Feature Extraction from Micro-blogs for Comparison of Products and Services

Peng Zhao, Xue Li, and Ke Wang

Nanjing University, 210093 China,
The University of Queensland, QLD 4072 Australia,
Simon Fraser University, Vancouver, Canada
zp10@software.nju.edu.cn,
xueli@itee.uq.edu.au,
wangk@cs.sfu.ca

Abstract. Social networks are a popular place for people to express their opinions about products and services. One question would be that for two similar products (e.g., two different brands of mobile phones), can we make them comparable to each other? In this paper, we show our system namely *OpinionAnalyzer*, a novel social network analyser designed to collect opinions from Twitter micro-blogs about two given similar products for an effective comparison between them. The system outcome is a structure of features for the given products that people have expressed opinions about. Then the corresponding sentiment analysis on those features is performed. Our system can be used to understand user's preference to a certain product and show the reasons why users prefer this product. The experiments are evaluated based on accuracy, precision/recall, and F-score. Our experimental results show that the system is effective and efficient.

Keywords: social network, feature extraction, opinion mining, sentiment analysis

1 Introduction

In recent years, social networks are becoming a popular place for people to express their opinions about social-economical issues or on products and services. Twitter, as a representative of social networks, is a microblogging service [16]. A tweet is a post or status update on Twitter. There has been a significant increase of daily posted tweets in recent years: 50 million in 2010, 200 million in 2011, and 400 million in 2012.

The direct implication of this trend is that it is difficult for consumers to compare two similar products or services that offer similar functions but with different properties. Many studies have been done on analyzing consumer reviews from forums [6] [5] [10] [13]. Bing Liu *et al* [9] explored opinion features and they analyzed certain types of customers' reviews, but their paper didn't go further to conduct sentiment analysis for the features of the products and services. Popescu and Etzioni in [13] conducted the product feature extraction and the sentiment analysis. However, their approach is based on the known consumer reports that are written by experts and this kind of second-hand information can be biased or misleading. They did not use the features discussed by the first-hand consumers in social networks. Both [6] and [13] worked on consumer reviews, their work would provide only qualitative assessment rather than quantitative analysis on consumer opinions toward those product features. To the best of our knowledge, our work is the first of this kind that directly compare and contrast two similar products or services based on the social network analysis.

In this paper, we approach the feature extraction problem based on the text in tweets for the given products or services. There are three questions to be answered, 1) what features can we extract from tweets about the given products? 2) what are people's opinions about the products based on the features extracted? and 3) how do we make recommendations based on the sentiment analysis of those extracted features? For the first question, the products are made up with some features that people would like to comment, so the first question turns out to be a feature extraction problem. For the second question, people would like, love, or even hate the products because of some features of the products. People's emotion regarding the features of products can be divided into three kinds: *positive, negative*, or *neutral*. Thus the second question becomes a sentiment analysis problem. As for the third question, recommendations can be made if the overall statistical information can be made available for all of those features identified over the given products. So we conduct comparisons on the given products not only by offering recommendations but also by explaining why people like them, based on the sentiment analysis on the extracted features.

It should be pointed out that the physical features of a product can be easily obtained from product specifications where a product is advertised. Here, we do not want to get product features in this way because there are many features that consumers do not care and there are features that might not be listed in product specifications, for which consumers may like to comment. Therefore for a particular product, features should be chosen by consumers in anyway they prefer. We call the user-preferred features as *hot features*.

There are many feature extraction algorithms such as those given in Weka [17] like Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [7], and Latent Dirichlet Allocation (LDA) [1]. An insightful review on feature extraction from text can be found in Hu and Liu [6] and an overview on effective feature extraction approaches can be found in Guyon *et al* [4]. Sentiment analysis is an approach used to categorize the overall attitude of a sentence or a paragraph towards a certain subject [13] [20].

In this paper, a new algorithm based on the *formal concept analysis* [3] is proposed to solve the feature extraction problem in our *OpinionAnalyzer* system. The difference between our feature extraction algorithm and the other feature extraction algorithms is that the feature space of a product or service is often hierarchically structured and we need an approach that will extract features in a lattice structure with a partial order, so to make features comparable to each other. As far as we know, this is the first work that extracts features and analyzes sentiments from social networks in this way. Another challenging issue is on processing social network data that is noisy and sparse in terms of the variety of words used by tweets (a tweet has about 28 words on average). Our system has shown a good performance on those data. A multi-node tree is constructed with the proposed formal concept analysis (FCA) algorithm. Consumers' opinions are analyzed through several algorithms which have been evaluated with precision/recall, as well as F-score. The evaluation shows that *OpinionAnalyzer* is effective and efficient.

The rest of the paper is organized as follows: Section 2 shows the related work. Section 3 discusses the approaches including feature extraction, sentiment analysis, and product recommendation. Section 4 discusses the results of our experiments. Section 5 presents conclusions of our work.

2 Related Work

Our approach is mostly relevant to Hu and Liu [6] and Popescu and Etzioni [13]. In [6], they use Part-of-Speech (POS) tagging to collect nouns or noun phrases since features are nouns mostly. They produced POS tags for each word (whether the word is a noun or a verb etc). After that, association rule mining is applied to filter out the frequent feature itemsets. The result of their research shows a good performance in analysing electronic products like DVD player, MP3 player, digital camera and cellular phone. Obviously, our research is related but different from theirs in many ways. POS tagging and association rules mainly focused on noun features which may skip some words of their inputs that can imply features. For instance, there are some smart phones that people prefer white ones rather than other colors. In such condition, people may talk about their preference about "white phone" when they refer to appearance. But "white" is adjective in those sentences. Which means it would be filtered off when they try to sum up all the features. Our system based on the feature extraction does not have this problem. We did not remove words by part of speech. Instead, we comprehensively analyze input words from both frequency and relationship between different words. Moreover, they use comments on products from e-commerce Web sites as input. While we use data from social networks that have a large number of short text with sparse words, which makes association rules not applicable. They demonstrated their algorithm with a small data set (500 records), while we tested our algorithm with more than 8,200 records. Our work is also different from the feature extraction method in [13], that they perform mining of consumer reviews and sentiment classification without comparing the pair of user-specified products based on the corresponding product features.

Based on the above reasons this paper will provide the effectiveness and efficiency studies for our pioneer work.

2.1 Feature extraction methods

When dealing with a large volume of text, we should scan through the text only once and generate a list of features or properties that can best represent the content of text. In doing so, we consider to extract the meaningful keywords and calculate their TFIDF [14] to represent text as feature vectors for the computational purpose. Feature extraction is an important step in text processing to transform input text into feature vectors. Guyon *et al* [4] provided a comprehensive review on feature extraction from text data and relevant applications. Insightful discussions can also be found in [6] and [13]. Weka has provided open source tools for feature extractions [17].

The task of feature extraction in this paper is to transform text data into a feature space that can best describe the interests of social network users who comment on the products or services. In brief, our feature extraction is to extract only *product features* [6] [12] that have appeared in the social networks. In the feature extraction process we need to firstly search for the relevant text from tweets where the given products are mentioned, then we apply the feature extraction algorithms on the text to derive the features for those specific products.

In order to make products comparable to each other, the output product features need to be constructed as a tree structure which can be transformed from a concept lattice where some features are general and some features are specific. This requirement especially matches with the idea of discovering concept hierarchy by formal concept analysis (FCA) approaches [3].

2.2 Formal concept analysis

The classical Formal Concept Analysis (FCA) [3] builds up a concept hierarchy by comparing the subset relationships amongst the associated terms of a concept. In FCA a concept can be associated with a single term or a set of terms. A term is a regarded as a meaningful word not appearing in the stop word list. When a term is used in describing a concept it is considered as an attribute of that concept. All the attributes that are associated with all concepts can be organized in a two dimensional matrix: one dimension (columns) is to list all attributes and the other (rows) is to list all the concepts. Then FCA algorithm will check the columns that corresponding to the matrix and form a lattice from that. It has been proven that there is a one-to-one mapping between each matrix and its corresponding lattice [3]. It can be seen that the critical step in FCA algorithm is to generate the attribute matrix for every concept by scanning the text only once.

3 Our Approach

As an example, given a smart phone product, one might be curious about how well it is received by customers. Does it have a *good* battery life? Or does it have a *good* camera? Does it look pretty? In order to answer those questions, we might need to know people's opinions towards these features. But what features would people care about mostly? How do people like it? To give answers, firstly, we need to know what features people talk mostly about. Secondly, we need to know whether people like these features or not, or in other words, the opinions of people towards the product features.

The proposed system has three main steps: (1) feature extraction, (2) sentiment analysis, and (3) recommendations. The system is initially given a set of pre-defined keywords (e.g., brands of smart phones) to search at Twitter Web site [16]. The output of the system is the counting of people's opinions towards the extracted features of the given products and the recommendations we offer.

Given a pair of products, the system firstly crawls the tweets related to the given keywords and processes them before storing them in data set. Then the feature extraction function, which is the main contribution of this paper, will extract the features that people talk about frequently. The sentiment analysis is then applied to analyze people's opinions towards the given products and classifies the opinions into three categories: *positive, negative*, and *neutral*, with respect to each feature. The recommendation process will provide conclusive comments on the products as the product that has overall positive feedback on its relevant features. It can be seen that the recommendation is evidence-based and is not only qualitative but also quantitative. In following subsections, we discuss these steps in turn.

3.1 Feature extraction

Before the product features are identified, we firstly pre-process the tweets that we crawled. Since people express opinions casually within social networks, there may have either explicit and complete sentences [5], which we can easily know what they mean; or there may have implicit sentences that are incomplete sentences or just some phrases. For example, an implicit sentence in following is difficult for identifying its feature: "This mobile phone works for a long time". In this case, it is difficult to tell whether this sentence is referring the battery life or not. Such sentences would have several different ways to express the same meaning which makes it even more difficult to find the patterns of features. Fortunately, we observed that those implicit sentences do not appear much in our data set (with less than 15% of the sentences). So we can focus on explicit statements in this paper and leave the process of implicit sentences to the future work.

Our algorithm (i.e., Algorithm 1) can filter those words that are popular but not regarded as product features. It analyzes the processed tweets and finds out the hierarchy of the high TFIDF words.

In processing tweets, the stop words, *url, user name, date*, and non-English characters are removed. The smart phone brand names are replaced by the code names for the privacy preservation. Different words were changed to their original form with the help of *lucene snowball* [11].

Suppose there are two random words in tweets: w_1, w_2. The tweets set that contains all the appearance of word w_1 is namely set c_1. Similarly, the tweets set that contains all the appearance of word w_2 is namely set c_2. If set c_1 is a superset of set c_2, then more likely, w_2 is a sub-concept of w_1. A tree structure is used to express the hierarchy like w_1, w_2 instead of a lattice as it can be seen from Figure 1.

Algorithm 1 Feature Extraction

Input:
Tw: Crawled and pre-processed tweets
Output:
T: Concept hierarchy of product features
Description:

```
begin
  Count tf of each word(wi) in Tw
  for(each word wi)  (filter infrequent words appear in Tw)
     If tf>0.01 (set this by experiments, words' tf lower than 0.01 are not meaningful)
        add wi to word set W
  for(wi1 in W) (the double loop analyzes connections between any two words)
     for(wi2 in W)
        if(wi2 != wi1) {
           tweets collection c1, every element of it contains wi1
           tweets collection c2, every element of it contains wi2
           if (c1 \(\supset\) c2)
              wi2 is a sub-concept of wi1
              T add (wi1,wi2)
              for(any win that wi2 is a sub-concept of win)
                  if (wi1 \(\supset\) win)
                     win is a sub-concept of wi1
                     wi2 is a sub-concept of win
                  else if (win \(\supset\) wi1)
                        wi1 is a sub-concept of win
                        wi2 is a sub-concept of wi1
        }
  return T
end
```

3.2 Sentiment analysis

This step is to explore people's opinions about the *hot* product features (features extracted in Section 3.1). People's opinions about products can be divided into three types, *positive, negative* and *neutral*. People use certain conventional words to express their feelings. Here are two examples:

"I am so glad that I have chosen this new cell phone with such a great camera."

"I love this white case."

Both sentences express positive feelings through words. One sentence uses the adjective "glad" to express the feeling toward a smart phone camera. The other uses a sentiment word "love" to express the feeling. All these words are essential words that can show consumer's sentiments. But there are also some words that people may use in social networks but have no contribution to analysis like "am". So we delete such kind of words same as that was done in [21] (However, we did some changes by removing sentimental verbs from the list). Then, we narrow down these input words again by using WordNet [18] to eliminate the words that are seldom used. We also delete the none-existing words. By tagging the existing words, a bitmap is established (listing all those existing words, tagging the existing words appeared in a tweet with value 1 the others with 0). Also tagging the orientation of each sentence is based on the sentimental orientation like *positive, negative*, or *neutral*. We show the result in the evaluation section. Besides, people's emotion can be divided into more types. WordNet [18] has divided some sentiment words into six types including *disgust, anger, fear, sad, surprise* and *joy*. Each of these six types shows different levels of emotions which may make the analysis more sophisticated. We calculated the precision and recall of those experiments as shown in Table 1.

The taxonomy of product features provides an overview of *hot features* as well as the results of sentiment analysis of those features (Figure 1).

3.3 Recommendation

This step is to produce recommendations based on both qualitative and quantitative analysis on people's preferred features. Products that have strong similarities, like both smart phones have highly satisfied cameras that can take high quality pictures and both have good performances in battery lives. If one customer likes one of these products, probably he would like the other one as well. We use a simple tree-similarity comparison algorithm based on the results of the sentiment analysis of the *hot features*. For the limitation of the space, the recommendation algorithm is not given in this paper.

Based on our random data collection from Twitter with 8,200 tweets, we have an example summary of brands namely *htc* and *Samsung*. Both of them received good responses from customers about their applications and systems. We only used 100 of each to plot the pictures in Figure 1.

In this given set of tweets, people talked about *Samsung* appearance and seem to like it more than *htc*. While there are more people thinking *htc* is easier to use while others think *Samsung* is a little bit complex to use. Price of *htc* seems to be a problem to this band. While unlock and some fancy things about *Samsung* are both liked and disliked by customers as it can be seen from the plotted dots in Figure 1.

Apparently the above opinions are summarized based on a small data set that has only 8,200 tweets and are obviously biased and by no means representative or comprehensive in judging products. However, this does not prevent us to show the potential of our method for its application on a large scale when a large volume of tweets becomes available. In that case, the both quantitative and qualitative analysis on two similar and competitive products can be conducted significantly and precisely over social networks by using our approach.

4 Experiments and Evaluation

Our system is called *OpinionAnalyzer* and is implemented in Java. We gathered about 8,200 tweets from [16] as the original data. The time complexity of the system is in linear growth. We gathered 103 tweets in 54 seconds and for each 1,000 in ten minutes (despite the waiting time due to Twitter's limited rate which means we can only have 100 or so tweets in an hour).

We conducted the feature extraction experiments with data related to three brands of cell phones from Twitter. Original tweets include information such as *url, twitter id, post time*. We deleted *url* and pre-processed crawled tweets before storing them in our data set. After that, we tagged the data for training purpose

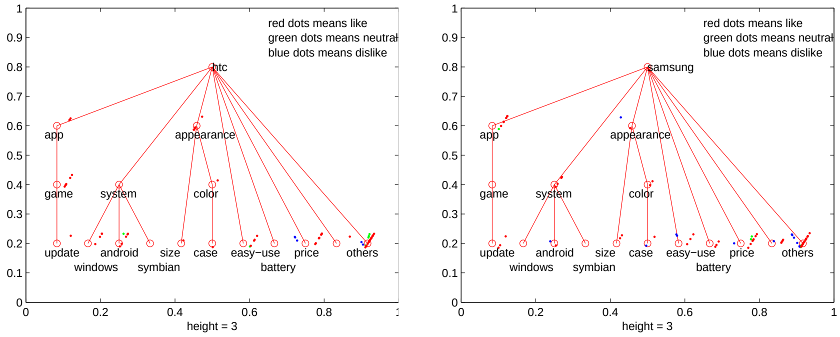

Fig. 1. Hot Features and their Sentiment Analysis for htc and Samsung

with different numbers for different classifiers and received the sentiment analysis results as *Case 1* in Table 1.

There are some words that are not frequently used in general. Reducing such kind of words would not affect the judgement of sentiment of sentences. We identified those infrequent words with the help of WordNet [18]. "Reducing infrequent words" improvements in our system are shown in *Case 2* in Table 1 as well. Compared to data for Case 1, about 10% words were removed in *Case 2*.

Also, different kinds of words might show different levels of sentiment. Mostly, "love" would show a higher preference than "like", though both of them mean "preferable". Similarly, many words would show different levels of sentiment. A "sentiment corpus" is a corpus that provides with levels of words related to emotions. In the third case, we show the results of experiment including "sentiment corpus" as *Case 3* in Table 1.

The accuracy of our system in feature extraction is above 71% which far prevails other classification algorithms, like BayesNet (33.01%), Random Forest (43.69%) and J48 (50.49%). Primarily, removing noisy data helps us filter off words not related to our topic. In addition, the feature taxonomy we derived from our algorithm helps us locate the feature. For instance, if one tweet is talking about "android" while another one is talking about "symbian". Since both of them are child-nodes of "system" in our taxonomy, we would know both of them are discussing about one thing: "system". But for any other feature extraction algorithms, the result of that would be three different unrelated features "android", "symbian", and "system". In our approach, words in the tweet show that the tweet might be talking about "android", but it mentioned "system" instead, the tagging would applied to "system" in this case. In fact, all those tweets were talking about one generic concept: "system".

Experimental results are evaluated in terms of precision (p), recall (r), and $F1$ score. In Table 1, all results are the average numbers of cross-validation experiments. With the increasing size of our data set, from 1,000 tweets to 8,000 tweets, increased 1,000 tweets every time, $F1$ score is stabilized at 0.75 for all algorithms. Among all applied classifiers, with all of our experiments with different sizes of data sets, Random Forest classifier has the best performance at all time. "Reducing infrequent words" as *Case 2* improves the performance a little in every classifier. But the sentiment corpus has no improvements as shown in *Case 3* for two classifiers. The reason might be that in social networks, those sentiment words are not used so often and straight. People in social networks have invented many more colloquial words or phrases to express their feelings.

As a result of these experiments, we would be able to recommend products to people with the reasoning based on the sentiment analysis of those specific product features. We had observed that Twitter often posted the news about a product several weeks in advance before actual official announcement of the product. For instance, we found out that the new release of *htc butterfly* had gained a good reputation among customers two weeks before the product official announcement released. This might indicate a new research issue on the authentication of the sentiment analysis.

Table 1. Evaluation of Sentiment Analysis with Different Data Pre-processing

	BayesNet			J48			Random Forest		
	p	r	f1	p	r	f1	p	r	f1
Case 1	0.741	0.752	0.745	0.675	0.676	0.638	0.798	0.8	0.77
Case 2	0.75	0.757	0.752	0.701	0.724	0.667	0.829	0.805	0.768
Case 3	0.75	0.757	0.752	0.701	0.724	0.667	0.789	0.781	0.733

5 Conclusions

When people comment on products or services online, they may choose any features they prefer. Then their feelings towards those features are expressed. For different products to be compared to each other, we need to extract those commonly discussed features to make them comparable. Those product features can subsume each other in a hierarchical structure. This paper has proposed a new algorithm based on the formal concept analysis (FCA) to have derived a taxonomy of *hot features* of the products. Then sentiment analysis is performed on those features. Based on the analysis of those features, an overview on how people like or dislike these products can be presented.

In future research, we will improve the effectiveness and scalability of our method for mining social opinions on a wide range of products and services.

References

1. Blei, D.M., Andrew Y.Ng, and Jordan, M.I.: Latent dirichlet allocation. Journal of Machchine Learning Research, Vol. 3 , 993–1022 (2003)
2. Breiman, L.:Random forests.: Machine learning, 45(1): 5–32 (2001)
3. Ganter, B., Wille,R.: Applied lattice theory: Formal concept analysis, In General Lattice Theory, G. Grätzer editor, Birkhä user (1997)

4. Guyon, I., et al (ed) : Feature Extraction: Foundations and Applications, Springer,(2006)
5. Hu, M., Liu, B.: Mining and Summarizing Customer Reviews. In: Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, pages 168–177. ACM Press New York, NY, USA (2004)
6. Hu, M., Liu, B.: Mining opinion features in customer reviews, Proceedings of the National Conference on Artificial Intelligence. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; pp. 755–760 (2004)
7. Lachenbruch, Peter A., and M. Goldstein.: Discriminant analysis. Biometrics (1979): 69–85.
8. Linden,G.,Smith, B., York, J.: Amazon.com recommendations: Item-to-item collaborative filtering. Internet Computing, IEEE, 7(1): 76–80 (2003)
9. Liu, B., Hu, M., and Cheng, J.: Opinion observer: Analyzing and comparing opinions on the web. In: Proceedings of the 14th international conference on World Wide Web, pp.342–351. ACM(2005)
10. Liu, B.: Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies,5(1),1-167. Morgan Claypool Publishers(2012).
11. Lucene snowball, http://lucene.apache.org/core/old_versioned_docs/versions/3_0_0/api/contrib-snowball/
12. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002, pp.79–86.
13. Popescu, A.M., Etzioni O.: Extracting Product Features and Opinions from Reviews. In Natural Language Processing and Text Mining, 2007, pp.9-28
14. Salton, G.,Buckley,C.: Term-weighting approaches in automatic text retrieval. Information processing & management, 24(5):513-523 (1988)
15. Sarwar, Badrul, et al.: Item-based collaborative filtering recommendation algorithms. Proceedings of the 10th international conference on World Wide Web. ACM, 2001.
16. twitter, https://twitter.com/
17. WEKA The University of Waikato, http://www.cs.waikato.ac.nz/ml/weka/
18. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press, http://wordnet.princeton.edu
19. NLProcessor-Text Analysis Toolkit.2000. http://www.inforgistics.com/textanalysis.html
20. Yi, J., Nasukawa, T., Bunescu, R., Niblack, W. : Sentiment Analyzer: Extracting Sentiments about a Given Topic Using Natural Language Processing Techniques. In: Procs. of ICDM. 2006 pp.1073–1083 (2003)
21. http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop