Latent Syntactic Structure-Based Sentiment Analysis

Viktor Hangya, Zsolt Szántó and Richárd Farkas
Department of Computer Algorithms and Artificial Intelligence
University of Szeged
Árpád tér 2., Szeged, H6720
Email: {hangyav, szantozs, rfarkas}@inf.u-szeged.hu

***Abstract*—People share their opinions about things like products, movies and services using social media channels. The analysis of these textual contents for sentiments is a gold mine for marketing experts, thus automatic sentiment analysis is a popular area of applied artificial intelligence. We propose a latent syntactic structure-based approach for sentiment analysis which requires only sentence-level polarity labels for training. Our experiments on three domains (movie, IT products, restaurant) show that a sentiment analyzer that exploits syntactic parses and has access only to sentence-level polarity annotation for in-domain sentences can outperform state-of-the-art models that were trained on out-domain parse trees with sentiment annotation for each node of the trees. In practice, millions of sentence-level polarity annotations are usually available for a particular domain thus our approach is applicable for training a sentiment analyzer for a new domain while it can exploit the syntactic structure of sentences as well.**

Keywords-sentiment analysis; syntax parsing; text classification

I. INTRODUCTION

People publicly share their opinion using social media on a variety of topics, like products and political issues. The task of sentiment analysis (SA) is to automatically extract opinions from textual contents. Most of the SA systems assign polarity labels (e.g. positive, negative and neutral) to textual elements like documents and sentences. The basic solution for SA is to represent the texts in a bag-of-word model and train supervised classifiers or/and employ polarity lexicons for polarity classification [1].

Recent studies have been investigating the utilization opportunities of the syntactic structure of the sentences for enhancing sentiment analyzers. Most of these proposals use hand-crafted rules based on the syntactic parse of the sentence [2]. These rules are engineered to address certain restricted set of in-sentence SA's challenges, like negation and intensification.

In the Stanford Sentiment Treebank [3] a polarity label was manually assigned to each constituent of the sentence's phrase structure parse. This treebank can be utilized as a training dataset for statistical structure prediction methods and it introduces the opportunity of exploiting the syntactic structure of sentences without restricting the models to a closed set of language phenomena (like negation and intensifiers), neither demands the direct modeling of those phenomena. It enables the application of supervised machine learning techniques to model how morphosyntactic and lexical structures alter the polarity of a constituent. On the other hand, the supervised approach has the disadvantage of requiring a manually annotated treebank. This treebank is domain-dependent, i.e. sentiment analyzers trained on it work fine only on movie reviews and the annotation of new treebanks for other domains is expensive.

In this work, we focus on the exploitation strategies of syntactic structures for in-sentence and sentence-level SA. Usually, sentence-level polarity labels can be easily obtained in a huge amount for various domains, take for instance pro/con or bottom-line summaries of the product review sites. Hence, we propose a machine learning framework for sentence-level and in-sentence polarity classifiers by using exclusively sentence-level polarity annotation for training. Our approach can predict the sentiment labels assigned to the constituents of a phrase structure parse tree without an annotated sentiment treebank by handling the polarity labels of internal nodes in parse trees as latent variables. Figure 1. exemplifies the difference between a fully annotated sentiment tree and our latent representation.

We shall introduce two experimental setups for the investigation of the proposed approach. The objective of our experiments' first batch (in Section IV) is to investigate whether the sentence-internal latent structure helps the prediction of sentence-level polarity. The second batch of experiments (in Section V) shall show that the sentence-internal latent structures themselves are also meaningful when we extract features from them for a target-oriented sentiment analysis task.

The chief contribution of this work is to propose a latent syntactic structure-based approach which requires only sentence-level polarity labels for training. Our experiments on three domains (movie, IT products, restaurant) support that sentiment analyzers are domain-dependent. We shall show that a sentiment analyzer that exploits syntactic parsing and has access only to sentence level polarity annotation for in-domain sentences can outperform state-of-the-art models that were trained on out-domain parse trees with sentiment annotation for each node. In practice, millions of sentence-level polarity annotations are usually available for a particular domain and community while there

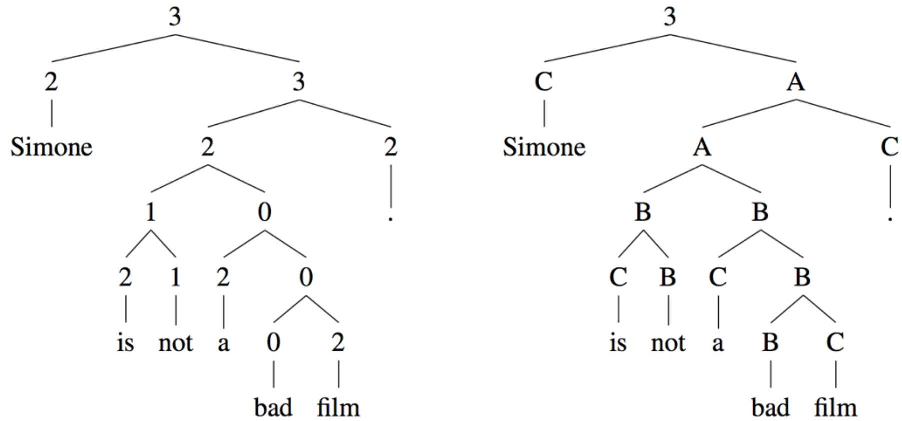

Figure 1. Representation of sentiment trees in the Stanford Sentiment Treebank [3] (left) contains 5-level polarity annotation {0=very negative, 4=very positive} for each node of the binary syntactic tree. On the other, we assume that we have access only sentence-level polarity annotation, i.e. only the label of the root is given (right). Here, the states of the inner nodes are described by latent discrete variables {A,B,C}.

is only a single treebank annotated for polarity, which consists of sentences from the movie review domain.

II. Related Work

SA has became an actively researched area due to the fact that a huge amount of data is available on the Internet [4]. In the early stages most of the systems were based on supervised machine learning techniques using bag-of-words representation, see [1] for a survey on SA. Several shared tasks were organized to promote research in SA. For instance, the goal of the SemEval-2014 Task 9 – Sentiment Analysis in Twitter [5] was to classify short messages into polarity classes. Most of the participating systems were based on supervised machine learning techniques. Besides the standard bag-of-words representation, various lexical resources [6, 7] were also employed in order to improve the performance of these systems. A drawback of these systems is that they cannot exploit the syntactic and semantic structure of texts, i.e. negations, intensifiers, discourse relations, etc.

Recently, several studies have been published on exploiting syntactic parsers for SA. For instance, in [2] a dependency parser was employed in order to detect intensifications and negations. They used hand crafted rules over dependency parses and lists of intensifiers and negation words respectively. The relation between text fragments can influence the polarity of a document as well. In [8] a joint model for unsupervised induction of sentiment, aspect and discourse information was proposed. They showed that performance can be improved by incorporating latent discourse relations (but, and, when, etc.) in the model.

In the Stanford Sentiment Treebank [3] a polarity label was manually assigned to each constituent of the sentence's phrase structure parse and they introduced a Recursive Neural Tensor Network-based procedure to capture the compositional effects of the sentences. Although this approach provides a more free representation for in-sentence SA, it has the disadvantage of requiring a manually annotated treebank. To the best of our knowledge, the Stanford Sentiment Treebank is the only available sentiment-annotated treebank. This treebank is domain-dependent and the annotation of new treebanks for other domains is expensive. Our proposal is related to [3] as we also start from syntactically parsed sentences but we are handling the polarity labels of internal nodes as latent variables. This way the inputs for training our system are texts annotated only on the sentence level.

A SA approach which is based on in-sentence structures was also introduced in [9]. They also propose a system which can learn in-sentence sentiment structures using exclusively sentence-level annotation. On the other hand, their system contains several hand-crafted assumptions and rules (e.g. they handle negations and intensifiers by dedicated rules) while our latent representation introduces the opportunity of exploiting the syntactic structure of sentences without restricting the models to a closed set of language phenomena, neither demands the direct modeling of those phenomena. Another difference is that we use a syntactic parser to provide the in-sentence structure while they use a CYK sentiment parser. Although their approach provides an opportunity of learning also the structure itself the running time is cubic hence it is not feasible to train on several hundred thousand sentences.

Sentiment analysis can be applied at different levels depending on the depth of target information [10]. The aim of the so called target-oriented SA is to classify sentiments which are related to a given target [11]. In this case extracting bag-of-word features is not enough. The aim of the SemEval-2014 Task 4 – Aspect Based Sentiment Analysis [12] shared task was to compare systems on SA tasks. Many participated systems used syntactic parsers to identify text parts which are related to the target phrase in question. One of the novelties of our paper is that we also experimented with target-level SA and we shall show that the induced latent sentiment trees has a considerable added value in a target-level SA system.

Our approach is also related to semantic parsing. For instance, in [13] latent temporal types were used in a latent CFG to learn temporal expressions. This semantic parsing problem is very similar to ours, both methods require a sentence level label in the training phase and use latent variables in the non-terminals (except the root). They assigned temporal types to non-terminal nodes, in contrast, we have polarity labels in these nodes. They employed an EM-style bootstrapping approach for training the models.

We used structured perceptron for machine learning which is successfully applied for structured prediction with latent variables in other areas of natural language processing. In [14] a system similar to ours was introduced for coreference resolution with latent tree structures of mention clusters. They used the passive-aggressive algorithm for this training task and updated against the highest scored tree with correct clustering of mentions.

We used structured perceptron for machine learning which is successfully applied for structured prediction with latent variables in other areas of natural language processing. In [14] a system similar to ours was introduced for coreference resolution with latent tree structures of mention clusters. They used the passive-aggressive algorithm for this training task and updated against the highest scored tree with correct clustering of mentions.

III. LATENT SYNTACTIC STRUCTURE-BASED SA

We propose a procedure for predicting the polarity label for each constituent of a sentence's phrase structure parse and we assume that we have access to exclusively sentence-level polarity labels during training.

A. Preprocessing

Sentences are tokenized by the Stanford CoreNLP toolkit [15]. The syntactic structure of a sentence is fixed, i.e. we syntactically parse each sentence in a preprocessing step. We employed the BerkeleyParser [16], a state-of-the-art phrase structure parser, with the English 6th iteration model. We used right-branch binarized and unlabeled – both POS tags and internal node labels are deleted – syntactic parse trees for sentiment parsing.

B. Latent State Representation

We assume that the system has access to sentence-level binary polarity (positive and negative) annotations, which serve as the label of the syntactic parse tree's root node. A latent discrete random variable is assigned to each internal node of the parse tree. In our experiments, we use latent variables with three possible states {A,B,C}. Although the number of possible states can be easily changed, we postpone the investigation on the effect of different state space sizes for future research.

C. Decoder

We use a structured perceptron to decode the labels for the nodes of the syntactic parse tree. The decoder iterates through the tree in a bottom-up order and employs only local features which are described in Section III.E. Preliminarily we experimented with non-local features along with a beam-search decoder but their improvement was not considerable while running times increased exponentially.

The decoder selects the top scored derivation with latent variables {A, B, C} at the internal nodes and polarity labels {positive, negative} for the root node. It is an exact search, i.e. the derivation space is complete, we do not filter the possible derivations. The branching factor of the derivation space is 9 as we work with binary parse trees and 3 possible states.

We use the hypergraph representation of the derivation space along with the Viterbi decoder from the Joshua software package[1].

D. Training

At training time, only the root label is available for the preprocessed sentences and the in-sentence polarity labels at the nodes of the parse tree is handled to be latent variables. We follow an Expectation-Maximization (EM) approach for training the structured perceptron. In the E-step, we select the top scoring derivation of the gold standard root label. In this way, we update the weights of the perceptron against a latent sentiment tree which is easily learnable by the structured perceptron [14].

In the M-step, the update rule of the averaged perceptron is employed [17]. The learning rate parameter was set to 0.1 and the batch update size to 30. These parameters were set based on a grid search metaparameter optimization on a development dataset. We run 15 epochs of training in each of our experiments and we use our in-house implementation for training.

E. Features

We implemented three feature templates to extract information from derivation candidates. We use only local features, i.e. which can be extracted from the 1-level subtree of the derivation (see Figure 2). The bottom-up decoder extracts the new features for a new node of the derivation and adds them to the feature vectors of the two daughters' feature vector.

1 Joshua is a JAVA package available at joshua-decoder.org

We note that our main objective was to investigate whether our latent representation can improve in-sentence SA. There

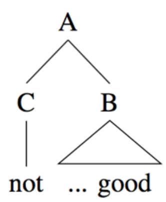

Figure 2. The Subtree with latent labels {A, B, C} is the subject of local feature extraction.

is a plenty of space for feature engineering, i.e. introducing other local or non-local features (e.g. the cube pruning approach provides an efficient procedure for incorporating non-local features [18]).

- The **Word features** extract the co-occurrence of a latent polarity label and the unigrams in the yield of the syntactic parse tree's node. From node A on Figure 2 we extract the following features: A-NOT, A-GOOD.
- The **label features** describe the latent structure as they are the rules in the context-free grammar terminology. From our example we get: A-C-B.
- The **Compositional feature**s are similar to the label features but we exchange one of the daughters' state label with the head of the particular constituent of the syntactic parse tree. This feature template is designed to capture the lexical dependencies of polarity changing words. In the case of current example we get: A-C-GOOD, A-NOT-B. We experimented with two head finding strategies but the difference between taking the right-most word and the semantic head finding rules of Collins [19] was negligible.

IV. Sentence-Level Polarity Classification

The goal of the sentence-level SA is to classify the polarity of the sentiment which a given sentence conveys. In our first batch of experiments, we investigated whether the sentence internal latent structure helps the prediction of the sentence-level polarity.

A. Datasets

For our experiments, we used 3 corpora from the IT products, restaurants and movies domains, all of which were annotated with positive or negative labels on the sentence level. We split our corpora to train and test sets with sizes of 100,000 and 1,000 text examples.

1) IT Products

We downloaded reviews from the Newegg[2] site. Each review on this site must contain the short pro and con summaries of the review in free textual form. We have downloaded the pros and cons of those products which were in the IT category and used them as positive and negative examples, respectively. The downloaded texts were noisy because many of them did not contain the appropriate sentiment (e.g. PRO: I didn't find any.). To overcome this problem, we used only those texts which token length is between 6 and 40 tokens and contains only one sentence.

2) Restaurants

In the case of the restaurant review domain, we applied a similar procedure. We used the dataset provided by Yelp[3], which contains reviews about businesses (we only used those which are related to restaurants). Each review is annotated with stars from 1 to 5 by the reviewer. We selected only the ones which were annotated with 1 or 5 as negative and positive examples, respectively. In order to filter out noise, we applied the same method as before.

3) Movie

For the movie review domain, we downloaded reviews similarly to [3] and [9] from www.rottentomatoes.com. We filtered this dataset as well and used only the reviews with score 1 and 5 as negative and positive examples.

B. Experimental Setup

We predict the whole sentiment tree for the test sentences and we considered the label on the resulting trees' root node as the sentence-level polarity.

For comparison reasons, we ran the RNTN system introduced in [3], which – similarly to our system – yields syntactic trees with a polarity level on each node. The difference of this system and ours is that it was pre-trained[4] on fully annotated trees from the Stanford Sentiment Treebank[5]. The system can predict polarity labels along with their probability values on a five-level scale (very negative, negative, neutral, positive, very positive). Using the probability values, we mapped its prediction to positive or negative labels according to the highest probability value ignoring the neutral label[6].

C. Results on Sentence-Level SA

Our results can be seen in Table I, which contains the accuracy of the systems of each domain. Our baseline system used only unigram features, so it could not exploit

---

2 www.newegg.com

3 www.yelp.com/dataset challenge

4 We re-trained the system on the train part of the Stanford Sentiment Treebank.

5 The Stanford Sentiment Treebank was composed of movie reviews from RottenTomatoes.

6 We experimented with various mapping strategies from 5 polarity levels to 2 levels but the difference between the achieved accuracies were negligible.

the inner structure of the sentences. We used a smaller (10K sentences) and a bigger (100K sentences) training sets along with the same test set for evaluating the unigram baseline and the proposed latent representation-based models.

Two conclusions can be made based on Table I. Firstly, in the case of all the three domains with 100K train the latent system outperformed the baseline. This shows us that by exploiting the latent structure of the sentences, the performance of the SA system could be increased. With the feature templates introduced, our system managed to learn structures and using this it can classify more sentences correctly than the simple bag-of-words models. It also shows that 10K train sentences are not enough to the latent method, it could even achieve worse results than the baseline in the IT product dataset.

On the other hand, it can be seen that the baseline and our system outperformed the reference system in the case of the IT product and restaurant domains but not in the movie domain. The reason why the RNTN system performed well on the movies domain but not on the other two is that it was trained on movie reviews. This confirms the fact that it is important to train an SA model on a domain which is similar to the one on which it will be used. If a fully annotated treebank is available in the given domain, the supervised model is more efficient but competitive results can be achieved with this employing a 10-times bigger training dataset and the proposed latent representation.

TABLE I. ACCURACIES ACHIEVED ON THE THREE DOMAINS. RNTN IS OUR REFERENCE SYSTEM [3], THE BASELINE IS A UNIGRAM MODEL AND LATENT REFERS TO THE PROPOSED SYSTEM.

	IT products	**restaurants**	**movies**
Most frequent class	53.0%	88.0%	50.1%
RNTN	62.1%	79.1%	76.9%
baseline 10k	77.4%	91.8%	67.8%
latent 10k	76.5%	91.9%	67.9%
baseline 100k	82.9%	93.6%	75.7%
latent 100k	83.4%	93.8%	76.6%

V. TARGET-LEVEL POLARITY CLASSIFICATION

In the second batch of experiment, we investigated the utility of the latent sentiment annotation for target-level polarity classification. The task of target-oriented SA is to classify sentiments which refer to a given target. The difficulty of this task is that a sentence can contain multiple targets, e.g. The food was good, but it was too expensive. In this example, a positive sentiment refers to the food quality but a negative one refers to the prices. Using a SA model which is not aware of the targets can easily misclassify the sentiments. We utilized the sentiment trees for target-oriented SA by inducing the sentiment trees then extract features from them for a target-level polarity classifier.

A. Target-Level Dataset

For the evaluation of target-level classifiers we used the dataset provided by the organizers of SemEval-2014 Task 4 – Aspect Based Sentiment Analysis [12], which consists of laptop and restaurant review sentences. For each review, aspects of an entity are annotated, such as the battery life of a laptop or the prices of a restaurant. The aspect mentions are the targets of the sentiment analysis task. For each aspect notation, the polarity level is given depending on the sentiments related to the given aspect in that review. In the database, 4 polarity labels were used which were positive, negative, neutral and conflict (when both positive and negative sentiments were referring to a target). We did not use the conflict class because of the small number of occurrences in the corpus. The resulting database consists of 2,300 laptop and 3,602 restaurant reviews, which will be referred as absa-laptop and absa-restaurant. We only used the train sets of the official datasets and ran 10-fold cross-validation to obtain our results. The reason for this decision is that in our early experiments we noticed that the standard deviation of the accuracy among each fold and the test set is high (2.9% and 2.3% for the laptop and restaurant datasets respectively) thus by cross validating we got much robust results.

B. Exploitation of Latent Sentiment Trees in Target-Level SA

To solve the target-oriented SA problem we used a bag-of-features model with Naïve Bayes classifier from the MALLET toolkit [20] with default parameter values. The features describing a sentence consist of word unigrams along with features derived from the predicted latent sentiment tree. We selected a subtree of the whole latent sentiment tree in order to emphasize the part of the text which is related to the target in question. This subtree is the smallest subtree which 1) contains the target mention and 2) has at least as many leaves as the quarter of the number of words in the sentence.

The exact features used by the classifier are the following:

- word unigrams
- label of the sentiment subtree's root
- the label sequence on the path from the root to the target in the subtree
- the number of each polarity label in the above path respectively
- the collapsed label sequence on the path from the root to the target, more precisely we collapsed the consecutive equal labels, e.g. 0_A_A_C_B_B_B → 0_A+_C_B+
- the same as the last 4 features but by using the entire tree

C. Results on Target-Level SA

The accuracy of the target-oriented system can be seen in Table II for both the absa-laptops and absa-restaurants databases. The baseline for this experiment is a simple bag-of-words model (unigrams without the sentiment tree features). The other rows in the table differ in the model used for predicting the sentiment tree for the sentences.

Similar to the sentence-level task, we used the pre-trained fully supervised RNTN system for comparison reasons. In the case of the last row our models trained on 100,000 sentence-level annotated IT products and restaurants datasets were used for the absa-laptops and absa-restaurants respectively.

TABLE II. ACCURACY SCORES OF THE TARGET-ORIENTED 3-CLASS CLASSIFIER WHOSE FEATURE SET IS ENRICHED BY SENTIMENT-TREE BASED FEATURES. WE CALCULATED THE ACCURACY USING 10-FOLD CROSS VALIDATION ON THE ABSA-LAPTOP AND ABSA-RESTAURANT DATABASES USING THE SENTIMENT TREE BASED FEATURES.

	absa-laptops	**absa-restaurants**
baseline	64.30%	67.42%
baseline + RNTN features	64.81%	66.50%
baseline + latent-tree features	67.47%	69.95%

From the results it can be seen that the performance of the target-oriented system could be considerably improved by using additional features derived from the sentiment tree. The RNTN system was trained on out-domain data, thus it only helped on the laptop dataset but not on the restaurant reviews. Because our model was trained on in-domain data it managed to capture latent semantics of the given domain more accurately and by using the sentiment tree-based features we managed to increase the accuracy on both target-level corpora.

VI. DISCUSSION

We manually and statistically investigated the output of the models used in our experiments in order to reveal the reasons for accuracy differences.

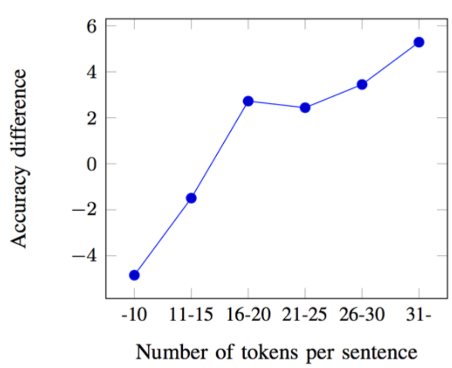

Figure 3. Average accuracy improvements in percentage points of the latent system over the baseline system on the movie test dataset in the function of sentence length.

The reason why our latent model can outperform the supervised RNTN system [3] lies in the domain differences which were used to train the systems. The RNTN system was trained on movie reviews and it performed better on the Movies test corpus but worse on the other two one comparing to our system which was trained on the same domain as the test domain. The domain difference can be captured at the lexical level. For instance, the word cheap has opposite polarity content in the IT and movie domains as it is positive in case we want to buy a device but negative in case of a movie because it implies the poor quality of a film. Similarly, fast and quiet acts the same. There are some strongly IT related terms like WiFi or Gigabit which are positive in this domain but neutral in the movies domain thus the RNTN system interprets it incorrectly on IT reviews. The restaurant domain act similarly, there are domain specific words as well like Mexican which bears a different polarity content in case of cuisines and otherwise. We investigated the differences between the outputs of the baseline unigram classifier and our latent structure-based model. The only considerable explanation we found is that our in-sentence structure-based method could outperform the baseline with a greater advance at longer sentences. Figure 3. depicts the difference between the accuracies achieved by the two system on the Restaurant database in the function of sentence length.

In case of the target-oriented evaluation, the performance increase was achieved by both the full sentiment tree and the selected subtree. In cases when only one target was presented in a sentence the correct label on the root of the sentiment tree helped the classification. Because our latent model can only predict positive or negative on the root (due to the fact that it was trained using binary training data) this could not help in the case of the neutral label. On the other hand, when multiple targets were in a sentence, the label of the selected subtree helped the classification. We sorted the feature of the Naïve Bayes model by the absolute value of learnt feature weights. The top two features from the sentiment subtree-based features were the label C which indicated the neutral class and the number of each polarity labels on the path from the root to the target. On the other hand, the full and the collapsed path-based label sequence features were less effective because their data sparsity. With the additional sentiment tree based features we managed to improve the classification of the positive and negative labels on both absa datasets and in case of the laptop domain we increased the accuracy of the neutral class as well. This latter result is surprising because our latent model was trained only on positive and negative class labels.

VII. CONCLUSIONS

We introduced a sentiment analysis framework which uses a latent state representation on the syntactic structure of the sentence in question. The main contribution of the system is that it uses only sentence-level polarity annotations for training while it is not required to manually handle in-sentence language phenomena (like negation and intensification). The experimental results introduced support the fact that polarity classification is a highly domain-dependent task as the analyzers trained on out-domain sentences failed. They also showed that our sentiment analyzer that has access only to sentence-level polarity annotation for in-domain sentences can outperform state-of-the-art models that were trained on out-domain parse trees with sentiment annotation for each node of the trees. In practice, millions of sentence-level polarity annotations are usually available for a particular domain thus our approach is applicable for training a sentiment analyzer for a new domain and it can exploit the syntactic structure of sentences.

Besides the evaluation of sentence-level polarity classifiers, we utilized the internal structure of the sentiment trees in target-level polarity classification as well. The features extracted from the sentiment trees had a considerable added value for target-level polarity classification and we also show that the latent sentiment trees predicted by models trained in-domain are more useful than the concrete sentiment trees predicted by RNTN which was trained on an out-domain treebank.

ACKNOWLEDGMENT

The research of Richárd Farkas was funded by the János Bolyai Scholarship of the Hungarian Academy of Sciences.

REFERENCES

[1] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," Knowledge-Based Systems, vol. 89, pp. 14–46, 2015.

[2] D. Vilares, M. A. Alonso, and C. Gómez-Rodriguez, " A syntactic approach for opinion mining on Spanish reviews," Natural Language Engineering, vol. 21, no. 01, pp. 139–163, 2013.

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

[4] B. Liu, "Sentiment analysis and opinion mining," Synthesis Lectures on Human Language Technologies, vol. 5, no. 1, pp. 1–167, 2012.

[5] S. Rosenthal, P. Nakov, A. Ritter, and V. Stoyanov, "Semeval-2014 task 9: Sentiment analysis in twitter," in Proc. SemEval, no. SemEval, 2014, pp. 73–80.

[6] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining," in Proc. LREC, 2010.

[7] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity in phrase-level sentiment analysis," in Proc. HLT/EMNLP, 2005, pp. 347–354.

[8] A. Lazaridou, I. Titov, and C. Sporleder, "A Bayesian Model for Joint Unsupervised Induction of Sentiment, Aspect and Discourse Representations." in Proc. ACL, 2013, pp. 1630–1639.

[9] L. Dong, F. Wei, S. Liu, M. Zhou, and K. Xu, "A statistical parsing framework for sentiment classification," Computational Linguistics, vol. 41, no. 2, pp. 265–308, 2015.

[10] R. Feldman, "Techniques and applications for sentiment analysis," Communications of the ACM, vol. 56, no. 4, p. 82, apr 2013.

[11] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter Sentiment Classification," in Proc. ACL, 2011, pp. 151–160.

[12] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "Semeval-2014 task 4: Aspect based sentiment analysis," in Proc. SemEval, ser. SemEval '14, 2014, pp. 27–35.

[13] G. Angeli, C. D. Manning, and D. Jurafsky, "Parsing time: Learning to interpret time expressions," in Proc. NAACL HLT, ser. NAACL HLT '12, Stroudsburg, PA, USA, 2012, pp. 446–455.

[14] A. Björkelund and J. Kuhn, "Learning structured perceptrons for coreference resolution with latent antecedents and non-local features," in Proc. ACL, 2014, pp. 47–57.

[15] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky, "The Stanford CoreNLP natural language processing toolkit," in Proc. ACL, 2014, pp. 55–60.

[16] S. Petrov, L. Barrett, R. Thibaux, and D. Klein, "Learning accurate, compact, and interpretable tree annotation," in Proc. ACL, 2006, pp. 433–440.

[17] M. Collins, "Discriminative training methods for hidden Markov models," in Proc. EMNLP, vol. 10, 2002, pp. 1–8.

[18] L. Huang, "Forest reranking: Discriminative parsing with non-local features," in Proc. ACL HLT, 2008, pp. 586–594.

[19] M. Collins, "Head-driven statistical models for natural language parsing," Computational linguistics, vol. 29, no. 4, pp. 589–637, 2003.

[20] A. K. McCallum, "MALLET: A Machine Learning for Language Toolkit," 2002.

AUTHORS' BACKGROUND

Your Name	Title*	Research Field	Personal website
Viktor Hangya	Phd candidate	Natural Language Processing	http://www.inf.u-szeged.hu/~hangyav/
Zsolt Szántó	Phd candidate	Natural Language Processing	http://www.inf.u-szeged.hu/~szantozs/
Richárd Farkas	assistant professor	Natural Language Processing	http://www.inf.u-szeged.hu/~rfarkas/

*This form helps us to understand your paper better, the form itself will not be published.

*Title can be chosen from: master student, Phd candidate, assistant professor, lecturer, senior lecture, associate professor, full professor