and Speech Technologies and Applications, Universitat Politecnica de Catalunya, Barcelona, Spain, from 2011 to 2012. He is a Senior Lecturer and a Researcher with the Department of Computer Engineering, Boğaziçi University. He is a member of the Artificial Intelligent Laboratory and the Text Analytics and Bioinformatics Laboratory at the department. He teaches undergraduate and graduate level courses on the topics of artificial intelligence, natural language processing, machine translation, and algorithm analysis. He participated as the Project Leader in projects about developing an adaptive question answering system for primary and secondary education students, developing concept mining methods for document analysis, developing a hand-written recognition system using a large lexicon, morphology-based language modeling for speech recognition, and developing structure-preserving and query-biased automated summarization methods. The projects were funded by the Turkish Scientific and Technological Research Council of Turkey and the national funds. He has published about 90 scientific articles, and participated in several research projects and conference organizations. His research interests include natural language processing, machine translation, machine learning, and pattern recognition.