

INSIGHT_Galway: Syntactic and Lexical Features for Aspect Based Sentiment Analysis

Sapna Negi

Insight Centre for Data Analytics
National University of Ireland
Galway
{sapna.negi, paul.buitelaar}@insight-centre.org

Paul Buitelaar

Insight Centre for Data Analytics
National University of Ireland
Galway
{sapna.negi, paul.buitelaar}@insight-centre.org

Abstract

This work analyses various syntactic and lexical features for sentence level aspect based sentiment analysis. The task focuses on detection of a writer's sentiment towards an aspect which is explicitly mentioned in a sentence. The target sentiment polarities are positive, negative, conflict and neutral. We use a supervised learning approach, evaluate various features and report accuracies which are much higher than the provided baselines. Best features include unigrams, clauses, dependency relations and SentiWordNet polarity scores.

1 Introduction

The term *aspect* refers to the features or aspects of a product, service or topic being discussed in a text. The task of detection of sentiment towards these aspects involves two major processing steps, identifying the aspects in the text and identifying the sentiments towards these aspects. Our work describes a submitted system in the *Aspect Based Sentiment Analysis* task of SemEval 2014 (Pontiki et al., 2014). The task was further divided into 4 subtasks; our work corresponds to the subtask 2, called *Aspect Term Polarity Detection*. We predict the polarity of sentiments expressed towards the aspect terms which are already annotated in a sentence. The target polarity types are *positive*, *negative*, *neutral* and *conflict*.

We employ a statistical classifier and experiment with various syntactic and lexical features. Selected features for the submitted system include words which hold certain dependency relations with the aspect terms, clause in which the aspect

term appears, unigrams, and sum of lexicon based sentiment polarities of the words in the clause.

2 Related Work

Pang et al. (2002) proved that unigrams and bigrams, adjectives and part of speech tags are important features for a machine learning based sentiment classifier. Later, verbs and adjectives were also identified as important features (Chesley, 2006). Meena and Prabhakar (2007) performed sentence level sentiment analysis using rules based on clauses of a sentence. However, in our case we cannot simply consider the adjectives and verbs as features, since they might relate to different aspects. For example, in the sentence 'The pizza is the best if you like thin crusted pizza.', sentiment towards 'pizza' is positive because of the adjective 'best'; however for the term 'thin crusted pizza', 'like' would be the sentiment verb. Therefore, only those adjectives and verbs which relate to the target aspect, can be considered as the indicator of their polarity. Wilson et al. (2009) showed that the words which share certain dependency relations with aspect terms, tend to indicate the sentiments expressed towards those terms.

Saif et al. (2012) showed the co-relation between topic and sentiment polarity in tweets, and asserted that majority of people tend to express similar sentiments towards same topics, especially in the case of positive sentiments. The baseline approach for this task (Pontiki et al., 2014) also associates polarity with aspect terms. Therefore, we also consider aspect term as a potential feature. Our approach for this task is based on our observation of the data, with a provenance of the above mentioned findings.

3 Approach

We employ a statistical classifier which trains on the provided training datasets.

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Page numbers and proceedings footer are added by the organisers. Licence details: <http://creativecommons.org/licenses/by/4.0/>

Datasets: Training datasets comprise of 3000 sentences from laptop and restaurant reviews. Training sentences were tagged with the target aspect term and the corresponding polarity, where more than one aspect term can be tagged in a sentence.

3.1 Feature Sets

We divide the candidate features into four feature sets.

1. **Non-contextual:** These features comprise of training vocabulary. They do not target aspect based sentiments, but the overall sentiment of the sentence. There might be cases where the aspect based sentiment is same as the overall sentiment of the sentence. The feature set comprises of three feature types, unigrams, bigrams, adjectives and verbs of the sentence.
2. **Lexicon Non-Contextual:** These features are the Sentiwordnet_v3.0 polarity scores (Andrea Baccianella and Sebastiani, 2010) of the words obtained from the best non-contextual feature type. This feature set would include two numerical features, positive polarity score and negative polarity score of the best non-contextual feature types. Best non-contextual feature type is decided by comparing the classification accuracies of individual feature types, with cross validation on the training data (Table 1). We evaluated two algorithms to obtain the positive and negative polarities of words using SentiWordNet. Later, we would provide details of these algorithms.
3. **Contextual:** These features target aspect based sentiments. Feature types comprise of the clause in which an aspect term appears, the adjective and verbs of this clause, aspect term itself, and the words which hold certain dependency relations with aspect term. We only considered the Stanford parser dependencies ‘nn’, ‘amod’, and ‘nsubj’. The dependency relations were chosen on the basis of best classification accuracy in a cross validation trial, where the only features were the words holding different dependency relations with the aspect term. However, we only list the accuracy from the best performing dependency relations in the Tables 1, 3. By the fea-

ture type *clause*, we mean the unigrams contained in a clause.

4. **Lexicon Contextual:** Similar to Lexicon Non-Contextual features, these are the numeric values obtained from SentiWordNet polarity scores of the best performing contextual feature type.

Polarity Calculation using SentiWordNet:

WordNet (Fellbaum, 1998) is a lexical database for the English language. It assigns each listed word the senses to which it may belong, where each unique sense is represented by a synset id. SentiWordNet is built on the top of WordNet, where a pair of positive and negative polarity score is assigned to each sense of a word. SentiWordnet entry for each word comprises of all the possible parts of speech in which the word could appear, all the senses corresponding to each part of speech, and a pair of polarity scores associated with each sense¹. The magnitude of positive and negative polarity scores for each sense ranges from 0 to 1.

In order to automatically obtain the polarity scores corresponding to the desired sense of a word, word sense disambiguation is required to be performed. We did not perform sense disambiguation, and picked the polarity scores simply on the basis of word and part of speech matching. This gives more than one candidate senses, and thus more than one pair of polarity scores for each word. We evaluated the following 2 methods to assign single values of sentiment polarity scores to each word.

1. **Default:** The SentiWordnet website² provides a basic algorithm to assign sentiwordnet based polarities to a word. SentiWordnet assigns a rank to each sense of a word, where most commonly appearing sense is ranked as 1. The default algorithm first calculates an overall polarity (Positive score - Negative score), for each sense of a word. It then calculates a weighted sum of the overall polarity scores of all the senses of a word, where the weights are the ranks of senses. This sum is considered as a single value polarity score of a word, which can be a positive or negative number.

¹<http://sentiwordnet.isti.cnr.it/search.php?q=good>

²<http://sentiwordnet.isti.cnr.it>

2. **Our algorithm:** We do not obtain an overall polarity score for each word, but we obtain a pair of aggregated negative and positive score for each word. Aggregate positive score is obtained by taking the average of the positive scores of each sense of the word, and same goes for the aggregate negative score.

One reason for keeping the positive and negative scores separate in our algorithm is that the task also involves sentiment classes *conflict* and *neutral*. Using only the overall polarity score results in a loss of information in the case of very low negativity and positivity (neutral sentiments), or high but comparable negativity and positivity (conflicting sentiments). Also, our algorithm produced better results when used with an SVM classifier, with features as unigrams and their polarity scores.

3.2 Classifier Model

Our system is built on the state of the art LibSVM classifier (EL-Manzalawy and Honavar, 2005). We used Weka 3.7.10 toolkit (Hall et al., 2009) for our experiments. The parameters³ of the SVM classifier are tuned to the values which give best results with unigrams. Table 2 provides the tuned parameters, rest of the parameters are set at default values.

Pre-processing: We perform stemming using Weka’s implementation of Snowball stemmer, convert strings to lower case and filter out stopwords. We use a customised list of stopwords, based on our observations of the data. The customised list is created using the stopword list of Weka, with certain words removed. For example, negators like ‘not’, ‘didn’t’ etc. are important for negative sentiments, for example ‘I can barely use any usb devices because they will not stay properly connected’. Words like ‘but’, ‘however’ are prominent in conflicting sentiments, for example ‘No backlit keyboard, but not an issue for me’. Tables 1, 3 show the difference in results on using filtered stopword list, compared against no stopword removal, and original stopword list.

G	R	C	E	Z
0.10	1.0	2	1.0	normalise

Table 2: Parameter Settings for SVM Classifier.

³<http://weka.sourceforge.net/doc.stable/weka/classifiers/functions/LibSVM.html>

3.3 Feature Evaluation

We evaluated our features using 8-fold cross validation on the training data. We evaluated each feature by using it as the only feature for the classifier (Tables 1, 3). We performed experiments on different combinations of features, but we only present the best performing combination of features in the last row of the tables. The baseline approach (Pontiki et al., 2014) provided by the organisers, produced an accuracy of 47% for laptop and 57% for restaurant, by splitting the training data.

Metrics include, F score for each class, and overall classification accuracy. F score ranges from 0-1, and overall accuracy range from 0-100.

4 Submission and Results

Submission involved the prediction of sentiment polarity towards the already tagged aspect terms in two test datasets. There were 800 sentences in each test dataset. The laptop test dataset was obtained by dividing the original laptop data into training and test. However, restaurant test dataset and training dataset come from different sources.

We trained our classifier using the provided training dataset and the highlighted features (last row) in the Tables 1, 3. In order to evaluate the submission, gold standard datasets corresponding to each test dataset were later released, and submission’s accuracy was compared against it.

Results: The system performance was evaluated and ranked on the basis of overall accuracy of sentiment prediction. We were ranked as 20/32 for the laptop domain, and 16/34 for the restaurant domain. The task organisers reported that 8 polarity predictions for laptop data, and 34 for restaurant data were missing from our submission. We later debugged our system, and obtained the actual accuracy which our system is capable of producing with the given test data. The results are summarised in Table 4.

5 Observations and Analysis

We hypothesize that aspect terms should serve as features when training data and test data come from same source, which means that they relate to the same brand, product, service etc. This is because aspect terms change with data, for example names of dishes would change with different restaurants even if the domain is same. In our case, the laptop test data was obtained from the

Feature Set	Features	Positive	Negative	Conflict	Neutral	Accuracy
non-contextual	unigrams,bigrams	0.827	0.590	0.210	0.422	70.699
	unigrams	0.830	0.584	0.154	0.413	70.962
	adjectives,verbs	0.704	0.412	0.000	0.257	63.465
	adjectives	0.623	0.430	0.000	0.000	56.410
non-contextual + lexicon	unigrams, unigram polarity scores	0.833	0.596	0.154	0.414	71.300
contextual	clause	0.823	0.571	0.117	0.456	71.170
	adjective, verbs within clause	0.784	0.472	0.000	0.257	66.465
	aspects	0.734	0.154	0.000	0.264	59.442
	dependencies	0.751	0.235	0.000	0.061	61.257
contextual + lexicon	clause, clause polarity score	0.735	0.000	0.000	0.000	58.101
combined	unigrams, clause, dependencies, clause polarity score, filtered stopword list	0.837	0.610	0.162	0.418	71.960
	used original stopword	0.825	0.587	0.078	0.371	70.830
	no stopwords used	0.830	0.610	0.151	0.435	72.000

Table 1: Feature Analysis for Restaurant Reviews.

Feature Set	Features	Positive	Negative	Conflict	Neutral	Accuracy
Non-Contextual	unigrams,bigrams	0.827	0.590	0.210	0.422	70.699
	unigrams	0.781	0.747	0.110	0.484	71.202
	adjectives,verbs	0.569	0.620	0.000	0.164	54.516
	adjectives	0.521	0.613	0.000	0.090	51.230
non-contextual + lexicon	unigrams, unigram polarity scores	0.783	0.754	0.179	0.529	71.850
Contextual	Clause	0.823	0.571	0.117	0.456	71.170
	adjective, verbs within clause	0.569	0.620	0.000	0.164	54.510
	aspects	0.602	0.259	0.000	0.050	45.240
	dependencies	0.590	0.078	0.000	0.000	42.480
contextual + lexicon	clause, clause polarity score	0.750	0.705	0.000	0.407	67.230
combined	unigrams, clause, dependencies, clause polarity score, filtered stopword list	0.786	0.752	0.100	0.498	71.600
	weka stopword list	0.780	0.744	0.113	0.442	70.590
	no stopwords	0.782	0.758	0.154	0.530	72.170

Table 3: Feature Analysis for Laptop Reviews.

same dataset which was used to prepare training data, while restaurant was from a different source. We observed that, although aspect terms produced better results with cross validation, it did not happen in the case of test data. The restaurant test data produced better accuracy without aspect term features, while laptop test data produced better accuracy with aspect term features. We submitted our systems without using aspect terms as features. If aspect terms were used as features, the laptop test data would have been classified with an accuracy of 60.8 %. Another interesting observation is, unigrams produce better results on their own, as compared to adjectives and verbs. Dependency and clauses also seem to be very important features, since they produce an accuracy of above 60% on their own. We also observed that some stopwords are important features for this task, and complete removal of stopwords lowers the classification accuracy.

Domain	Baseline	Best System	Submitted System	Debugged System
laptop	51.07	70.48	57.03	59.15
restaurant	64.28	80.95	70.70	71.44

Table 4: Results on Gold Standard Data.

6 Conclusion

We presented an analysis and evaluation of syntactic and lexical features for performing sentence level aspect based sentiment analysis. Our features depend on part of speech tagging and dependency parsing, and therefore the accuracy might vary with different parsers. Although our system did not produce the highest accuracy for the task, it is capable of achieving accuracies much above the baselines. Therefore, the proposed features can be worth testing on different datasets and can be used in combination with other features.

Acknowledgement

This work has been funded by the European project EUROSENTIMENT under grant no. 296277, and the Science Foundation Ireland under Grant Number SFI/12/RC/2289 (Insight Center).

References

Stefano Esuli Andrea Baccianella and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*

- (LREC'10). European Language Resources Association (ELRA).
- Paula Chesley. 2006. Using verbs and adjectives to automatically classify blog sentiment. In *In Proceedings of AAAI-CAAW-06, the Spring Symposium on Computational Approaches*, pages 27–29.
- Yasser EL-Manzalawy and Vasant Honavar, 2005. *WLSVM: Integrating LibSVM into Weka Environment*.
- Christiane Fellbaum, editor. 1998. *WordNet An Electronic Lexical Database*. The MIT Press.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software, an update. *SIGKDD Explorations*, 11:10–18.
- Arun Meena and Prabhakar T.V. 2007. Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In *ECIR*, volume 4425 of *Lecture Notes in Computer Science*. Springer.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10, EMNLP '02*, pages 79–86, Stroudsburg, PA, USA. Association for Computational Linguistics.
- 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 International Workshop on Semantic Evaluation, SemEval 2014*, Dublin, Ireland.
- Hassan Saif, Yulan He, and Harith Alani. 2012. Semantic sentiment analysis of twitter. In *International Semantic Web Conference (1)*, volume 7649 of *Lecture Notes in Computer Science*, pages 508–524. Springer.
- Wilson Theresa, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, pages 399–433.