DLIREC: Aspect Term Extraction and Term Polarity Classification System

Zhiqiang Toh
Institute for Infocomm Research
1 Fusionopolis Way
Singapore 138632
ztoh@i2r.a-star.edu.sg

Wenting Wang
magicwwt@gmail.com

Abstract

This paper describes our system used in the Aspect Based Sentiment Analysis Task 4 at the SemEval-2014. Our system consists of two components to address two of the subtasks respectively: a Conditional Random Field (CRF) based classifier for Aspect Term Extraction (ATE) and a linear classifier for Aspect Term Polarity Classification (ATP). For the ATE subtask, we implement a variety of lexicon, syntactic and semantic features, as well as cluster features induced from unlabeled data. Our system achieves state-of-the-art performances in ATE, ranking 1st (among 28 submissions) and 2rd (among 27 submissions) for the restaurant and laptop domain respectively.

1 Introduction

Sentiment analysis on document and sentence level no longer fulfills user's needs of getting more accurate and precise information. By performing sentiment analysis at the aspect level, we can help users gain more insights on the sentiments of the various aspects of the target entity. Task 4 of SemEval-2014 provides a good platform for (1) aspect term extraction and (2) aspect term polarity classification.

For the first subtask, we follow the approach of Jakob and Gurevych (2010) by modeling term extraction as a sequential labeling task. Specifically, we leverage on semantic and syntactic resources to extract a variety of features and use CRF as the learning algorithm. For the second subtask, we simply treat it as a multi-class classification problem where a linear classifier is learned to predict the polarity class. Our system achieves good performances for the first subtask in both domains, ranking 1st for the restaurant domain, and 2nd for the laptop domain.

The remainder of this paper is structured as follows: In Section 2, we describe our ATE system in detail, including experiments and result analysis. Section 3 describes the general approach of our ATP system. Finally, Section 4 summarizes our work.

2 Aspect Term Extraction

This subtask is to identify the aspects of given target entities in the restaurant and laptop domains. Many aspect terms in the laptop domain contains digits or special characters such as "17 inch screen" and "screen/video resolution"; while in the restaurant domain, aspects in the sentences are specific for a type of restaurants such as "pizza" for Italian restaurants and "sushi" for Japanese restaurants.

We model ATE as a sequential labeling task and extract features to be used for CRF training. Besides the common features used in traditional Named Entity Recognition (NER) systems, we also utilize extensive external resources to build various name lists and word clusters.

2.1 Preprocessing

Following the traditional BIO scheme used in sequential labeling, we assign a label for each word in the sentence, where "B-TERM" indicates the start of an aspect term, "I-TERM" indicates the continuation of an aspect term, and "O" indicates not an aspect term.

All sentences are tokenized and parsed using the Stanford Parser[1]. The parsing information is used

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

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 235–240,
Dublin, Ireland, August 23-24, 2014.

to extract various syntactic features (e.g. POS, head word, dependency relation) described in the next section.

2.2 General (or Closed) Features

In this section, we describe the features commonly used in traditional NER systems. Such features can easily be extracted from the training set or with the help of publicly available NLP tools (e.g. Stanford Parser, NLTK, etc).

2.2.1 Word

The string of the current token and its lowercase format are used as features. To capture more context information, we also extract the previous and next word strings (in original format) as additional word features.

2.2.2 POS

The part-of-speech (POS) tag of the current token is used as a feature. Since aspect terms are often nouns, the POS tag provides useful information about the lexical category of the word, especially for unseen words in the test sentences.

2.2.3 Head Word

This feature represents the head word of the current token. If the current token does not have a head word, the value "null" is used.

2.2.4 Head Word POS

This feature represents the POS of the head word of the current token. If the current token does not have a head word, the value "null" is used.

2.2.5 Dependency Relation

From the dependency parse, we identify the dependency relations of the current token. We extract two different sets of strings: one set contains the relation strings (e.g. "amod", "nsubj") where the current token is the governor (i.e. head) of the relation, the other set contains the relation strings where the current token is the dependent of the relation. For each set, we only keep certain relations: "amod", "nsubj" and "dep" for the first set and "nsubj", "dobj" and "dep" for the second set. Each set of strings is used as a feature value for the current token, resulting in two separate features.

2.2.6 Name List

Name lists (or gazetteers) have proven to be useful in the task of NER (Ratinov and Roth, 2009). We create a name list feature that uses the name lists for membership testing.

For each domain, we extract two high precision name lists from the training set. For the first list, we collect and keep those aspect terms whose frequency counts are greater than c_1. Since an aspect term can be multi-word, we also extract a second list to consider the counts of individual words. All words whose frequency counts greater than c_2 are collected. For each word, the probability of it being annotated as an aspect word in the training set is calculated. Only those words whose probability value is greater than t is kept in the second list. The specified values of c_1, c_2 and t for each domain are determined using 5-fold cross validation.

2.3 Open/External Sources Generated Features

This section describes additional features we use that require external resources and/or complex processings.

2.3.1 WordNet Taxonomy

This feature represents the set of syntactic categories (e.g "noun.food") of the current token as organized in WordNet lexicographer files (Miller, 1995). We only consider noun synsets of the token when determining the syntactic categories.

2.3.2 Word Cluster

Turian et al. (2010) used unsupervised word representations as extra word features to improve the accuracy of both NER and chunking. We followed their approach by inducing Brown clusters and K-means clusters from in-domain unlabeled data.

We used the review text from two sources of unlabeled dataset: the Multi-Domain Sentiment Dataset that contains Amazon product reviews (Blitzer et al., 2007)[2], and the Yelp Phoenix Academic Dataset that contains user reviews[3].

We induce 1000 Brown clusters for each dataset[4]. For each word in the training/testing set, its corresponding binary (prefix) string is used as the feature value.

We experiment with different prefix lengths and use the best settings using 5-fold cross validation.

[2]We used the unprocessed.tar.gz archive found at http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

[3]http://www.yelp.com/dataset_challenge/

[4]Brown clustering are induced using the implementation by Percy Liang found at https://github.com/percyliang/brown-cluster/.

For the laptop domain, we create a Brown cluster feature from Amazon Brown clusters, using prefix length of 5. For the restaurant domain, we created three Brown cluster features: two from Yelp Brown clusters, using prefix lengths of 4 and 8, and the last one from Amazon Brown clusters, using prefix length of 10.

K-means clusters are induced using the `word2vec` tool (Mikolov et al., 2013)[5]. Similar to Brown cluster feature, the cluster id of each word is used as the feature value.

When running the `word2vec` tool, we specially tune the values for word vector size (size), cluster size (classes) and sub-sampling threshold (sample) for optimum 5-fold cross validation performances. We create one K-means cluster feature for the laptop domain from Amazon K-means clusters (size $= 100$, classes $= 400$, sample $= 0.0001$), and two K-means cluster features for the restaurant domain, one from Yelp K-means clusters (size $= 200$, classes $= 300$, sample $= 0.001$), and the other from Amazon K-means clusters (size $= 1000$, classes $= 300$, sample $= 0.0001$).

2.3.3 Name List Generated using Double Propagation

We implement the Double Propagation (DP) algorithm described in Qiu et al. (2011) to identify possible aspect terms in a semi-supervised way. The terms identified are stored in a list which is used as another name list feature.

Our implementation follow the Logic Programming approach described in Liu et al. (2013)[6]. We write our rules in Prolog and use SWI-Prolog[7] as the solver.

We use the seed opinion lexicon provided by Hu and Liu (2004) for both domain[8]. In addition, for the restaurant domain, we augment the opinion lexicon with addition seed opinion words by using the 75 restaurant seed words listed in Sauper and Barzilay (2013). To increase the coverage, we expand this list of 75 words by including related words (e.g. antonym, similar to) in WordNet. The final expanded list contains 551 words.

Besides the seed opinion words, we also use the last word of each aspect term in the training set as a seed aspect word.

The propagation rules we use are modifications of the rules presented in Liu et al. (2013). A total of 11 rules and 13 rules are used for the laptop and restaurant domain respectively. An example of a Prolog rule concerning the extraction of aspect words is stated below:

```
aspect(A) :-
  relation(nsubj, O, A),
  relation(cop, O, _),
  pos(A, P),
  is_noun(P),
  opinion(O).
```

For example, given the sentence "The rice is amazing.", and "amazing" is a known opinion word, we can extract "rice" as a possible aspect word using the rule.

All our rules can only identify individual words as possible aspect terms. To consider a phrase as a possible aspect term, we extend the left boundary of the identified span to include any consective noun words right before the identified word.

2.4 Algorithms and Evaluation

We use the CRFsuite tool (Okazaki, 2007) to train our CRF model. We use the default settings, except for the negative state features (`-p feature.possible_states=1`).

Feature	F1
Word	0.6641
+ Name List	0.7106
+ POS	0.7237
+ Head Word	0.7280
+ DP Name List	0.7298
+ Word Cluster	0.7430
+ Head Word POS	0.7437
+ Dependency Relation	0.7521

Table 1: 5-fold cross-validation performances on the laptop domain. Each row uses all features added in the previous rows.

2.5 Preliminary Results on Training Set

Table 1 and Table 2 show the 5-fold cross-validation performances after adding each feature group for the laptop and restaurant domain respectively. Most features are included in the optimum feature set for both domains, except for WordNet Taxonomy feature (only used in the restaurant domain) and Dependency Relation feature (only used in the laptop domain).

[5] `https://code.google.com/p/word2vec/`
[6] We did not implement incorrect aspect pruning.
[7] `http://www.swi-prolog.org/`
[8] We ignore the polarity of the opinion word.

	laptop			restaurant		
System	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
DLIREC constrained	0.7931	0.6330	0.7041 (C)	0.8404	0.7337	0.7834 (C)
DLIREC unconstrained	0.8190	**0.6713**	0.7378 (U)	0.8535	**0.8272**	**0.8401** (U)
Baseline	0.4432	0.2982	0.3565 (C)	0.5255	0.4277	0.4716 (C)
Ranked 1st	**0.8480**	0.6651	**0.7455** (C)	0.8535	**0.8272**	**0.8401** (U)
Ranked 2nd	0.8190	**0.6713**	0.7378 (U)	**0.8625**	0.8183	0.8398 (C)
Ranked 3rd	0.7931	0.6330	0.7041 (C)	0.8441	0.7637	0.8019 (C)

Table 3: Results of the Aspect Term Extraction subtask. We also indicate whether the system is constrained (C) or unconstrained (U).

Feature	**F1**
Word	0.7541
+ Name List	0.7808
+ POS	0.7951
+ Head Word	0.7962
+ DP Name List	0.8036
+ Word Cluster	0.8224
+ WordNet Taxonomy	0.8252
+ Head Word POS	0.8274

Table 2: 5-fold cross-validation performances on the restaurant domain. Each row uses all features added in the previous rows.

For each domain, we make submissions in both constrained and unconstrained settings. The constrained submission only uses the Word and Name List features, while all features listed in Table 1 and Table 2 are used in the unconstrained submission for the respective domain.

2.6 Results on Test Set

Using the optimum feature set described in Section 2.5, we train separate models for each domain and evaluate them against the SemEval-2014 Task 4 test set[9]. Table 3 presents the official results of our submissions. We also include the official baseline results and the results of the top three participating systems for comparison (Pontiki et al., 2014).

As shown from the table, our system performed well for both domains. For the laptop domain, our system is ranked 2nd and 3rd (among 27 submissions) for the unconstrained and constrained setting respectively. For the restaurant domain, our system is ranked 1st and 9th (among 28 submissions) for the unconstrained and constrained setting respectively.

Our unconstrained submissions for both domains outperformed our constrained submissions, due to a significantly better recall. This indicates the use of additional external resources (e.g. unlabeled data) can improve the extraction performance.

2.7 Further Analysis of Feature Engineering

Table 4 shows the F1 loss on the test set resulting from training with each group of feature removed. We also include the F1 loss when all features are used.

Feature	**laptop**	**restaurant**
Word	0.0260	0.0241
Name List	0.0090	0.0054
POS	-0.0059	-0.0052
Head Word	0.0072	0.0038
DP Name List	0.0049	0.0064
Word Cluster	0.0061	0.0185
WordNet Taxonomy	-	-0.0018
Head Word POS	-0.0040	-0.0011
Dependency Relation	-0.0105	-
All features	-0.0132	0.0014

Table 4: Feature ablation study on the test set. The quantity is the F1 loss resulted from the removal of a single feature group. The last row indicates the F1 loss when all features are used.

Our ablation study showed that a few of our features are helpful in varying degrees on both domains: Word, Name List, Head Word, DP Name List and Word Cluster. However, the use of the rest of the features individually has a negative impact. In particular, we are surprised that the POS and Dependency Relation features are detrimental to the performances, even though our 5-fold

[9]We train each model using only single-domain data.

cross validation experiments suggested otherwise. Another observation we make is that the WordNet Taxonomy feature is actually useful for the laptop test set: including this feature would have improved our laptop unconstrained performance from 0.7378 F1 to 0.7510 F1 (+0.0132), which is better than the top system performance. We also note that our restaurant performance on the test set can potentially be improved from 0.8401 F1 to 0.8454 F1 (+0.0052) if we originally omit the POS feature.

Overall, we see that all the features we proposed are potentially beneficial to the task. However, more thorough feature selection experiments should be conducted to prevent overfitting and to identify the settings (e.g. domain) in which each feature may be useful.

3 Aspect Term Polarity

In this section, we describe a baseline classifier for ATP, where we treat the problem as a multi-class classification problem.

To correctly identify the polarity of an aspect term, it is crucial to know which words within the sentence indicate its sentiment. A general lexicon or WordNet is not sufficient. Thus, we attempt to build the aspect lexicon based on other information such as POS (Sauper and Barzilay, 2013). For example, sentiment words are more likely to be adjectives.

3.1 Features

3.1.1 Aspect Word

This is to model the idea that certain aspects tend to have a particular polarity most of the time. We compute the most frequent polarity of each aspect in the training set. For each aspect instance, the feature corresponding to its most frequent polarity is set to 1.

3.1.2 General Sentiment Word Lexicon

One sentence may express opinions on multiple aspect terms. According to our observations, words surrounding the aspect term tend to be associated with it. Based on the best settings obtained from 5-fold cross validation, we set a window size of 12 words and consider words with the following POS: JJ*, RB*, VB*, DT and NN*[10].

Some sentiment words are consistent across aspects. For example, "great" for positive and "terrible" for negative. On the other hand, some sentiment words are quite distinct between aspects. In certain cases, they may have opposite sentiment meanings for different aspects (Kim et al., 2013). For example, "fast" is positive when describing boot up speed but negative when describing battery life. Therefore, a general sentiment word lexicon is created from the training set.

If a general sentiment word occurs in the surrounding context of the aspect instance, the feature value for the matched sentiment word is 1. Since the training set does not contain every possible sentiment expression, we use synonyms and antonyms in RiTa WordNet[11] to expand the general sentiment word lexicon. The expanded lexicon contains 2419 words for the laptop domain and 4262 words for the restaurant domain.

3.1.3 Aspect-Sentiment Word Pair

Besides general sentiment word lexicon, we also build aspect-sentiment word pair lexicon from the training set. This lexicon contains 9073 word pairs for the laptop domain and 22171 word pairs for the restaurant domain. If an aspect-sentiment word occurs in the surrounding context of the aspect instance, the feature value for the matched aspect-sentiment word pair is 1.

3.2 Experiments and Results

We use LIBLINEAR[12] to train our logistic regression classifier using default settings.

	laptop	**restaurant**
5-fold cross validation	0.6322	0.6704
DLIREC unconstrained	0.3654	0.4233

Table 5: Accuracy of the Aspect Term Polarity subtask.

Table 5 shows the classification accuracy of our baseline system on the training and test set for each domain. The performance drops a lot in the test set as we use very simple approaches to generate the lexicons. This may cause overfitting on the training set. We also observe that in the test set of both domains, more than half of the instances are positive. In the future, we can explore on using more sophisticated ways to build more effective features and to better model data skewness.

[10] NN* is only used in the restaurant domain.

[11] http://www.rednoise.org/rita/wordnet/

[12] http://www.csie.ntu.edu.tw/~cjlin/liblinear/

4 Conclusion

For ATE subtask, we leverage on the vast amount of external resources to create additional effective features, which contribute significantly to the improvement of our system. For the unconstrained setting, our system is ranked 1st (among 28 submissions) and 2rd (among 27 submissions) for the restaurant and laptop domain respectively. For ATP subtask, we implement a simple baseline system.

Our current work focus on implementing a separate term extraction system for each domain. In future, we hope to investigate on domain adaptation methods across different domains. In addition, we will also address the feature sparseness problem in our ATP baseline system.

Acknowledgements

This research work is supported by a research project under Baidu-I^{2}R Research Centre.

References

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic, June.

Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA.

Niklas Jakob and Iryna Gurevych. 2010. Extracting Opinion Targets in a Single and Cross-Domain Setting with Conditional Random Fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1035–1045, Cambridge, MA, October.

Suin Kim, Jianwen Zhang, Zheng Chen, Alice Oh, and Shixia Liu. 2013. A Hierarchical Aspect-Sentiment Model for Online Reviews. In *AAAI Conference on Artificial Intelligence*.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2013. A Logic Programming Approach to Aspect Extraction in Opinion Mining. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 276–283, Nov.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Commun. ACM*, 38(11):39–41, November.

Naoaki Okazaki. 2007. CRFsuite: a fast implementation of Conditional Random Fields (CRFs).

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris 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.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, 37(1):9–27.

Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado, June.

Christina Sauper and Regina Barzilay. 2013. Automatic Aggregation by Joint Modeling of Aspects and Values. *J. Artif. Int. Res.*, 46(1):89–127, January.

Joseph Turian, Lev-Arie Ratinov, and Yoshua Bengio. 2010. Word Representations: A Simple and General Method for Semi-Supervised Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394, Uppsala, Sweden, July.