ECNU at SemEval-2016 Task 5: Extracting Effective Features from Relevant Fragments in Sentence for Aspect-Based Sentiment Analysis in Reviews

Mengxiao Jiang[1], **Zhihua Zhang**[1], **Man Lan**[1,2*]
[1]Department of Computer Science and Technology,
East China Normal University, Shanghai, P.R.China
[2]Shanghai Key Laboratory of Multidimensional Information Processing
{51151201080, 51131201039}@ecnu.cn, mlan@cs.ecnu.edu.cn*

Abstract

This paper describes our systems submitted to the Sentence-level and Text-level Aspect-Based Sentiment Analysis (ABSA) task (i.e., Task 5) in SemEval-2016. The task involves two phases, namely, Aspect Detection phase and Sentiment Polarity Classification phase. We participated in the second phase of both subtasks in laptop and restaurant domains, which focuses on the sentiment analysis based on the given aspect. In this task, we extracted four types of features (i.e., Sentiment Lexicon Features, Linguistic Features, Topic Model Features and Word2vec Feature) from certain fragments related to aspect rather than the whole sentence. Then the proposed features are fed into supervised classifiers for sentiment analysis. Our submissions rank above average.

1 Introduction

Aspect-Based Sentiment Analysis task (ABSA), i.e., task 5 in SemEval-2016, is an interesting task, which focuses on the sentiment analysis based on the target and certain categories. The organizers set up three subtasks, i.e., Sentence-level ABSA (i.e, Subtask 1), Text-level ABSA (i.e., Subtask 2), and Out-of domain ABSA (i.e., Subtask 3). For subtask 1 and 2, the training data in each domain is provided, while no labeled data is provided for subtask 3. Given an opinionated document in a domain, both subtask 1 and subtask 2 are grouped into two phases, i.e., Aspect Detection phase and Sentiment Polarity Classification phase. We participated in the second phase of these two subtasks, aiming to identify the sentiment polarity for each given aspect which is made up of *<E#A, OTE>* in two domains.

Specifically, for Sentence-level ABSA, focusing on identifying all the opinion tuples (i.e., *<E#A, OTE, polarity>*) in each sentence of the reviews, the Aspect Detection phase contains two slots. The Slot1 is to identify every entity (i.e., E) and attribute (i.e., A) pair (also named as category, e.g., *RESTAURANT-PRICES*) according to given sentence, and the Slot2 focuses on detecting Opinion Target Expression (i.e., *OTE* or target in short). For example, in *"Pizza here is consistently good"*, the participants are required to recognize *Pizza* as *OTE* and *FOOD#QUALITY* as *E#A*. The second phase, i.e., Sentiment Polarity Classification (Slot3), is to determine the sentiment polarity (i.e., positive, negative, or neutral) for each aspect (i.e., *<E#A, OTE>*). As for Text-level ABSA, aiming at identifying the opinion tuples (i.e., *<E#A, polarity>*) expressed in each review, the Aspect Detection phase is to identify the *E#A* pairs and the second phase is to assign the sentiment label (positive, negative, neutral or conflict) for each detected *E#A*. The conflict label is assigned when the dominant sentiment of the opinion is not clear.

In previous work, (Kim et al., 2013) presented a hierarchical aspect sentiment model to classify the polarity of aspect terms from unlabeled online reviews. (Jiménez-Zafra et al., 2015) proposed a syntactic approach for identifying the words that modify each aspect. (Branavan et al., 2009; He et al., 2012; Mei et al., 2007) used topic or category information. (Saias, 2015) used a 3-class classifier and

Proceedings of SemEval-2016, pages 373–378,
San Diego, California, June 16-17, 2016.

some handcrafted features to perform ABSA. (Lin and He, 2009; Jo and Oh, 2011) presented LDA-based models, which incorporated aspect and sentiment analysis together to model sentiments towards different aspects. Unlike these work, which try to extract features from the whole sentence, we propose a method which just takes certain fragments related to the given aspect from the sentence into consideration to perform feature engineering for the ABSA task.

The rest of this paper is structured as follows. In Section 2, we describe our system in details, including motivation, preprocessing, feature engineering, evaluation metric and algorithm, etc. Section 3 reports data sets, experiments and result discussion. Finally, Section 4 concludes our work.

2 System Description

2.1 Motivation

At sentence-level ABSA, generally, a review consists of several sentences and one single sentence may contain mixed opinion tuples (i.e., *<OTE, E#A, polarity >*) towards different *OTE* or *E#A*. The goal of our system is to identify the polarity for each opinion tuple. We found that the given aspect is just related to certain fragments in corresponding sentence. Therefore, in order to extract features from the relevant fragments, we proposed a two-step method to acquire potential words related to given aspect as pending words for future feature extraction. This approach consists of two steps, i.e., *Segmentation step*, which is to split each sentence into several fragments, and *Selection Step*, which selects out one or more fragments from sentence for each aspect.

Specifically, in the *Segmentation Step*, we used punctuation marks (i.e., {*,.?!*}) and conjunctions (i.e., {*and, but*}) to split the sentence into several *candidate fragments*. It is worth noting that the *OTE* is the entity or attribute words in reviews the users explicitly indicated. When there is no explicit mention of the entity, the *OTE* takes the value *NULL*. In restaurant domain, both *E#A* and *OTE* are provided in reviews, whereas only *E#A* are annotated and provided in the laptop domain, we assumed its *OTE* is *NULL*. Therefore, we adopted two strategies for *Selection Step*. In the case that the targets are provided, we located the fragment which contains the target as *target fragment* and selected the words ranging from the prior *target fragment* (not include) to the current *target fragment* (included) as pending words. In another case that the targets are *NULL*, we automatically assigned a *target fragment* for it as follows. We firstly divided all sentences in training data into several subsets according to their attributes (i.e., A in E#A). If multiple attributes exist in the same sentence then the sentence is shared in corresponding subsets. Then we calculated the *tfidf* score for each word in each subset. Finally, we summed up the *tfidf* scores of all words in each fragment according to the attribute in given opinion. The fragment with top score is set as *target fragment*. After locating the *target fragment*, the approach to select the pending words is the same as the case that the targets are provided.

As for text-level ABSA, the opinion tuples (i.e., *<E#A, polarity>*) are endowed with each review rather than the sentence. Based on the statistic of the training data, we found that the labels of *E#A* in text-level are consistent with the most frequent polarity of the corresponding *E#A* in one review at sentence-level. Thus, for subtask 2 (i.e., text-level), we counted the number of positive, negative and neutral labels for each *E#A* in each review from the results of subtask 1. Then the most frequent polarity of each *E#A* is set as the label for corresponding *E#A* in each review in subtask 2.

For each domain, the participants are required to submit two runs, (1) *constrained:* only the provided data can be used; (2) *unconstrained:* any additional resources can be used. In this task, we adopted external resources, i.e., 8 sentiment lexicons and 100 billion words from Google News, to train the *Sentiment Lexicon* features and the *Word2vec* (Mikolov et al., 2013) feature. Thus, the difference between our two systems lies in these two features. For both systems, we also extracted many traditional types of features to build classifiers for classification.

2.2 Data Preprocessing

The original data is provided in XML format. So we first removed the XML tags from data and then transformed the abbreviations to their normal for-

mat. We used *Stanford Parser tools*[1] for tokenization, POS tagging and parsing. Then, the WordNet-based Lemmatizer implemented in NLTK[2] was adopted to lemmatize words to their base forms with the aid of their POS tags.

2.3 Feature Engineering

Four types of features extracted from the pending words are adopted to build the classifiers, i.e., Linguistic features, Sentiment Lexicon features, Topic Model features and Word2vec feature.

2.3.1 Linguistic Features

Word *N*-grams: For all pending words, after transforming them into lowercase, we extracted the unigram, bigram, trigram and 4-gram as word *N*-grams features.

Lemmatized Word *N*-grams:

Pending words were lemmatized by NLTK firstly, then we extracted four types of *N*-grams from the lemmatized form as Lemmatized Word *N*-grams, i.e., unigram_L, bigram_L, trigram_L and 4-gram_L.

Word Nchars: We recorded presence or absence of contiguous sequences of 3, 4, and 5 characters from pending words as *N-chars* features.

POS: We counted the number of nouns (the corresponding POS tags were *NN, NNP, NNS and NNPS*), verbs (*VB, VBD, VBG, VBN, VBP and VBZ*), adjectives (*JJ, JJR and JJS*) and adverbs (*RB, RBR and RBS*) in pending words as the pos feature.

Allcaps: It indicated the number of uppercase words in pending words.

Elongated: We recorded the number of the words contained the repeating characters (e.g., *slowwwwww*) as the elongated feature.

Punctuation: Customers often use exclamation mark (!) and question mark (?) to express surprise or emphasis, so we recorded the number of exclamation and question marks in pending words as the punctuation features.

Negation: Negation comprised various kinds of devices to reverse the truth value of a proposition, thus the identification of negations is very essential. In our work, we collected 29 negations from Internet and designed this binary feature to indicate whether there is negation in pending words.

[1]http://nlp.stanford.edu/software/lex-parser.shtml

[2]http://nltk.org

2.3.2 Sentiment Lexicon Features

Giving the pending words, we first converted them into lowercase and then calculated five sentiment scores for each sentiment lexicon to construct *Sentiment Lexicon Features* (*SentiLexi*) (1) the ratio of positive words to pending words, (2) the ratio of negative words to pending words, (3) the maximum sentiment score, (4) the minimum sentiment score, (5) the sum of sentiment scores. We transformed the sentiment scores in all sentiment lexicons to the range of $[-1, 1]$, where "$-$" denotes negative sentiment. If the pending word does not exist in one sentiment lexicon, its corresponding score is set to zero. The following 8 sentiment lexicons are adopted in our systems: *Bing Liu opinion lexicon*[3], *General Inquirer lexicon*[4], *IMDB*[5], *MPQA*[6], *AFINN*[7], *SentiWordNet*[8], *NRC Hashtag Sentiment Lexicon*[9], *NRC Sentiment140 Lexicon*[10].

2.3.3 Topic Model Features

With the aid of *LDA-C tool*[11] with default parameter setting, we generate topic-related features from all training data as follows.

Sent2Topic: The *LDA* could generate the document distribution among predefined topics. We extracted this distribution as *Sent2Topic* feature.

Top Topic word (TopTopic): Since the topic probability of each word indicates its significance in corresponding topic, we set 20 topics and collect the top 25 words in each topic to build *TopTopic* feature.

2.3.4 Word2vec Feature

Google Word2vec (GoogleW2V): We used the publicly available *word2vec* tool[12] to get word vectors with dimensionality of 300, which is trained on 100 billion words from Google News as *GoogleW2V*.

[3]http://www.cs.uic.edu/liub/FBS/sentiment-analysis.html#lexicon

[4]http://www.wjh.harvard.edu/inquirer/homecat.htm

[5]http://anthology.aclweb.org//S/S13/S13-2.pdf#page=444

[6]http://mpqa.cs.pitt.edu/

[7]http://www2.imm.dtu.dk/pubdb/views/publication_details.php?id=6010

[8]http://sentiwordnet.isti.cnr.it/

[9]http://www.umiacs.umd.edu/saif/WebDocs/NRC-Hashtag-Sentiment-Lexicon-v0.1.zip

[10]http://help.sentiment140.com/for-students/

[11]http://www.cs.princeton.edu/ blei/lda-c/

[12]https://code.google.com/archive/p/word2vec

2.4 Evaluation Measure and Algorithm

To evaluate the performance of different systems, the official evaluation measure *accuracy* is adopted. We employ the *Logistic Regression* algorithm with the default parameter implemented in *liblinear* tools[13] to build the classifiers for its good performance in our preliminary experiments. The 5-fold cross validation is adopted for system development.

3 Experiment

3.1 Datasets

In restaurant domain, the opinion tuple is composed of target, category and polarity (i.e., <*OTE, E#A, Polarity*>). And in laptop domain, the *OTE* is not taken into account in opinion tuple (i.e., <*E#A, Polarity*>). The restaurant domain contains 6 entities (e.g., *AMBIENCE*, *DRINKS*, *FOOD*, *RESTAURANT*, etc) and 5 attributes (i.e., *GENERAL*, *PRICES*, *STYLE_OPTIONS*, *QUALITY*, *PRICES*, etc). While in laptop, 22 entities (e.g., *BATTERY*, *SUPPORT*, *CPU*, *COMPANY*, etc.) and 9 attributes (e.g., *USABILITY*, *GENERAL*, *QUALITY*, etc) are tagged. Table 1 shows the statistics of the data sets used in our experiments.

Data	Reviews	Sentences	Opinions	Positive	Negative	Neutral	Conflict
Restaurant(SB1):							
train	350	2,000	2,506	1,657	751	98	0
test	90	676	859	611	204	44	0
Laptop(SB1):							
train	450	2,500	2,908	1,634	1,086	188	0
test	80	808	801	481	274	46	0
Restaurant(SB2):							
train	335	1,950	1,435	1012	327	55	41
test	90	676	404	286	84	23	11
Laptop(SB2):							
train	395	2,373	2,082	1,210	708	123	41
test	80	808	545	338	162	31	14

Table 1: Statistics of training and test dataset of two subtasks in laptop and restaurant domains. *Positive, Negative, Neural, Conflict* stand for the number of corresponding labels.

3.2 Experiments on Training data

For both laptop and restaurant domains, we adopted similar methods, i.e, employing rich features to build classifiers, and performed constrained systems and unconstrained systems respectively. Since *Sentiment lexicon* feature and *GoogleW2V* feature utilized the external data, we did not use these two types of features in the constrained system. As for unconstrained systems, all features were employed. As for feature selection, a *hill climbing* algorithm is adopted to find out the contributions of different features, which is described as: keeping adding one type of feature at a time until no further improvement can be achieved. Table 2 shows the results of feature selection experiments for unconstrained and constrained systems in restaurant and laptop domains.

According to Table 2, it is interesting to find that (1) *3-char*, *4-char* and *negation* are beneficial to this task. The possible reason may be that there exists a lot of derivations in training data, e.g., *relax* and *relaxing*. Besides, the negator always reverses the sentiment polarity of corresponding review, which results in the good contribution of *negation* feature. (2) *SentiLexi* features are effective in two domains. In our preliminary experiments, we found that the *SentiLexi* features made great contribution to sentiment analysis task, which indicates that this type of features are indeed significant. (3) *POS* features are not quite effective in all systems. The possible reason may be that *POS* aims at identifying the subjective instances from objective ones, but the objective records just occupy a small proportion. (4) The majority of features are more valid in unconstrained system than that in constrained system. The possible reason may be that there are certain overlapped information between the *SentiLexi* features, the Word2vec feature and the other features.

In our preliminary experiments, we conducted the baseline system where features are extracted from the whole sentence without the consideration of *OTE* and *E#A*. The result showed that the method described in section 2.1 outperformed the baseline. Thus, we used the strategy that extracting features from certain relevant fragments rather than the whole sentence for this task.

3.3 Results and Discussion on test data

Using the optimum feature set shown in Table 2 and the algorithm described in section 2.4, we trained separate models for each domain and evaluated them against the test set in SemEval-2016 Task 5. For both subtask 1 and 2, we constructed 4 systems for unconstrained and constrained systems in restaurant and laptop domains respectively.

From the Table 3, we find that: (1) The uncon-

[13]https://www.csie.ntu.edu.tw/ cjlin/liblinear/

Features		Laptop Domain		Restaurant Domain	
		constrain	unconstrain	constrain	unconstrain
Linguistic	unigram		√	√	√
	bigram		√		
	trigram		√		√
	forgram		√		√
	unigram_L		√		√
	bigram_L		√		√
	trigram_L		√		
	forgram_L		√		
	trichar	√	√	√	√
	forchar	√	√	√	
	fifchar		√		
	POS				√
	AllCaps		√	√	
	Elongated	√	√		√
	Punctuation		√		√
	Negation	√	√	√	√
Sentiment Lexicon	SentiLexi	-	√	-	√
Topic Model	Sent2Topic			√	
	TopTopic		√		√
GoogleW2V	GoogleW2V	-	√	-	√
Accuracy (%)		76.81	81.09	77.81	83.36

Table 2: Results of feature selection experiments for restaurant and laptop domains on training datasets.

strained system performed better than constrained system in both laptop and restaurant domains. This implicates that the *SentiLexi* feature and the *GoogleW2V* feature are effective for performance improvement in sentiment classification. (2) The accuracy in restaurant domain is higher than that in laptop. One reason may be that in laptop domain, the *OTE* are not provided.

Subtask	TeamID	Restaurant		Laptop	
		Con	Uncon	Con	Uncon
SB1	ECNU	80.559(5)	83.586(4)	70.037(6)	78.152(3)
	XRCE	88.126(1)	-	-	-
	LeeHu	-	-	75.905(1)	-
	IIT-T	-	86.729(1)	-	82.772(1)
SB2	ECNU	78.713(2)	81.436(2)	67.523(3)	75.046(1)
	UWB	80.941(1)	81.931(1)	74.495(1)	-

Table 3: Performance of our systems and the top-ranked systems for laptop and restaurant domains in terms of *Accuracy(%)* on test datasets. *Con* stands for *constrained* and *Uncon* represents *unconstrained*. The numbers in the brackets are the rankings on corresponding submissions.

4 Conclusion

In this paper, we extracted several types of features, i.e., *Linguistic* features, *SentiLexi* features, *Topic Model* features and *Word2vec* feature, and employed the Logistic Regression classifier to detect the sentiment polarity in given aspect for reviews. Moreover, we have demonstrated a two-step approach to acquire the pending words from the relevant fragments instead of the whole sentences for feature extraction. This enables the system to capture the relationship between the sentiment of the sentence and its opinion adherent. The results on test and training data showed the effectiveness of our method for the ABSA task. For the future work, it would be interesting to explore domain-specific sentiment lexicons to improve the performance and examine more advanced ways of using sentiment lexicons and word embedding features.

Acknowledgments

This research is supported by grants from Science and Technology Commission of Shanghai Municipality (14DZ2260800 and 15ZR1410700), Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of Things (ZF1213).

References

SRK Branavan, Harr Chen, Jacob Eisenstein, and Regina Barzilay. 2009. Learning document-level semantic

properties from free-text annotations. *Journal of Artificial Intelligence Research*, 34(2):569.

Yulan He, Chenghua Lin, Wei Gao, and Kam-Fai Wong. 2012. Tracking sentiment and topic dynamics from social media. In *Sixth International AAAI Conference on Weblogs and Social Media*.

Salud M. Jiménez-Zafra, Eugenio Martínez-Cámara, M. Teresa Martín-Valdivia, and L. Alfonso Ureña López. 2015. Sinai: Syntactic approach for aspect-based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 730–735, Denver, Colorado, June. Association for Computational Linguistics.

Yohan Jo and Alice H Oh. 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the 4th ACM international conference on Web search and data mining*, pages 815–824.

Suin Kim, Jianwen Zhang, Zheng Chen, Alice Oh, and Shixia Liu. 2013. A hierarchical aspect-sentiment model for online reviews. In *Proceedings of The Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-13)*. AAAI, July.

Chenghua Lin and Yulan He. 2009. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on WWW*, pages 171–180.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

José Saias. 2015. Sentiue: Target and aspect based sentiment analysis in semeval-2015 task 12. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 767–771, Denver, Colorado, June. Association for Computational Linguistics.