Feature-based Customer Review Mining

Jingye Wang Heng Ren
Department of Computer Science
Stanford University
{johnnyw, hengren}@stanford.edu

Abstract

The large amount of the information is a big challenge to the customer's patience to read all the feedbacks. Topical classification and sentimental classification are proposed to be used in information classification. Several machine learning methods, such as the Naive Bayes, Maximum Entropy classification, and Support Vector Machines are good approaches to solve this problem. However, classic sentimental classification does not find what the reviewer liked or disliked. Our approach generalizes an overall rating and user comments on several features for each product. It calculates an overall rating of the product based on PIM-IR algorithm and generalizes these comments on features using feature-based classification.

1 Introduction

There have been more and more customer feedbacks on the business website, e.g., amazon, ebay and so forth. Most of them are long and redundant, or even have nothing to do with the product itself. Scanning all of these reviews would be tedious and fruitless. It would be good if these reviews could be preprocessed automatically and customers are provided with the generalized information.

Usually a customer-feedback consists of a *rating*, which is a number, and a *quote*, a paragraph of judgments.

Most current works (Pang and Lee, 2002) are based on the assumption that the rating is binary – good or bad, which is easy to process because of its polarity, but it is not always the case. For example, amazon provides five-star rating, which means that customers could have five choices for rating instead of two.

Besides the fact that we loose the polarity here, the meaning of the rating is quite vague. For example, does a three-star rating mean that the product's quality is not so good in customer's mind or does it mean that it is not so bad? Moreover, at most time, customers need an overall opinion about the product, rather than a number of particle reviews. That means later customers want to see an overall rating of the product instead of something like ten four-star ratings and ten three-star ratings.

There are even more problems with processing the quotes. In the syntactic level, many sentences in review frequently use oral English words or sentence snippets and thus could not be successfully parsed. In the semantic level, many sentences in the feedback are not related to the product itself. For example, they may describe a short interesting story about how they get to know the product, or, they may compare the current product with the others and therefore a lot of information should be related to the other product instead of the current one.

There might be some problem with the classic sentimental classification too. It generalizes the opinion of the customer to polarized ones – Excellent or Poor, but it does not find what the reviewer liked or disliked. After all, a negative sentiment on an object does not imply that the user did not like anything about the product and a positive sentiment does not imply that the user liked everything about the product.

Our approach generalizes an overall rating and user comments on several features for each product. It calculates an overall rating of the product based on PIM-IR algorithm and generalizes these comments on features using feature-based classification.

Feature Extraction turns out to be a very hard problem and that accounts for the fact that classifying and extracting quotes now are almost done by the human beings – website editors and customers. However, it turns out to be a tedious job. Thus the modern approach is combining the automation with labor force, that is, use automation to extract some potential features and let the human beings check out whether they are really useful.

The better result can be get by automation, the less work would be left for the human beings. Here we try to use the feature based sentimental classification to automatically extract features and customers' opinions toward these features out of the quotes and use PIM algorithm to try to give customers an overall rating of the product.

2 Related Works

Most previous research on sentimental classification has been at least partially knowledge-based. Some of this work focuses on classifying the semantic orientation of individual words or phrases, using linguistic heuristics or a pre-selected set of seed words (Hatzivassiloglou and McKeown, 1997; Turney and Littman, 2002). Since these methods take into account the human being subjective factors, some argue that humans may not always have the best intuition for choosing discriminating words than some statistic methods do (Pang and Lee, 2002). However, in our project, it turns out that human subjective opinions would provide more precise result than statistical method. One possible explanation to the contradiction is that the number of samples used in Pang's result is too small.

Pang, *et. al* also tried to examine whether it suffices to treat sentiment classification simply as a special case of topic-based categorization (with the two "topics" being positive sentiment and negative sentiment). Three standard algorithms: Naive Bayes classification, maximum entropy classification, and support vector machines are tested. It is reported that the results produced via machine learning techniques are quite good in comparison to the human-generated baselines. In terms of relative performance, Naive Bayes tends to do the worst and SVMs tend to do the best, although the differences aren't very large. However, their approach is based on the assumption that users' opinions are polarized.

3 System Overview

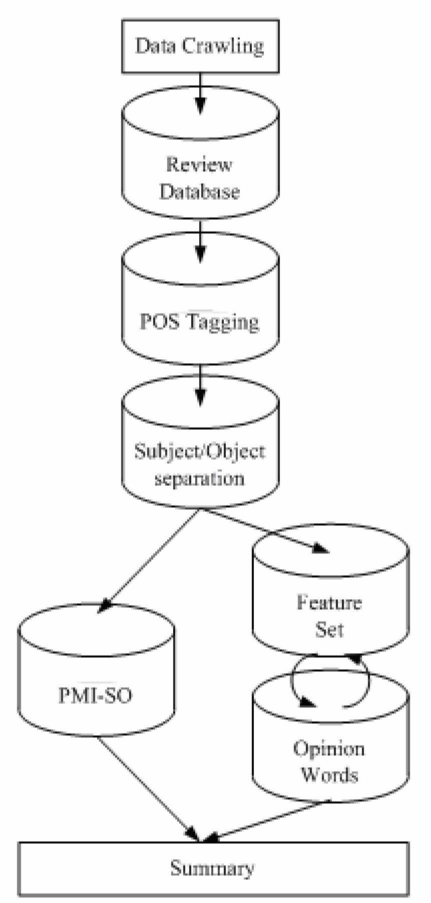
Our program consists of seven parts, as the following figure shows.

First, we extract the customer feedbacks from the websites and add them to the review database; second, we use POS Tagging to tag these reviews; third, by employing the subject/object review separation described in (Yeh, 2006), we eliminate the objective descriptions, which are not related to the opinion of the customers, from the reviews.

Then, on one hand, PMI-IR (Turney, 2002) algorithm is used to calculate the mutual information between the review and the polarized words to generate the weight of the ratings. The overall rating is the weighted average of each single rating coming along with the product review.

On the other hand, we use the idea based on (Hu and Liu, 2005) to extract the features out from the customer review and find the opinion words toward each of these features. We prune the feature set by the relationship between the features and opinion words.

Finally, for each product, the program returns an overall rating and several pairs of <feature, opinion words> as the summary of the reviews.

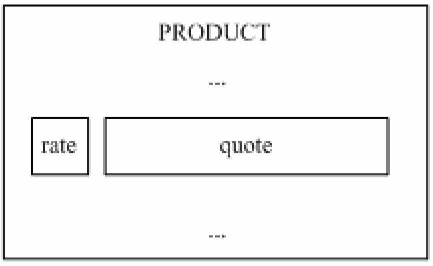


4 Data Collection

We extract data from Buy.com (http://www.buy.com). The first approach we used to extract data was using JSpider, an open source Java implementation of a flexible and extensible web spider engine. However, JSpider is not so stable nowadays. Therefore, we wrote a short program taking advantage of the wget function in Linux. As the first step, we download all the web pages in Electronic category in Buy.com as text html files. Then we extract the useful information – the ratings and the comments by a Java program based on HtmlParser, a Java library used to parse HTML in either a linear or nested fashion. HtmlParser is primarily used for transformation or extraction, it features filters, visitors, custom tags and easy to use JavaBeans. It turns out to be a fast, robust and well tested package. After extraction, we have approximately 1000 ratings and reviews for 92 products.

When more data is needed, we would refer to other websites. Since the document format might be completely different among websites, we might have to use some alternative approaches. One of them is using Document Object Model. Another approach is using wrappers, which is described in (Irmak and Suel, 2006). Currently the data from Buy.com is sufficient and all the page formats there are consistent.

Finally, the data format is described in the following figure. For each product, we have several items. Each item is a combination of a *rating* and a *paragraph review* from a single user. Moreover, we also have the name of the product



5 Subjective/Objective Partition

A review can be broken down into subjective sentences that are expressive of the reviewer's sentiment about the product, and objective sentences that do not have any direct or obvious bearing on or support of that sentiment. For example, the following review could be divided into subjective part, which is in the underscored format, and objective part, which is in the italic format:

"

I purchased this TV locally to upgrade a sunroom tv to hi-def. Now it is sitting side by side with my two year old Sony Bravia and a recently purchased Samsung. <u>The picture was not as bright, not as sharp, and with any sports presentation, the screen was a blur. I returned it because the overall quality was unacceptable, even at the low price.</u>

"

The idea of segmenting the essay into several coherent parts comes from TextTiling, a method for partitioning full-length text documents into coherent multi-paragraph units, proposed in (Hearst, 1997). The layout of text tiles is meant to reflect the pattern of subtopics contained in an expository text. The approach uses lexical analyses to determine the extent of the tiles, incorporating thesauri information via a statistical disambiguation algorithm. The tiles have been found to correspond well to human judgments of the major subtopic boundaries of science magazine articles.

We employed the idea of (Yeh, 2006) to separate the reviews into subjective parts and objective parts. This idea of cohesiveness was used to indicate segments of a review that are more subjective in nature versus those that are more objective. Yeh showed that reviews usually have a smooth continuance of subjective and objective sentences, instead a haphazard and disjoint mixture, as we can see from the above example. Yeh proposed the following ratio,

$$\frac{Subjectivity}{Objectivity}$$

and used a corpus of subjective and objective sentences for training, obtained from http://www.cs.cornell.edu/people/pabo/movie-review-data/ (Pang and Lee, 2004). We use a large tagged training set to get the probability for each word to be subjective or objective, and the probability of a sentence to be subjective or objective is calculated using the unigram model.

6 POS Tagging

Based on the idea of (Turney, 2002), a part-of-speech tagging is applied to the review.

The parts-of-speech we are interested in are nouns and adjectives, since they correspond to feature/attribute pairs that we want to extract. However, we could not extract the words independently because, consider tagging the sentence like "*This product is not too bad*", where customer's opinion is negated by the adverb "not". Therefore, we employ the idea proposed in (Turney, 2002), where two consecutive words are extracted from the review if their tags conform to any of the patterns in Table 1.

In Table 1, *JJ* tags indicate *adjectives*, the *NN* tags are *nouns*, the *RB* tags are *adverbs*, and the *VB* tags are *verbs*. The first pattern means that two consecutive words are extracted if the first word is adjective and the second is noun; the second pattern means that two consecutive words are extracted if the first word is adverb and the second word is adjective, but the third word - which is not extracted - cannot be a noun; the third pattern means that two consecutive words are extracted if they are all adjectives but the following word is not noun. Singular and plural proper nouns are avoided, so that the names of the items in the review cannot influence the classification.

Word 1	Word 2	Word 3
JJ	NN/ NNS	anything
RB/ RBR/ RBS	JJ	Not NN or NNS
JJ	JJ	Not NN or NNS
NN/ NNS	JJ	Not NN or NNS
RB/ RBR/ RBS	VB/ VBN/ VBD/ VBG	anything

Table 1

7 PMI-IR Algorithm

PMI algorithm uses mutual information as a measure of the strength of semantic association between two words (Church and Hanks, 1989). The Pointwise Mutual Information (PMI) between two words, word1 and word2, is defined as follows

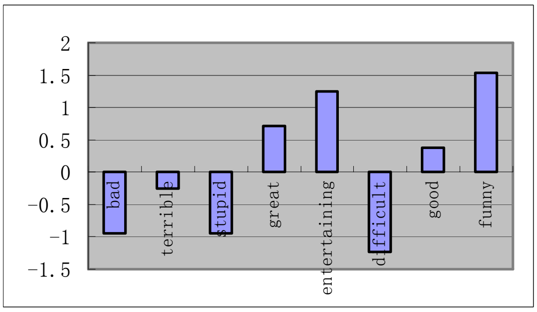
$$PMI(word_1, word_2) = \log\left(\frac{P(word_1 \& word_2)}{P(word_1)P(word_2)}\right)$$

Where *P(word1 & word2)* is the probability that word1 and word2 occur at the same time. If the words are independent of each other, their mutual information value is zero. The ratio is thus a measure of the degree of dependence between the words. The log of this ratio is the amount of information that we acquire about the presence of one of the words when we observe the other. As proposed in (Turney, 2002), the *Semantic Orientation* of a phrase is calculated as follows,

$$PMI(phrase, "excellent") - PMI(phrase, "poor")$$

The reference words “excellent” and “poor” were chosen because, in the five star review rating system, it is common to define one star as “poor” and five stars as “excellent”. *Semantic Orientation* is positive when phrase is more strongly associated with “excellent” and negative when phrase is more strongly associated with “poor”.

Turney calculates the PMI for each phrase using the search engine and replaces the probability of the word with the number of hits of that word in the search engine. We chose to train our PMI on 2000 movie reviews that were available to us. The following figure shows some of the result:



Positive adjectives generally show a positive SO value, while negative adjectives generally show a negative SO value.

Then we calculate the average of the *Semantic Orientation* of the phrases in each review. We don’t use the average *Semantic Orientation* directly as the judge of the product but use these *SO* to be the weight of the ratings provided by customers. Thus the final overall rating of the product is given by

$$\sum_R rating(R) * SO(R),$$

where *R* is a paragraph of review and *rating(R)* is the rating provided by the customer coming along with the review. It turns out that the overall rating score calculate by this formula is more objective.

8 Feature Extraction

Sentiment classification is useful but it does not find what the reviewer liked or disliked. After all, a negative sentiment on an object does not imply that the user did not like anything about the product and a positive sentiment does not imply that the user liked everything about the product. The solution is to go to sentence and feature level.

Our aim is to find what people like and dislike about a given product. Therefore how to find out the product features that people talk about is an important step. However, it turns out to be a very hard problem. For example, in the sentence
"*this MP3 could not easily fit in pockets*", actually the customer is talking about the size but the word *size* is not explicitly mentioned in the sentence. However, implicit features occur much less frequent than explicit features. Thus we focus on finding features that appear explicitly as nouns or noun phrases in the reviews.

We have preprocessed the reviews by eliminating the objective part as mentioned in part 4 and POS tagged them as mentioned in part 5. To find features that people are most interested in, Hu and Liu proposed a method based on the association rule mining (Agrawal and Srikant 1994) to find all frequent item sets, where an item set is a set of words or a phrase that occurs together. The kernel of the method is using the association rule miner, CBA (Liu, Hsu and Ma 1998), which is based on the Apriori algorithm in (Agrawal and Srikant 1994). It finds all frequent item sets and they treat each resulting frequent item set as a possible feature.

We did not use such a complicated method to find the frequent features. Instead, we calculate the frequency of each noun in the review database and consider it as a useful feature if its probability is above some threshold. Later, we would refine the feature set by considering the opinion words that are associated with the feature as mentioned below.

9 Opinion Words Extraction

People use opinion words to express a positive or negative opinion such as "good", "bad", and so forth. Opinion words and product features are not independent of each other - the opinion words always locate around the feature in the sentence. For example, let us look at the following sentence:
"
The power is so limited that you have to always change the battery.
"
In this sentence, *power,* the feature, is near the opinion word *limited*.

Based on this observation, we can extract opinion words in the following way. For each sentence in the review database, if it contains any frequent feature, extract the nearby adjective. If such an adjective is found, it is considered an opinion word. A nearby adjective refers to the adjacent adjective that modifies the noun/noun phrase that is a frequent feature. In this way, we build up an opinion word list.

If we cannot find an opinion word for a feature that we get in the previous step, we consider such feature is not really a frequent feature and thus eliminate it from the feature list. Thus through the relationship between the feature and opinion word, we refine the feature list. Moreover, what if we find many opinion words but cannot find any feature word? We employ the following method. For each sentence in the review database, if it contains no frequent feature but one or more opinion words, find the nearest noun/noun phrase of the opinion word. The noun/noun phrase is then stored in the feature set as a potential feature.

10 Experiments

We tested our models on the 1000 reviews that we extracted from buy.com. Most of the reviews are on products such as mp3 players, laptops, and wireless adaptors.

10.1 Experiment I

For the first experiment, we trained our model on the entire wall-street journals database (wsj 200- 2499). Using our model, we attempted to extract feature/attribute pairs from reviews using the heuristic that most of these pairs are noun/adjective pairs in the same phrase. After we processed all the reviews for a particular product, we obtained an attribute list for each feature that was mentioned by any of the reviews:

price -> *[excellent : 2.0, low : 1.0]*
performance -> *[poor : 2.0, middling : 1.0]*
product -> *[good : 6.0, great : 5.0, awesome]*
unit -> *[little : 2.0, handy : 2.0, hard : 1.0]*

We would then use a stop list to filter out very generic terms such product, unit, value, etc… We also construct pros and cons lists with adjectives that have very clear polarity connotations. It is also interesting to flip the lists and see what features have each of the different attributes:

excellent -> *[strength : 4.0, security : 2.0]*
difficult -> *[connection : 1.0, setup : 1.0]*

Finally, we put all of these together and come up with a pros and cons list for each product based on the reviews. The more attributes we find in the pros list, the higher that feature will score. Conversely, the more attributes we find in the cons list, the lower that feature will score.

PRODUCT FILE NAME : D-Link DWL-G120
There are 40 review(s) total.
PROS : *setup (5.0)*
connection (4.0)
price (4.0)
card (2.0)
connections (2.0)
CONS: *performance (-5.0)*
port (-2.0)
sensitivity (-2.0)

10.2 Experiment II

For the second experiment, we wanted to test if differentiating the objective sentences from the subjective sentences and only using the subjective sentences to score will increase our performance.

To train, we used the data from Cornell that we mentioned at the end of section 5. There are 5000 objective sentences and 5000 objective sentences. We used a smoothed unigram model to obtain the maximum likelihood of each word being in a subjective/objective sentence.

OBJECTIVE : (-2.2) usually we get a 54 mbps signal depending on where her antenna is located
SUBJECTIVE : (4.157) best product at a low price

The number in parenthesis is the log value of the ratio of likelihoods of being subjective vs. being objective. A positive number indicates a subjective sentence and a negative number indicates an objective sentence.

The results for this experiment were almost identical to the previous one. We further investigated the causes for this lack of improvements. We concluded that this is due to the fact that only 15% of the review sentences are objective and most of those objective sentences do not have the desired noun/adjective pairs that we are looking for.

10.3 Experiment III

In this final experiment, we seek to find out whether automatically generated pros and cons lists will improve performance.

The Point-wise Mutual Information (PMI) method we used for this task was introduced in section 7. We automatically classify adjectives into positive and negative categories based on the frequency of their co-occurrence with “excellent” and “poor”. Words that occur more often with “excellent” are automatically added to the positive adjective list. Words that occur more often with “poor” are added to the negative adjective list. This is what the lists look like:

[excellent : 182.0, good : 10.0, great : 7.0, entertaining : 6.0, etc....]
[poor : 212.0, bad : 15.0 , little : 13.0, rich : 10.0, many : 9.0, other : 8.0, etc....]

Using the Semantic Orientation value of each adjective in the attribute list, we can now distinguish a strongly positive word from a weakly positive word. This enable us to more precisely rate and rank each feature of the product. The result seems to be more complete and reasonable.

PRODUCT FILE NAME : D-Link DWL-G120
There are 40 review(s) total.
PROS : price (6.209) settings (1.251) adapter(0.845)
CONS : performance (-4.20) card (-2.02) driver (-1.334) reception (-1.1)

As an evaluation criteria and a comparison among different approaches, we calculated accuracies based on the number of features in the pros and cons list that we think are reasonable. The following table shows this result:

Model	**Accuracy**
Base	57.1%
Subjective/Objective	60%
Auto Pro/Con Gen.	67.5%

11 Discussion

Our experiments have demonstrated that our model provides a viable way to extract features from product review. Given a sufficiently large number of existing reviews for a product, we are able to inform future buyers of this product what aspects of the product are good and bad according to the previous buyers. This is indeed very valuable information.

The idea of automatically classifying words into positive and negative categories is also very handy for the task we are trying to accomplish. If we have a large enough text database on electronic products reviews, we should be able to rate every adjective very accurately and precisely. This will greatly facilitate the generation and ranking of the pros and cons lists for products.

Reference

Turney, Peter D. 2002. *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised, Classification of Reviews.* Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), Philadelphia, Pennsylvania, USA, July 8-10, 2002. pp 417-424. NRC 44946.

Hatzivassiloglou, V., & McKeown, K.R. 1997. *Predicting the semantic orientation of adjectives*. Proceedings of the 35th Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL(pp. 174-181). New Brunswick, NJ: ACL.

Hearst, M.A. 1992. *Direction-based text interpretation as an information access refinement.* In P. Jacobs (Ed.), Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval. Mahwah, NJ: Lawrence Erlbaum Associates.

Eric Yeh, 2006. *CS224N/Ling237 Final Project Picking the Fresh from the Rotten: Quote and Sentiment Extraction from Rotten Tomatoes Movie Reviews.*

Minqing Hu and Bing Liu. *Mining Opinion Features in Customer Reviews.* Proceedings of Nineteeth National Conference on Artificial Intellgience (AAAI-2004), San Jose, USA, July 2004.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan, *Thumbs up? Sentiment Classification using Machine Learning Techniques*, Proceedings of EMNLP 2002.

Bo Pang and Lillian Lee, *A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts*, Proceedings of ACL 2004.

Bo Pang and Lillian Lee, *Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales*, Proceedings of ACL 2005.

Utku Irmak, Torsten Suel, 2006, *Interactive Wrapper Generation with Minimal User Effort*

Nitin Jindal and Bing Liu. *Mining Comprative Sentences and Relations.* Proceedings of 21st National Conference on Artificial Intellgience (AAAI-2006), July 16.20, 2006, Boston, Massachusetts, USA.

Bing Liu, Minqing Hu and Junsheng Cheng. *Opinion Observer: Analyzing and Comparing Opinions on the Web* Proceedings of the 14th international World Wide Web conference (WWW-2005), May 10-14, 2005, in Chiba, Japan.

Minqing Hu and Bing Liu. *Mining and summarizing customer reviews.* Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD-2004, full paper), Seattle, Washington, USA, Aug 22-25, 2004

Manning, C. and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*, MIT Press. May 1999.