ECNU: A Combination Method and Multiple Features for Aspect Extraction and Sentiment Polarity Classification

Fangxi Zhang, Zhihua Zhang, Man Lan*
Department of Computer Science and Technology
East China Normal University
51111201041,51131201039@ecnu.cn; mlan@cs.ecnu.edu.cn*

Abstract

This paper reports our submissions to the four subtasks of Aspect Based Sentiment Analysis (ABSA) task (i.e., task 4) in SemEval 2014 including aspect term extraction and aspect sentiment polarity classification (Aspect-level tasks), aspect category detection and aspect category sentiment polarity classification (Category-level tasks). For aspect term extraction, we present three methods, i.e., noun phrase (NP) extraction, Named Entity Recognition (NER) and a combination of NP and NER method. For aspect sentiment classification, we extracted several features, i.e., topic features, sentiment lexicon features, and adopted a Maximum Entropy classifier. Our submissions rank above average.

1 Introduction

Recently, sentiment analysis has attracted a lot of attention from researchers. Most previous work attempted to detect overall sentiment polarity on a text span, such as document, paragraph and sentence. Since sentiments expressed in text always adhere to objects, it is much meaningful to identify the sentiment target and its orientation, which helps user gain precise sentiment insights on specific sentiment target.

The aspect based sentiment analysis (ABSA) task (Task 4) (Pontiki et al., 2014) in SemEval 2014 is to extract aspect terms, determine its semantic category, and then to detect the sentiment orientation of the extracted aspect terms and its category. Specifically, it consists of 4 subtasks. The aspect term extraction (ATE) aims to extract the aspect terms from the sentences in two given domains (laptop and restaurant). The aspect category detection (ACD) is to identify the semantic category of aspects in a predefined set of aspect categories (e.g., food, price). The aspect term polarity (ATP) classification is to determine whether the sentiment polarity of each aspect is positive, negative, neutral or conflict (i.e., both positive and negative). The aspect category polarity (ACP) classification is to determine the sentiment polarity of each aspect category. We participated in these four subtasks.

Generally, there are three methods to extract aspect terms: unsupervised learning method based on word frequency ((Ku et al., 2006), (Long et al., 2010)), supervised machine learning method (Kovelamudi et al., 2011) and semi-supervised learning method (Mukherjee and Liu, 2012) where only several user interested category seeds are given and used to extract more categorize aspect terms. Since sentiments always adhere to entities, several researchers worked on polarity classification of entity. For example, (Godbole et al., 2007) proposed a system that assigned scores representing positive or negative opinion to each distinct entity in the corpus. (Kim et al., 2013) presented a hierarchical aspect sentiment model to classify the polarity of aspect terms from unlabeled online reviews. Moreover, some sentiment lexicons, such as SentiWordNet (Baccianella et al., 2010) and MPQA Subjectivity Lexicon (Wilson et al., 2009), have been used to generate sentiment score features (Zhu et al., 2013).

The rest of this paper is organized as follows. From Section 2 to Section 5, we describe our approaches to the Aspect Term Extraction task, the Aspect Category detection task, the Aspect Term Polarity task and the Aspect Category Polarity task respectively. Section 6 provides the conclusion.

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 252–258,
Dublin, Ireland, August 23-24, 2014.

2 Aspect Term Extraction System

For aspect terms extraction task, we first adopted two methods: a noun phrase (NP) based method and a Named Entity Recognition (NER) based method. In our preliminary experiments, we found that the NP-based method generates many noisy terms resulting in high recall and low precision, and the NER-based method performs inverse results. In order to overcome their drawbacks and make use of their advantages, we proposed a third method which combines the two methods by using the results of NP-based method as an additional name list feature to the NER system.

2.1 Preprocessing

We used Stanford Parser Tools[1] for POS tagging and for parsing while the Natural Language Toolkit[2] was used for removing stop words and lemmatization.

2.2 NP-based Method

(Liu, 2012) showed that the majority of aspect terms are noun phrases. Moreover, through the observation of the training set, we found that more than half of the aspects are pure noun phrases or nested noun phrases. So we considered these two types of noun phrases as aspect terms and adopted a rule-based noun phrases extraction system to perform aspect term extraction. This extraction is performed on parsed sentences. For example, from parsed sentence:

"*(CC but)*
(S
(NP (NN iwork))
(VP (VBZ is)
(ADJP (JJ cheap))
(PP (VBN compared)
(PP (TO to)
(NP (NN office))))))"

iwork and *office* with NN tag are extracted as aspect terms. However, to make a more precise extraction, we first removed white lines from parsed sentences. Then we performed extraction only using three continuous lines. Since the NPs we extracted contain much noise which only appear in NPs rather than in gold aspect terms list, we built a *stopwords* list containing these noisy terms especially the numeric expressions. Table 1 shows the set of manually built rules used for NP extraction.

[1]http://nlp.stanford.edu/software/lex-parser.shtml
[2]http://www.nltk.org/

Based on the experimental results on training data, we found the NP-based method achieves high recall and low precision as shown in Table 2. This indicates that we extracted plenty of NPs which consist of a large proportion of aspect terms and much noise such as irrelevant NPs and overlapping phrases. Thus the NP-based method alone has not produced good results.

2.3 NER-based Method

We also cast aspect term extraction task as a traditional NER task (Liu, 2012). We adopted the commonly used BIO tag format to represent the aspect terms in the given annotated training data (Toh et al., 2012), where *B* indicates the beginning of an aspect term, *I* indicates the inside of an aspect term and *O* indicates the outside of an aspect term. For example, given *"the battery life is excellent"*, where *battery life* is annotated as aspect term, we tagged the three words *the*, *is* and *excellent* as *O*, *battery* as *B* and *life* as *I*.

We adopted several widely used features for the NER-based aspect term extraction system.

Word features: current word (*word_0*), previous word (*word_-1*) and next word (*word_1*) are used as word features.

POS feature: the POS tag of current word (*POS_0*), the POS tags of two words around current word (*POS_-2, POS_-1, POS_1, POS_2*), and the combinations of contextual POS tags (*POS_-1/POS_0, POS_0/POS_1, POS_-1/POS_0/POS_1*) are included as POS features.

Word_shape: a tag sequence of characters in current word is recorded, i.e., the lowercase letter tagged as *a*, and the uppercase letter tagged as *A*.

Chunk: We extracted this feature from the POS tag sequence, which is defined as follows: the shortest phrase based on POS taggers, i.e., *"(VP (VBD took) (NP (NN way)) (ADVP (RB too) (RB long))"*, *took* labeled as *O*, *way* labeled as *B-NP*, *too* labeled as *B-ADVP*, *long* labeled as *I-ADVP*.

We implemented a CRF++ [3] based NER system with the above feature types.

2.4 Combination of NP and NER Method

Based on our preliminary experiments, we considered to combine the above two methods. To do so, we adopted the results of the NP system as additional name lists feature for the NER system. Through the observation on the results of the

[3]http://crfpp.googlecode.com/svn/trunk/doc/index.html

if (NP in line_1)	then select line_1 as candidate
if (NP in line_1 and PP in line_2 and NP in line_3)	then select line_1 + line_2 + line_3 as candidate
else if (VB in line_1 and NN in line_2)	then select line_1 + line_2 as candidate
else if (NP in line_1 and NP in line_2)	then select line_1 + line_2 as candidate
else if (NP in line_1 and CC in line_2 and NN in line_3)	then select line_3 as candidate
else if (JJ in line_1 and NN in line_2)	then select line_2 as candidate
if (current term in candidate existing in stopwords)	then remove current term
if (CD start candidate)	then remove CD
if (DT or PRP start candidate)	then remove DT or PRP
if (JJR in candidate)	then remove JJR
if (Punctuation in candidate)	then remove Punctuation

Table 1: The rules in NP-based method.

method	Laptop			Restaurant		
	Precision(%)	Recall(%)	F-score(%)	Precision(%)	Recall(%)	F-score(%)
NP-based	44.35	**74.43**	55.59	45.99	70.50	56.17
NER-based	70.46	48.27	57.29	80.87	68.24	74.02
Combination	**72.79**	55.11	**62.73**	**82.31**	**70.62**	**76.02**

Table 2: The F-scores of three methods on training data.

NP-based method and the NER-based method, we built two types of name lists for our combination method as follows:

Gold Namelist: containing the gold aspect terms and the intersection between the results of the NP-based method and the NER-based method.

Stop Namelist: the words in original sentences but not in gold aspect terms set or not in NPs set we extracted before.

Table 3 shows the results of feature selection for the combination method on training data. The best performance was obtained by using all features. Thus, our final submission system adopted the combination method with all features.

Feature	Dataset	
	Laptop	Restaurant
word:		
+word_0	40.35	58.58
+word_1	54.78	72.23
POS:		
+POS_0	55.81	71.11
+POS_1	57.07	74.02
+POS_2	57.18	73.24
+POS_0/POS_1	51.85	70.58
chunk:		
+chunk_0	56.74	73.45
word_shape:		
+word_shape_0	57.29	74.02
name_list:		
+Gold Namelist	62.66	75.39
+Stop Namelist	**62.73**	**76.02**

Table 3: The F-scores of combination method of subtask 1 on training data based on 2 cross-validation

Table 2 shows the results of the above three systems on training data. Comparing with other two methods, we easily find that the combination method outperforms the other two systems in terms of precision, recall and F values on both domains.

2.5 Result and Discussion

In constrained run, we submitted the results using the method in combination of NP and NER. Specifically, we adopted all features and the name lists listed in Table 3 and the CRF++ tool for the NER system. Table 4 lists the results of our final system and the top two systems officially released by organizers. On both domains, our system ranks above the average under constrained model, which proves the effectiveness of the combination method by using NP extraction and NER.

From Table 2 and Table 4 we find that the results on restaurant data are much better than those on laptop data. Based on our further observation on training data, the possible reason is that the number of numeric descriptions in laptop dataset is much larger than those in restaurant dataset and the aspect terms containing numeric description are quite difficult to be extracted.

Dataset	DLIREC	NRC-Canada	Our result
laptop	70.41	68.57	65.88
restaurant	78.34	80.19	78.24

Table 4: The F-scores (%) of our system and the top two systems of subtask 1 on test dataset.

3 Aspect Category Classification System

Aspect category classification task tries to assign each aspect one or more semantic category labels. Thus, we regarded this task as a multi-class classification problem. Following (Rennie, 2001), we built a binary model for each category, where bag-of-words is used as features.

3.1 Features

We adopted the bag-of-words schema to represent features as follows. Since not all training instances have annotated aspect terms, we extracted only annotated aspect terms from sentence if the sentence contains annotated aspect terms, or extracted all words from sentence which does not contain any annotated aspect terms as features, which results in 5200 word features in total.

3.2 Classification Algorithm

We adopted the maximum entropy algorithm implemented in Mallet toolkit (McCallum, 2002) to build a binary classifier for each category. All parameters are set as defaults. This subtask only provides restaurant data and there are five predefined categories (i.e., food, price, service, ambience and anecdotes/miscellaneous), thus we build five binary classifiers in total.

3.3 Results and Discussions

Table 5 lists the precision, recall and F-score of our final system along with the top two systems released by the organizers.

	Precision(%)	Recall(%)	F-score(%)
our system	65.26	69.46	67.30
rank 1 system	91.04	86.24	88.58
rank 2 system	83.23	81.37	82.29

Table 5: The results of our system and the top two systems of subtask 3 on the test data.

From Table 5, we find that there are quite a large room to improve our system. One main reason is that our system only uses simple features (i.e., bag-of-words) and these simple features may have poor discriminating power. Another possible reason may be that in training data there are at least half sentences without annotated aspect terms. In this case, when we used all words in the sentences as features, it may bring much noise. In future work, we consider to generate more effective features from external resources to indicate the relationships between aspects and categories to improve our system.

4 Aspect Term Sentiment Polarity Classification System

Once we extract aspect terms, this task aims at classifying the sentiment orientation of the annotated aspect terms. To address this task, we firstly extracted three types of features: sentiment lexicon based features, topic model based features and other features. Then two machine learning algorithms, i.e., SVM and MaxEnt, were used to conduct classification models.

4.1 Features

4.1.1 Sentiment Lexicon (SL) Features

We observed that the sentiment orientation of an aspect term is usually revealed by the surrounding terms. So in this feature we took four words before and four words after the current aspect term and then calculated their respective positive,negative and neutral scores. During the calculation we reversed the sentiment orientation of the term if a negation occurs before it. We manually built a negative list: {*no*, *nor*, *not*, *neither*, *none*, *nobody*, *nothing*, *hardly*, *seldom*}. Eight sentiment lexicons are used: Bing Liu opinion lexicon[4], General Inquirer lexicon[5], IMDB[6], MPQA[7], SentiWordNet[8], NRC emotion lexicon[9], NRC Hashtag Sentiment Lexicon[10] and NRC Sentiment140 Lexicon[11]. With regard to the synonym selection of SentiWordNet, we selected the first term in the synset as our lexicon. If the eight words surrounding the aspect term do not exist in the eight corresponding sentiment lexicons, we set their three sentiment scores as 0. Then we got 24 sentiment values for each word (3 polarities * 8 lexicons) and summed up the values of eight words for each sentiment polarity (i.e., positive, negative and neural). Finally we got 24 sentiment lexicon features for each aspect.

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

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

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

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

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

[9] http://mailman.uib.no/public/corpora/2012-June/015643.html

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

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

feature	F-pos(%)		F-neg(%)		F-neu(%)		Acc(%)	
	MaxEnt	SVM	MaxEnt	SVM	MaxEnt	SVM	MaxEnt	SVM
SL	72.50± 1.91	70.99± 5.91	65.10± 1.99	65.66± 3.48	25.54± 5.68	24.02± 9.28	62.28± 2.59	61.61± 4.68
+Other	72.92± 2.12	72.70± 1.44	65.93± 3.89	65.09± 3.67	31.14± 5.77	34.00± 7.31	62.88± 3.22	62.54± 3.17
+Topic	**73.14± 1.02**	72.21± 1.44	**65.55± 5.43**	65.58± 3.45	**34.34± 10.55**	12.16± 4.96	**63.00± 4.34**	61.74± 3.10

Table 6: The results of our system in subtask 2 on laptop training data based on 5-fold cross validation.

features	F-pos(%)		F-neg(%)		F-neu(%)		Acc(%)	
	MaxEnt	SVM	MaxEnt	SVM	MaxEnt	SVM	MaxEnt	SVM
SL	79.78± 1.37	79.85± 1.35	49.37± 3.54	47.96± 4.52	26.02± 3.62	31.67± 2.84	65.61± 2.59	65.45± 1.98
+Other	80.48± 2.18	79.09± 1.42	53.17± 2.70	50.51± 3.34	29.25± 3.60	33.13± 6.89	66.80± 2.33	65.21± 2.35
+Topic	**80.71± 1.71**	77.94± 1.34	**52.61± 2.52**	46.65± 3.17	**34.51± 3.35**	3.40± 2.79	**67.18± 2.52**	64.72± 1.48

Table 7: The results of our system in subtask 2 on restaurant training data based on 5-fold cross validation.

4.1.2 Topic Features

In this section we considered to use the bag-of-topics feature to replace the traditional bag-of-words feature since the bag-of-words feature are very sparse in the data set. To construct the clusters of topics, we used the LDA[12] based topic model to estimate the K topics (in our experiment, we set K to 50) from training data. Then we inferred the topic distribution from training and test data respectively as topic features.

4.1.3 Other Features

Besides, we also proposed the following other features in order to capture more useful information from the short texts.

Aspect distance This feature records the number of words from the current aspect to the next aspect in the same sentence. If the current aspect term is the last term in the sentence, this value is calculated as the negative number of words from the current aspect to the former aspect. If only one aspect term exists in a sentence, then the value is set to zero.

Number of aspects This feature describes the number of aspect terms in the current sentence.

Negation flag feature We set this feature as 1 if a negation word occurs in the current sentence, otherwise -1.

Number of negations This feature is the number of negation words in the current sentence.

4.2 Classification Algorithms

The maximum entropy and SVM which are implemented in Mallet toolkit (McCallum, 2002) and LibSVM (Chang and Lin, 2011) respectively are used to construct the classification model from training data. Due to the limit of time, all parameters are set as defaults.

4.3 Results and Discussions

4.3.1 Results on Training Data

To compare the performance of different features and different algorithms, we performed a 5-fold cross validation on training data of two domains. Table 6 and Table 7 show the results of two domains in terms of F-scores and accuracy with mean and standard deviation. The best results are shown in bold.

From above two tables, we found that (1) MaxEnt performed better than SVM on both datasets and all feature types, and (2) using all features achieved the best results. Moreover, the F-pos result was the highest in both datasets and the possible reason is that the majority of training instances are positive sentiment. We also found that in restaurant dataset, F-neg (52.61%) was much smaller than F-pos (80.17%). However, in laptop dataset, they performed comparable results. The possible reason is that the number of negative instances (805) is much smaller than the number of positive instances (2164) in restaurant dataset, while the distribution is nearly even in laptop dataset. So for restaurant data, we also conducted another controlled experiment which doubled the amount of negative instances of restaurant dataset. Table 8 shows the preliminary experimental results on the doubled negative training data. It illustrates that the F-neg increases a little but the overall accuracy without any improvement even slightly decreases after doubling the negative instances. This result is beyond our expectation but

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

no further deep analysis has been done so far.

Strategy	F-pos(%)	F-neg(%)	F-neu(%)	Acc(%)
Double	80.28	**55.11**	19.22	65.48
No double	**80.71**	52.61	**34.51**	**67.18**

Table 8: The results of controlled experiment on restaurant dataset (MaxEnt).

4.3.2 Results on Test Data

Based on above results on training data, our final system used all provided training data for both domains. The MaxEnt algorithm is used for our final system. Table 9 shows our results alone with the top two systems results released by organizers.

Our final results ranked the 12th on the laptop dataset and the 14th on the restaurant dataset. On one hand, the accuracy in restaurant dataset is higher than laptop dataset for the possible reason that the data size of restaurant dataset is much bigger than that of laptop dataset. On the other hand, our results ranked middle in both datasets. Since we utilized eight contextual words around aspect to extract features and it may bring some noise.

Dataset	laptop	restaurant
our system	61.16	70.72
rank 1 system	70.49	80.95
rank 2 system	66.97	80.16

Table 9: The Accuracy (%) of our system and the top two systems on test dataset in subtask 2.

5 Aspect Category Sentiment Polarity System

The aspect category sentiment polarity classification task is also only applicable to restaurant domain. For this task, we adopted the bag-of-sentiment_words representation, extracted sentiment features and used the supervised machine learning algorithms to determine the sentiment orientation of each category.

5.1 Features

To extract features, we firstly used eight sentiment lexicons mentioned in Section 4.1.1 to build a big sentiment words dictionary. Then we extracted all aspect words and all sentiment words in training set as features. In the training and test data, we used the sentiment polarity score of sentiment word and the presence or absence of each aspect term as their feature values.

5.2 Classification Algorithms

The MaxEnt algorithm implemented in Mallet (McCallum, 2002) with default parameters is used to build a polarity classifier.

5.3 Experiment and Results

We used all features and the maximum entropy algorithm to conduct our final system. Table 10 lists the final results of our submitted system along with top two systems.

As shown in Table 10, the accuracy of our system is 0.63 while the best result is 0.83. The main reason is that the features we used are quite simple. For the future work, more sufficient features are examined to help classification.

6 Conclusion

In this work we proposed a combination of NP and NER method and multiple features for aspect extraction. And we also used multiple features including eight sentiment lexicons for aspect and category sentiment classification. Our final systems rank above average in the four subtasks. In future work, we would expect to improve the recall of aspect terms extraction by extending name lists using external data and seek other effective features such as discourse relation, syntactic structure to improve the classification accuracy.

Systems	our system	rank 1 system	rank 2 system
Acc(%)	63.41	82.93	78.15

Table 10: The accuracy of our system and the top two systems of subtask 4 on test dataset

Acknowledgements

The authors would like to thank the organizers and reviewers for this interesting task and their helpful suggestions and comments. This research is supported by grants from National Natural Science Foundation of China (No.60903093) and Shanghai Knowledge Service Platform Project (No. ZF1213).

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *LREC*, volume 10, pages 2200–2204.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Namrata Godbole, Manja Srinivasaiah, and Steven Skiena. 2007. Large-scale sentiment analysis for news and blogs. *ICWSM*, 7.

Suin Kim, Jianwen Zhang, Zheng Chen, Alice Oh, and Shixia Liu. 2013. A hierarchical aspect-sentiment model for online reviews. In *Proceedings of AAAI*.

Sudheer Kovelamudi, Sethu Ramalingam, Arpit Sood, and Vasudeva Varma. 2011. Domain independent model for product attribute extraction from user reviews using wikipedia. In *IJCNLP*, pages 1408–1412.

Lun-Wei Ku, Yu-Ting Liang, and Hsin-Hsi Chen. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, volume 100107.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Chong Long, Jie Zhang, and Xiaoyan Zhut. 2010. A review selection approach for accurate feature rating estimation. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 766–774. Association for Computational Linguistics.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 339–348. Association for Computational Linguistics.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Haris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. in proceedings of the 8th international workshop on semantic evaluation (semeval 2014). *Dublin, Ireland*.

Jason DM Rennie. 2001. *Improving multi-class text classification with naive Bayes*. Ph.D. thesis, Massachusetts Institute of Technology.

Zhiqiang Toh, Wenting Wang, Man Lan, and Xiaoli Li. 2012. An ner-based product identification and lucene-based product linking approach to cprod1 challenge: Description of submission system to cprod1 challenge. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 869–871. IEEE.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational linguistics*, 35(3):399–433.

Tian Tian Zhu, Fang Xi Zhang, and Man Lan. 2013. Ecnucs: A surface information based system description of sentiment analysis in twitter in the semeval-2013 (task 2). *Atlanta, Georgia, USA*, page 408.