Sentiment Analysis of Movie Reviews on Discussion Boards using a Linguistic Approach

Tun Thura Thet, Jin-Cheon Na, Christopher S.G. Khoo, Subbaraj Shakthikumar
Wee Kim Wee School of Communication and Information
Nanyang Technological University
31 Nanyang Link, Singapore 637718
{ut0001et, tjcna, assgkhoo,ssubbaraj}@ntu.edu.sg

ABSTRACT

We propose a linguistic approach for sentiment analysis of message posts on discussion boards. A sentence often contains independent clauses which can represent different opinions on the multiple aspects of a target object. Therefore, the proposed system provides clause-level sentiment analysis of opinionated texts. For each sentence in a message post, it generates a dependency tree, and splits the sentence into clauses. Then it determines the contextual sentiment score for each clause utilizing grammatical dependencies of words and the prior sentiment scores of the words derived from SentiWordNet and domain specific lexicons. Negation is also delicately handled in this study, for instance, the term "*not superb*" is assigned a lower negative sentiment score than the term "*not good*". We have experimented with a dataset of movie review sentences, and the experimental results show the effectiveness of the proposed approach.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *text analysis*; H.2.8 [**Database Management**]: Database Application – *data mining*

General Terms

Algorithms, Experimentation

Keywords

Sentiment analysis, movie reviews, dependency tree, discussion board

1. INTRODUCTION

With the recent proliferation of Web 2.0 applications, users now have enormous opportunities to express opinions and share experiences on the Internet. This trend has drawn the attention of organizations, businesses, and researchers around the world who are interested in opinions expressed by people on various topics.

We investigate sentiment analysis of movie review documents in this study mainly due to the challenging nature of such reviews. Our approach makes use of SentiWordNet [2] to assign initial values (prior sentiment score) for each word in a sentence. We also use a set of domain specific lexicons to improve the accuracy of sentiment analysis. The sentence is grammatically processed and divided into separate independent clauses. Then, the contextual sentiment score of each clause focusing on individual aspects is processed.

The rest of this paper is organized as follows. Section 2 surveys related work and section 3 discusses our approach for clause level sentiment analysis. After presenting our experimental results in Section 4, we conclude the paper in Section 5.

2. RELATED WORK

Most works on sentiment analysis focus on polarity classification. The overall sentiment of a given text is classified into two opposing sentiment classes. Pang et al. [5] used machine learning techniques for the classification of documents by overall sentiment. They employed three machine learning methods: Naïve Bayes, maximum entropy classifications and support vector machines (SVM) [3]. The results produced by machine learning methods are better in comparison to the human generated baselines. Another work by Pang and Lee [6] proposed a novel machine-learning method by applying text categorization techniques only to the subjective portions of the document. They showed that employing the minimum-cut framework for subjectivity detection can compress reviews into much shorter extracts and still retain polarity information which is comparable to that of the full review.

A deeper linguistic analysis of syntactic relations within a feature sets can be used for sentiment analysis. Kudo and Matsumoto [4] have proposed a sentence-level sentiment classification approach using subtree-based boosting algorithm to capture sub-structures embedded in texts. They have experimented the approach with semi-structured texts, and their method outperformed standard bag-of-words approaches. Wiebe and Riloff [8] studied how syntactic patterns can be effectively used for subjective detection which is a prior step to sentiment classification. Yi et al. [9] proposed another method of sentiment analysis using natural language processing techniques to extract positive and negative sentiments for specific subjects from a document, instead of classifying the whole document into positive or negative. They used semantic analysis with a syntactic parser and sentiment lexicons. The prototype system was experimented with online review and news articles. Shaikh et al. [7] has proposed a sentence level sentiment analysis approach. A linguistic tool called SenseNet was developed for domain independent sentiment analysis and visualization of the results. The approach analyses the semantic verb frames of a sentence and performs semantic dependency analysis to calculate the contextual valence of the whole sentence.

Many studies have been carried out on sentiment analysis using machine learning and linguistic approaches. In our study, sentiment analysis is performed at clause level so that different opinions on multiple aspects can be processed separately for a

TSA'09, November 6, 2009, Hong Kong, China.

given sentence. In addition, more refined sentiment score calculation methods are also used to determine how one clause can be more positive or negative than the others.

3. Clause Level Sentiment Analysis

3.1 Overview

Generally, there are three types of sentences in the English language: simple sentence, compound sentence and complex sentence. Simple sentences contain only one independent clause whereas compound sentences contain two or more independent clauses, and complex sentences contain at least one independent clause and one dependent clause. The structure of a clause contains two parts, a subject and a predicate, where predicate is a combination of verb, object, complement, and adverbial.

Figure 1 shows the overview of the algorithm used for sentiment analysis of review texts at clause level processing. Firstly, for each sentence of the review texts, semantic annotation is performed, and prior sentiment scores are assigned for each word. Then, the grammatical dependencies are generated, and the sentence is broken into independent clauses. For each clause, the contextual sentiment score is calculated by processing the dependency tree based on its clause structure. The review aspect of the clause is determined by looking at the occurrence of the feature words.

```
FOR each sentence
    Perform Semantic Annotation
    Assign prior sentiment scores to words
    Generate grammatical dependencies
    Break a sentence into clauses
    FOR each clause
        Calculate clause level Sentiment Score (Single Aspect)
            Subject or Object (Adjective + Noun)
            Verb phrase (Adverb + Verb)
            Predicate (Verb phrase + Object/Complement)
            Clause (Subject and Predicate)
        Determine Aspect
    END FOR
END FOR
```

Figure 1. Algorithm for clause level sentiment analysis

3.2 Semantic Annotation

The movie names (e.g., "*beautiful mind*" and "*fantastic four*") often contain sentiment words. However, these words (i.e., 'beautiful' and 'fantastic') should not be processed for sentiment analysis. The objective of semantic annotation is to tag the data such as the movie, director, cast, and character names, so that they will be ignored during the processing. The data about each movie (i.e. movie, director, cast, and character names) are collected from movie review websites and stored in a movie specific feature list. In addition, the generic words such as '*directing*', '*animation*', '*scene*', '*music*', and '*sound effects*' are also prepared and stored in a generic feature list. Annotation is performed by tagging the longest matching terms found in the generic and movie specific feature lists.

3.3 Prior Sentiment Scores

The prior sentiment score of the words ranges between -1 and +1, with 0 being neutral. The values are derived from SentiWordNet and domain specific lexicons which we have manually prepared. The contextual sentiment score of the clause is calculated based on the prior sentiment scores of the words in a given input sentence.

3.3.1 SentiWordNet

We have used SentiWordNet (http://sentiwordnet.isti.cnr.it/) to derive prior sentiment scores of the standard terms. SentiWordNet is a lexical resource for sentiment analysis, and its sentiment scores are automatically calculated using a semi-supervised method [2]. In this study, we ignore sense disambiguation and we simply use the average score of all the senses of a word as the prior sentiment score of the word.

3.3.2 Domain Specific Lexicons

We have also prepared a set of domain specific sentiment words from a separate dataset for the movie review domain. Although the word '*unpredictable*' is considered negative in a general context, it often reflects positive sentiment for a movie storyline. Some informal words such as '*sucks*' and '*rocks*' are often used in movie reviews and they are added to the domain specific list.

3.4 Dependency Tree

We have used the Stanford NLP library [1] to process grammatical relationships of words in a sentence. The Stanford typed dependencies are binary grammatical relationships between two words: a governor and a dependent. The whole sentence is represented by the entire tree, which can be divided into subtrees, each representing a clause focusing on individual aspect. Division of a dependency tree (the whole sentence) into subtrees (clauses) is performed by one of the two methods: splitting or multiplying. A compound sentence with two or more independent clauses with complete clause structure can be divided by splitting them directly. For example, the sentence "*I love Tom but I hate the movie*" can be simply divided into "*I love Tom*" (cast aspect) and "*I hate the movie*" (overall aspect) as shown in Figure 2 so that their sentiment can be analyzed more accurately for each aspect.

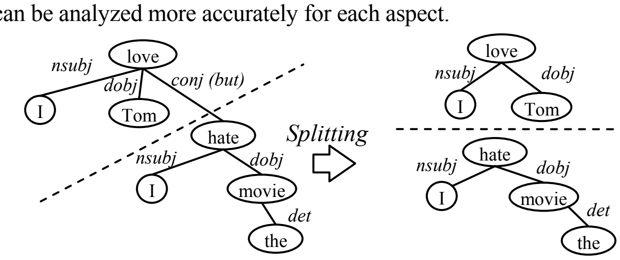

Figure 2. Splitting a dependency tree

For the structures of sentences with compound predicates or compound objects, the tree needs to be multiplied since they cannot be split directly. For example, the sentence "*I love the music but not the story*" needs to be multiplied into "*I love the music*" and "*negation(I love the story)*" as shown in Figure 3, so that their sentiment can be analyzed more accurately.

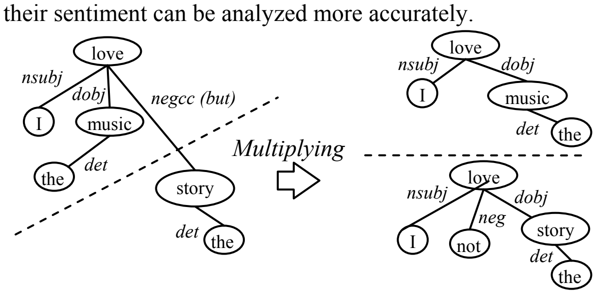

Figure 3. Multiplying a dependency tree

3.5 Contextual Sentiment Score

The contextual sentiment score is calculated by traversing and processing each node in the dependency tree based on the type of grammatical relations between parent and children, and parts of speech (POS) information. The algorithm is recursive and it processes until it reaches all the leaf nodes. The calculation methods and formulas for processing contextual sentiment scores will be discussed in the following subsections.

3.5.1 Subject or Object

Table 1. Contextual sentiment for subject or object

ID	(A)djective	(N)oun	Output	Examples
F1.1	+ / 0*	+ / 0	+	good portrayal
F1.2	+ / 0	-	-	biggest crap
F1.3	-	+ / 0	-	lousy performance
F1.4	-	-	-	worst disaster

*: 0 indicates neutral.

Formula

F1.1) Positive A and Positive N (positive includes neutral)
=> + (abs(N) + (1-abs(N)) * abs(A))
E.g. +0.6 and +0.5 => + (0.5 + (1-0.5) * 0.6) = +0.8
F1.2) Positive A and Negative N => - (abs(N) + (1-abs(N)) * abs(A))
E.g. +0.6 and -0.5 => - (0.5 + (1-0.5) * 0.6) = -0.8
F1.3) Negative A and Positive N => - abs(A) [or the value A]
E.g. -0.6 and +0.5 => - 0.6
F1.4) Negative A and Negative N => - (abs(N) + (1-abs(N)) * abs(A))
E.g. -0.6 and -0.5 => - (0.5 + (1-0.5) * 0.6) = -0.8

The subject or object of a clause can be a noun phrase which consists of a noun and an adjective. When inputs (i.e. adjective and noun) are of the same sentiment orientation (F1.1 and F1.4), they tend to intensify each other. The absolute value of the output should be larger than the absolute values of the two inputs but less than 1. Therefore, the formula (abs(N) + (1-abs(N))*abs(A)) is applied. When the adjective is positive and noun is negative (i.e., F1.2), the adjective still intensifies the noun, but when the adjective is negative and noun is positive (i.e. F1.3), the output is the value of the negative adjective.

3.5.2 Verb Phrase

A verb phrase can contain a verb and an adverb. Similar formulas are used for F2.1, F2.2, F2.3, and F2.4. For the negating adverb such as '*hardly'* and '*rarely'*, the negation method (F2.5) which will be discussed later (F6.1 and F6.2) is used.

Table 2. Contextual sentiment for verb phrase

ID	(A)dverb	(V)erb	Output	Examples
F2.1	+ / 0	+ / 0	+	portrayed accurately
F2.2	+ / 0	-	-	always freak
F2.3	-	+ / 0	-	performed poorly
F2.4	-	-	-	badly ruined
F2.5	- -	- +	+ -	hardly fail rarely good *negation

Formula

F2.5) Negative A and Negative V => + (abs(A) * (1-abs(V))
E.g. -0.6 and -0.5 => + (0.6 * (1-0.5)) = +0.3
Negative A and Positive V => - (abs(A) * (1-abs(V))
E.g. -0.6 and +0.5 => - (0.6 * (1-0.5)) = -0.3

3.5.3 Predicate

Table 3. Contextual sentiment for predicate

ID	(V)erb Phrase	(O)bject/ Complement	Output	Examples
F3.1	+ / 0	+ /0	+	provided good portrayal
F3.2	+ / 0	-	-	created disaster
F3.3	-	+ / 0	-	spoiled the show
F3.4	-	-	-	suffers a severe problems
F3.5	- -	- +	+ -	ceased boring stopped winning *negation

Formula

F3.5) Negative V and Negative O => + (abs(V) * (1-abs(O)))
E.g. -0.6 and -0.5 => + (0.6 * (1-0.5)) = +0.3
Negative V and Positive O => - (abs(V) * (1-abs(O)))
E.g. -0.6 and +0.5 => - (0.6 * (1-0.5)) = -0.3

A predicate can contain a verb phrase and an object or complement. Similar formulas are used for F3.1, F3.2, F3.3, and F3.4. For the negating verbs such as '*stop'* and '*cease'*, the negation method (F3.5) is used.

3.5.4 Clause

A clause contains a subject and a predicate. Similarly, when inputs are of the same sentiment orientation, they intensify each other (F4.1 and F4.4). However, when the inputs are of different sentiment orientations (F4.2 and F4.3), a different method is applied. The inputs values are compared, and the output is the input value having a greater absolute value.

Table 4. Contextual sentiment for clause

ID	(S)ubject	(P)redicate	Output	Examples
F4.1	+ /0	+ /0	+	The superstar performs perfectly.
F4.2	+ / 0	-	+/-	My favorite movie is out. The superstar performs poorly.
F4.3	-	+ / 0	+/-	This short film was incredible. Disaster started with him.
F4.4	-	-	-	Bad casting spoiled everything.

Formula

F4.2) Positive S and Negative P => If abs(S) > abs(P) Then +abs(S) Else –abs(P)
E.g. +0.6 and -0.5 => +0.6
F4.3) Negative S and Positive P => If abs(S) > abs(P) Then –abs(S) Else +abs(P)
E.g. -0.6 and +0.5 => -0.6

3.5.5 Complex-To Clause

Table 5. Contextual sentiment for complex-to

ID	Clause 1	Clause 2	Output	Examples
F5.1	+ / 0	+ / 0	+	I am keen to watch this again
F5.2	+ / 0	-	-	I will advise to throw away this.
F5.3	-	+ / 0	-	It is hard to like this movie.
F5.4	-	-	+	It is hard to find bad things.

Formula

F5.3) Negative C1 and Positive C2 => - (abs(C1) * (1-abs(C2)))
E.g. -0.6 and -0.5 => - (0.6 * (1-0.5)) = -0.3
F5.4) Negative C1 and Negative C2 => + (abs(C1) * (1-abs(C2)))
E.g. -0.6 and -0.5 => + (0.6 * (1-0.5)) = +0.3

For complex sentences with '*to*' dependency, the sentiment score of the second clause is intensified when the first clause is positive (F5.1 and F5.2), and the sentiment score of the second clause is negated when the first clause is negative (F5.3 and F5.4).

3.5.6 Negation of Clause

Handling negation is one of the key processes in sentiment analysis. Negation of a node is processed by looking at the negativity (i.e. negative sentiment score) of the negation word and the original sentiment score before the negation.

Formula

F6.1) Negation N and Positive Score S (e.g., not good) => - (abs(N) * (1 – abs(S)))
E.g. -1 and +0.45 => -1 * (1 – 0.45) = -0.55
F6.2) Negation N and Negative Score S (e.g., not bad) => + (abs(N) * (1 – abs(S)))
E.g. -1 and -0.45 => + (1 * (1 – 0.45)) = +0.55

If '*superb'* is more positive than '*good'*, "*not superb*" will be less negative than "*not good*". On the other hand, if we look at "not good" and "hardly good", the negation adverb *'hardly'* has less negativity than *'not'*.

4. Experiments

4.1 Datasets

We conducted experiments with a dataset of 370 sentences: 185 positive and 185 negative. The movie review texts were collected from the discussion board of a movie review site (www.imdb.com).

Two coders read the sentences, and manually classified the sentiment orientations toward the target aspects. The intercoder agreement using Cohen's kappa coefficient was 0.927. The conflicting labels by the two coders were reviewed and manually re-classified by one of the authors and these manually classified sentiment labels were used as answer keys.

4.2 Experimental Results

Table 6 shows the accuracies for the clause level sentiment analysis measured by comparing the system results to the answer keys (gold standard) prepared manually. Accuracy is relatively high because the dataset contains mostly simple and compound sentences posted on the discussion board.

Table 6. Accuracy for the clause level sentiment analysis

	Review Aspects	# of clauses	Accuracy
1	Overall	104	89%
2	Director	101	66%
3	Casts	100	80%
4	Storyline	44	82%
5	Scene	38	82%
6	Music/Sound	22	91%

Table 7 shows the precision, recall, and F-score for determining the review aspects of the clauses.

Table 7. Precision, Recall and F-Score for determining review aspects

	Review Aspects	Precision	Recall	F-Score
1	Overall	87%	100%	93%
2	Director	99%	96%	97%
3	Casts	100%	93%	96%
4	Storyline	100%	98%	99%
5	Scene	97%	100%	99%
6	Music/Sound	100%	95%	98%

Additionally, to verify the effectiveness of our approach, the system is also experimented with the 300 snippets (sentences) from the polarity dataset introduced in Pang and Lee [6], and achieved 81% accuracy. The polarity snippet dataset contains subjective sentences annotated with positive and negative sentence tags.

4.3 Error Analysis

After carefully analyzing the errors, the sources of errors are categorized into prior sentiment score, algorithm, parser, user and others. Most of the errors come from the wrong assignment of prior sentiment scores to the words. As shown in the following example, the word '*realistic*' suggests positive sentiment although it is a negative word in SentiWordNet.

- *The story is really realistic. (Answer=Positive; Result=Negative)*

In the following examples, the word '*miss'* has different senses. The prior score is negative, and the output is positive because of the negation.

- *You should not miss this movie. (Answer= Positive; Result= Positive)*
- *Bye, we will not miss you! (Answer= Negative; Result= Positive)*

Some of the clauses contain misleading words and the algorithm fails to correctly classify them. In the example below, the term '*difficult'* is misleading and the clause is wrongly classified.

- *A worthy entry into a very difficult genre (Answer= Positive; Result= Negative)*

In some cases, incomplete sentences are parsed incorrectly. As shown in the example, the term 'boring' is wrongly tagged as proper noun (NNP). With wrong POS information, the prior score could not be assigned correctly.

- *Boring Script! (Answer= Negative; Result= Neutral)*

Another source of errors is from the users. Users often make typo errors or grammar mistakes. When spellings are wrong, the prior scores are not assigned correctly.

- *Incompitence of the composer! (Answer= Negative; Result= Neutral)*

In the following example, the movie name, '*collateral damage*', is not tagged properly and thus, and the clause is wrongly classified.

- *Collateral damage finally delivers the goods for Schwarzenegger fans. (Answer= Positive; Result= Negative)*

5. Conclusion

Sentiment analysis of review documents should consider multiple sentiments towards different aspects of the reviewed entity. The proposed approach performs sentiment analysis at clause level. The experimental results show that the proposed approach is effective for sentiment analysis of short documents such as message posts in discussion boards. One of the limitations of this approach is its reliance on SentiWordNet and domain specific lexicons for prior sentiment scores. More evaluations and experiments will be carried out with larger datasets from different domains and genres.

6. REFERENCES

[1] de Marneffe, M.-C., MacCartney, B., and Manning, C. D. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, 2006.

[2] Esuli, A. and Sebastiani, F. Determining term subjectivity and term orientation for opinion mining, In *Proceedings of EACL*, 2006.

[3] Joachims, T. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine-learning*, pp. 137–142, 1998.

[4] Kudo T. and Matsumoto, Y. A boosting algorithm for classification of semi-structured text. In *Proceedings of EMNLP*, 2004.

[5] Pang, B., Lee, L., and Vaithyanathan, S. Thumbs up? Sentiment classification using machine-learning techniques, In *Proceedings of EMNLP*, pp. 79–86, 2002.

[6] Pang, B., and Lee, L. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, pp. 271-278, 2004.

[7] Shaikh, M. A. M., Prendinger, H., and Ishizuka, M. Sentiment assessment of text by analyzing linguistic features and contextual valence assignment. *Applied Artificial Intelligence*, 22 (6), 558-601, 2008.

[8] Wiebe, J. and Riloff, E. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *Proceedings of CICLing*, pp. 486-497, 2005.

[9] Yi, J., Nasukawa, T., Bunescu, R., and Niblack, W. Sentiment Analyzer: Extracting Sentiments about a Given Topic using Natural Language Processing Techniques. In *Proceedings of ICDM*, pp. 427-434, 2003.