Sentiment Analysis of Customer Reviews based on Integration of Structured and Unstructured Texts

Nabiha Asghar, Rija Sana
Department of Computer Science
Lahore University of Management Sciences

1 Motivation

With the explosive growth of user-generated content on the Web, the need for its useful analysis has grown exponentially. Successful mining of peoples sentiments and opinions from this plethora of posts on discussion boards, blogs, online forums and merchant sites, can be vital to various fields such as marketing and product development. Companies selling a product no longer have to rely on their survey teams to gather data on how well the product is doing in the market. All they have to do is, collect this review data from the web and utilize it for upgrading/improving the product. The main idea behind sentiment analysis of textual sources is that there is a huge volume of opinionated text available online. These opinions are hidden in long forum posts and blogs, and there are thousands and millions of these available on the Web. Hence sentiment analysis through data mining techniques provides an automated way of opinion discovery and summarization.

2 Introduction to Opinion Mining and Sentiment Analysis

The main tasks of Sentiment Analysis or Opinion Mining are:

1. to identify the features of a given product that have been commented on by reviewers e.g. if a review is about a camera, then a reviewer might comment on its weight, battery life and picture quality. These 3 things are known as features.

2. to find out the opinions expressed about features and determine their sentiment orientation e.g. finding out whether a particular review is positive or negative. E.g. This is a great camera expresses the writers general approval of the camera. On the other hand, The battery life is too short! refers to his disapproval of a specific feature of the camera.

Over the years, various techniques have been proposed to deal with Task 1. Techniques relying on association rule mining, statistical measures, subsequence similarity measures and syntactic dependencies between opinion words and features of a product etc have been proposed, among various others.

For Task 2, the goal gets a little complicated. Over here, the opinion mining and analysis can be done either on a document level (i.e. finding out the sentiment polarity of an entire document) or on a sentence level (finding the polarity of a single sentence). Sentence level analysis has can further be divided into the subtask of looking for specific portions of the sentence, which we will call chunks here, such that even if the rest of the sentence is discarded, enough information is obtained from these chunks. For this task, approaches involving lexicon based methods, log-likelihood ratio and dissection of a sentence into clauses have been proposed. We can see that Task 2 is of prime importance as far as data mining is concerned, and therefore it is more challenging in terms of development of an efficient algorithm, followed by an efficient implementation technique.

3 Related Work

For the convenience of the reader, we give a very brief insight into the specific techniques employed by researchers for undertaking Task 2. (Task 1 is fairly basic, therefore we omit its details in this report.)

1. In 2002, Pang and Lee used three machine learning techniques (nave Bayes, maximum entropy and SVM) for sentiment classification on a document level.

2. In 2003, Yu and Hatzivassiloglou determined the sentiment of a sentence by calculating the average sentiment orientation of all the opinion words in a single sentence. This calculation was based on log-likelihood ratio.

3. In 2004, Hu and Liu, in their paper Mining and summarizing customer reviews, provided a more sophisticated method by proposing a sentence level analysis algorithm. Theirs is a lexicon-based method to use opinion words (i.e. words commonly used to express

positive or negative sentiments e.g. amazing, awesome, expensive, ridiculous etc). Opinion lexicon is a collection of opinion words and opinion phrases (e.g. it cost me an arm and a leg). This method basically counts the number of positive and negative opinion words that are near the product feature in each review sentence. The opinion lexicon is obtained through WordNet.

4. In 2010, Thet, Na and Khoo provided yet a way of fine-grained aspect-based sentiment analysis on sentence level. Instead of simply determining the overall polarity of each sentence, this method determines both the sentiment orientation and sentiment strength of the reviewer towards various aspects of a movie. The idea is that each sentence contains various clauses, each of which expresses different sentiments toward different aspects of a movie. So each sentence is divided into clauses, and then the orientation and strength of each clause is determined. Finally the results are combined to give the overall polarity of each sentence as well.

5. In 2010, Li and Zeng produced yet another way of fine-grained sentiment analysis. They investigated how to mine product features and opinions from multiple review sources and proposed an integration strategy. This fine grained approach first extracts product features and opinions from semi-structured reviews to build a domain knowledge base, and then exploits this metadata to boost up the mining process of unstructured reviews. This results in generation of feature-opinion tuples (e.g. great picture-quality or awesome battery life) which serves our goal.

The traditional document-level and sentence-level classification approaches are highly coarse to provide a deep analysis of product reviews. A more in-depth analysis of natural languages is therefore more meritorious. Therefore, of these listed techniques, we focus on the last one namely Fine Grained Sentiment Analysis, since it is the most recent of the publications made in this regard.

4 Algorithm Overview

4.1 Definitions

Semi-structured Reviews: In semi-structured reviews, Pros and Cons are listed separately by the writer and the contents of Pros and Cons are usually short phrases. Short phrases are separated by commas or periods etc. We can view each semistructured review as a sequence of segments, where a segment could be a product feature or an opinion word or a feature-opinion pair.

Example:
Pros: Awesome picture quality, beautiful camera
Cons: Useless battery, redeye issues.

Unstructured Reviews: These are written in a free format by the writer, without any constraints. There is no formal separation of Pros and Cons and the content may consist of several sentences, where each sentence contains features and/or opinions. We can see from the example below that unstructured reviews have the potential to provide more abundant and detailed opinion information than its counterpart.

Example:
I love this camera! The picture quality is simply brilliant and the camera is so handy. Why didnt I buy it earlier?

Explicit Feature: If a feature f appears in the segment/chunk of a review sentence, the feature is called an explicit feature of a product. For example, in the segment the picture is wonderful, picture is an explicit feature.

Implicit Feature: If a feature f does not appear in the segment of review, but is implied, the feature is called an implicit feature of a product. For example, in the segment it is very expensive, price is an implicit feature, and expensive is a feature indicator.

4.2 Algorithm

The idea is that unstructured reviews usually contain more sentiment content because a reader can also express why he likes or dislikes a certain feature of the product, and he can compare and contrast two or more products. Since analyzing unstructured reviews is a significantly challenging task in terms of automation as compared to semi-structured ones, the idea is to mine the semi-structured reviews of a product to generate domain knowledge, which can then be utilized to carry out the mining process on the unstructured reviews. Hence, this algorithm works on the principle of an integration of sorts of multiple review sources for a single product. We give the salient features of the original algorithm now. Specific details of the implementation of each step as well as the results will follow.

Extraction of domain knowledge from semi-structured reviews is done in 3 steps:

1. extracting product features and opinions,

2. propagating product features and opinions, and

3. associating product features and opinions.

Step (i): Since product features and opinions are stated explicitly in semi-structured reviews, we manually develop some extraction rules to extract them. For this, parsers are available online that generate part of speech tags and dependency tree for each subjective sentence. From this, high frequency nouns, noun phrases, adjectives, adverbs etc are extracted. The sentiment orientation of each feature is found by whether the user has written it in the pros list or the cons list.

Step (ii) and (iii): Now the product features and opinion words are propagated to their synonyms and antonyms to enlarge our domain knowledge base. This way, new associations are formed. E.g. from the feature-product pair good picture, we can propagate picture to its synonym image and good to excellent. So now we have new pairs excellent image and good image.

Now we move on to the mining of unstructured reviews. Here, we first extract high-frequency words and phrases that match the extraction rules and then filter out the feature candidates, which are not true product features indicated by our domain knowledge base. Each candidate is given a confidence score that shows how confident the candidate is a product feature. This confidence score is computed by finding semantic similarity of words and phrases through Omiotis. Next, for each of these features we extract the preceding and the following word in the text of unstructured reviews. We manually clean these results so that we get valid opinion-feature tuples only that occur in the text. We take the most frequent tuples out, and this is our final result.

5 Experiments

The system architecture consists of four parts (Figure 1):

1. Data Set formation and pre-processing: Semistructured and Unstructured Reviews

2. Domain Knowledge Mining Engine (DKME): Generates feature-opinion tuples and their polarity from semistructured reviews

3. Unstructured Review Mining Engine (URME): Exploits the domain knowledge to assist feature extraction, followed by opinion polarity identification

4. Feature-Opinion Tuple Generator (FOTG): Merges results of (3) with (2) to obtain the final list of feature-opinion tuples.

5.1 Dataset Formation and Preprocessing

The product that we analyze is a digital camera: Canon PowerShot SD500. Two sets of data are required for this experiment: Semi-structured reviews and Unstructured Reviews.

The semistructured reviews are obtained manually from www.cnet.com and www.epinions.com. (For convenience, we store the Pros and Cons in two separate files for easy processing. The results are merged later in the Domain Knowledge Base). Unstructured reviews are collected manually from www.amazon.com. The descriptive statistics are shown in Figure 2.

	Semistructured Reviews		Unstructured Reviews	
Product	No. of Reviews	Average length (in words)	No. of Reviews	Average length (in words)
Camera	404	7	200	18

Figure 2: Descriptive Statistics of the Data

Key extraction rules and instances.

Rules	POS Tags	Instances
N → F	NN, NNS	Picture
NN → F	NN NN	Battery life
JN → O F	JJ NN(NNS)	Poor quality
JNN → O F	JJ NN NN	Small memory card
JTB → O to F	JJ TO VB	Difficult to use

Note. F = Product feature; O = opinion word; NN = noun; NNS = noun plural; JJ = adjective; VB = verb.

Figure 4: Key Extraction Rules and Instances

5.2 Data Knowledge Mining Engine

The task for this engine is to build a domain knowledge base from the semistructured reviews that can assist us in the mining of unstructured reviews. Through observation, it is found that semistructured reviews have two properties: First, they consist mostly of subjective segments that have product features and associated opinions written explicitly. Secondly, because the opinions are written under the separate headings of Pros and Cons, it is easier to identify the polarity of opinions.

First we use an NLProcessor linguistic parser (http://www.infogistics.com/textanalysis.html) to tag the semistructured reviews text file automatically. The tags are POS (Parts of Speech) tags that indicate nouns, noun groups, verb groups, adjectives etc wherever they occur in the text. The dependency tree of for each segment is also generated. Figure 3 illustrates the result of the tagging and the corresponding dependency tree:

Next, we extract nouns, noun groups, verb groups and adjectives according to the tagging and syntactic chunks (Code Reference: ruleextraction.java). For each of the lists, we just keep the high frequency words (Figure 5). This is done with the help of RapidMiner. We manually inspect the list of nouns, noun groups and verb groups obtained to remove any nouns that do not correspond to a feature of the camera. For example, a review may contain the word mind, which will be extracted since it is a noun. However it is not a feature of the camera so we remove it, since our goal is to keep only those nouns that are possible features of the camera. We also apply some extraction rules (shown in Figure 4) to get the most relevant segments of the text.

Now we enlarge the domain knowledge base by adding nouns/adjectives that are synonyms for the words occurring in the high frequency nouns/adjectives list we just obtained. By propagating each product feature and opinion

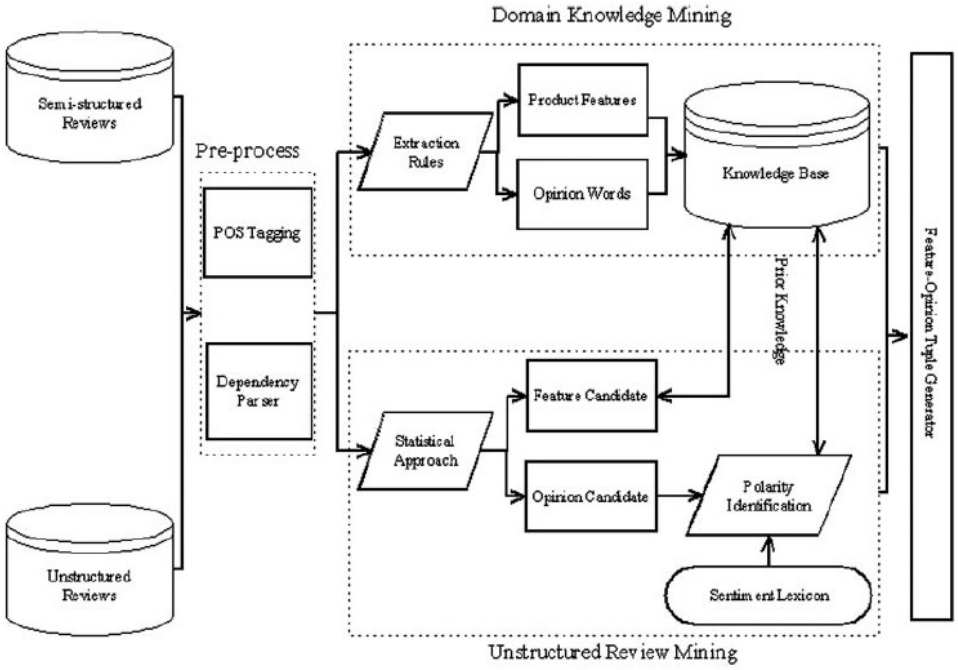

Figure 1: System Architecture

Figure 3: POS Tagging and Dependency Tree for Semi-structured Reviews. Key: S → Sentence, W → Word, NG → Noun Group, VG → Verb Group, JJ → Adjective.

```
picturequality,49
imagestabilization,16
screen,23
automode,8
buildquality,3
zoom,15
macro,6
price,12
color,10
shutterlag,8
weight,15
value,4
build,6
interface,3
durability,4
easeofuse,17
```

Figure 5: Some of the frequent nouns/features obtained through Java and RapidMiner

```
excellent zoom, positive
fair picture quality, positive
perfect design, positive
wonderful video quality, positive
low image stabilization, negative
amazing display, positive
huge size, negative
poor face detection, negative
slow auto mode, negative
blurry colors, negative
noncrisp resolution, negative
flimsy  build, negative
average design, neutral
```

Figure 6: A Portion of the Domain Knowledge Base

```
INPUT: Unstructured Reviews (uR) and Domain Knowledge (DK)
OUTPUT: Feature-Opinion Tuples
FOR each sentence s ∈ uR
do pos tagging and dependency parser
extract words and phrases match extraction rules
compute the frequency of words f(w) and phrases f(p)
IF f(w)>threshold, put w into feature candidate set (FCS)
IF f(p)>threshold, put p into feature candidate set (FCS)
FOR each fc ∈ FCS
 IF (fc is a word)
    FOR each feature f ∈ DK
      IF (fc matches f)
       merge fc to f
       DK=DK+fc
       FCS=FCS- fc
      ELSE
      compute the semanticSim (fc , f )
      merge fc to f = max (semanticSim (fc , f))
      DK=DK+ fc
      FCS=FCS- fc
 ELSE
   IF (fc is a phrase)
   FOR each feature f ∈ DK
    IF (fc matches f)
       merge fc to f
       DK=DK+fc
       FCS=FCS- fc
    ELSE
    compute the phrase Sim fc , f)
    merge fc to f = max (phrase Sim fc , f))
    DK=DK+ fc
    FCS=FCS - fc
```

Figure 7: Unstructured Review-Mining Algorithm

to its semantically related words, new associations can be formed. E.g. picture can be propagated to its synonyms image, photo, pic, pics. The word excellent can be propagated to awesome, fantastic etc.

Next, we take the list of adjectives and the list of frequent nouns, and concatenate each word in the first list with every word in the second list (Code Reference: concatenation.java). This gives us a list of all-possible opinion feature tuples that may occur in the unstructured reviews. We also map the implicit features to their actual corresponding features, for example heavy is associated with weight, and expensive is associated with price. The sentiment orientation of the final list of nouns obtained is identified based on each words occurrence in the pros or cons list (this is why we separated the data set into Pros and Cons in the beginning to make identification easy). Our domain knowledge base is complete now (Figure 6).

5.3 Unstructured Review-Mining Engine

Here we make use of the domain knowledge base derived from semistructured reviews to assist product features extraction, followed by opinion polarity identification from the unstructured reviews. The approach (Figure 7) is fairly hybrid.

In the beginning we follow the same steps that we performed on semistructured reviews. We apply the NLProcessor to get the dependency tree and POS tags of the text. We apply extraction rules to get the appropriate words and phrases. From here we filter out the features, now called feature candidates, which are not true product features based on our domain knowledge base. In particular, each feature candidate is compared to every feature present in the domain knowledge base and is replaced by the one that is semantically most similar to it.

To find the semantic similarity between two words, Omiotis is used which relies on Wordnet. Omiotis takes as input two words (or two phrases, since some features can be in the form of phrases too e.g. battery life) and calculates their semantic similarity based on predefined formulae.

This algorithm takes every element of the Feature Candidate Set (i.e. the list of high frequency nouns and noun phrases obtained from the unstructured reviews) and searches for it in the Domain Knowledge. If the word is found, were done but if it is not, we replace it with the one that has the highest semantic similarity to it. This way we expand our domain knowledge base to include more features of the camera.

5.4 Feature-Opinion Tuple Generator

This is the last portion of the system, where the opinions for the most frequent features in the domain knowledge base are extracted. For each frequent feature, we extract its preceding word and the following word from the unstructured reviews file (Code Reference: oftuplesfromunstruct.java). For example, for the feature price, we extract all 3-tuples such that the middle word is price. The results could be: the price sucks, the price of, great price of. From the results, we remove the invalid/unwanted entries (such as the last 2 ones given in the above example) and just keep the entries that are meaningful (such as the price sucks). Now, for each entry in the result obtained above, we find its opinion polarity through the domain knowledge base (Code Reference: findingpolarity.java). For example, we locate the entry great price in the domain knowledge. If it is not present there, we discard it. Otherwise, we extract its polarity (recall that the domain knowledge file contains opinion feature tuples along with their polarities. So all we need to do is locate great price in the list and simply output its corresponding polarity). The newly found results, combined with the entire domain knowledge base, are our final answer, as displayed in the Results section below.

6 Results and Discussion

Two major results that we obtained during the course of this project were the two sets of opinion feature tuples, one from semi structured and other from unstructured reviews. The first set formed our Domain knowledge after we had added polarity to each opinion feature tuple. The second set was a little more complicated to obtain because of the fact that these reviews are written in natural language format. Therefore, making it difficult to extract opinions associated with each feature. We applied a strategy that differs from the one mentioned in the reference paper . We used the code oftuplesfrounstruct.java to extract one word before every feature and one word after it, to extract all opinion feature tuples of the format e.g. great camera, long battery life etc. The reason we obtained opinion feature tuples in the following format was to make the comparison between this set and the one obtained from semi structured reviews which are of the same format. Use of this form of extraction in unstructured reviews restricted the results we have obtained and we were not able to obtain every opinion from the data set. Also there were some extracted single opinion feature tuples that we could not use since we didn't have enough of them to form an overall orientation of user reviews towards that feature. Another shortcoming of this project was that we were not able to compare polarity of some obtained opinion feature tuples from unstructured reviews that were not present in the opinion feature tuples from semi structured reviews.

7 Future Work

Text mining and sentiment analysis is a field which has been around for a while. A lot of work and research has been done in this field but by no means have the opportunities for new research ideas exhausted. Sentiment analysis is that it is hugely reliant on good and effective text mining techniques. But the problem with this reliance is that it is largely applied on data collected from blogs and opinion/review websites where users writing reviews write in very informal language and it is very difficult to figure out the context in which an opinion statement was written. For example in the following sentence it is difficult to make a decision about what "It" refers to? "We watched the movie and went to dinner; it was awful." Also sarcasm, abbreviations, cultural factors, poor spelling, poor punctuation, poor grammar and slang makes it difficult for a sentiment analysis algorithm to make accurate decision about polarity of the reviews. All the above mentioned issues are open research fields in natural language processing (NLP) and sentiment analysis. Also, our projects approach to sentiment analysis is by no means the only way to do sentiment analysis. Researchers are coming up with new innovative approaches to sentiment analysis and it is still very much an open research area.

We have applied an approach in which we have tried to integrate two different sources to decide on polarity of user opinions (using SentiWordNet) on various features of a particular product (in this case digital canon camera). This approach can be extended by applying more refined NLP techniques e.g. anaphora resolution((the problem of resolving what a pronoun, or a noun phrase refers to the e.g. of an ambiguous statement mentioned above) to improve the mining process. Also further extension to this project can be to integrate customer reviews from other evaluative texts such as blog articles and communities.

8 Concluding Remarks

The algorithm under discussion is a recent development in the field of opinion mining and sentiment analysis. The intuition of this strategy is to combine the advantages of both the Semistructured and the Unstructured reviews of products available on the web in abundance. It is easy to see that opinion mining of unstructured reviews is much more challenging as compared to the reviews where some sort of structure is imposed. Extracting meaningful results from them is a difficult task, if carried out on its own. It is not very efficient, and extremely sophisticated algorithms are required for this purpose. Hence, the integration strategy proposed in this algorithm can act as a bypass. Verification of the effectiveness of this algorithm through statistical measures such as t-test, precision, recall and f-measure shows that this technique is much more very effective than any stand alone techniques available previously. In particular, while previous algorithms fail to recognize some domain sensitive opinion words in unstructured reviews,

this algorithm successfully identifies them due to the added availability of domain knowledge i.e. product feature information, thereby making it one of the more effective techniques of sentiment analysis.

9 References

1. Li and Zeng. Fine-grained Opinion Mining by Integrating Multiple Review Sources. 2010

2. Pang, B., Lee, L., & Vaithyanathan, S. (2002). Sentiment classification using machine learning techniques. In M. Lapata & H.-T. Ng (Eds.), Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 7986). College Park, MD: Association for Computational Linguistics.

3. Yu, H., & Hatzivassiloglou,V. (2003).Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In M. Lapata & H.-T.Ng (Eds.), Proceedings of the Conference on Empirical Methods in Natural Language Processing (pp. 129136). College Park, MD: Association for Computational Linguistics.

4. Hu and Liu. Mining opinion features in customer reviews. In Proceedings of 9th National Conference on Artificial Intelligence, 2004.

5. Thet, Na and Khoo. Aspect-based sentiment analysis of movie reviews on discussion boards. Nanyang Technological University, Singapore. 2010

6. Ding, X., Liu, B., & Yu, P.S. (2008). A holistic lexicon-based approach to opinion mining. In Proceedings of the First ACM International Conference on Web Search and Data Mining (pp. 231240). New York: ACM Press.

7. Feldman, R., Fresko, M., Netzer, O., & Ungar, L. (2007). Extracting product comparisons from discussion boards. In Proceedings of the Seventh IEEE International Conference on Data Mining (pp. 469474). Piscataway, NJ: IEEE.

8. Hu, M., & Liu, B. (2004a). Mining and summarizing customer reviews. In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 168177). NewYork: ACM Press.

9. Hu, M., & Liu, B. (2004b). Mining opinion features in customer reviews. In Proceedings of the 19th National Conference on Artificial Intelligence (pp. 755760). Cambridge, MA: MIT Press.

10. Miao, Q.L., Li, Q.D., & Dai, R.W. (2008a).A unified framework for opinion retrieval. In Proceedings of the 2008 IEEE/WIC/ACM Conferences on Web Intelligence (pp. 739742). Piscataway, NJ: IEEE.

11. Miao, Q.L., Li, Q.D., & Dai, R.W. (2008b). An integration strategy for mining product features and opinions. In Shanahan et al. (Eds.), Proceedings of the International Conference on Information and Knowledge Management (pp. 13691370). NewYork: ACM Press.

12. NLProcessor Linguistic Parser (www.infologistics.com/textanalysis.html)

13. Stanford Parser (http://nlp.stanford.edu/software/lex-parser.shtml)

14. Omiotis Demo (Jamal Nasir, Lahore University of Management Sciences)

15. http://www.amazon.com

16. http://www.cnet.com

17. http://www.epinions.com