Aspect-based Product Review Summarizer

Hsiang Hui Lek and Danny C. C. Poo
School of Computing, National University of Singapore, 13 Computing Drive, Singapore, Singapore

Keywords: Sentiment Analysis, Sentiment Summarization, Opinion Mining.

Abstract: Consumers are now relying on product reviews websites to aid them in deciding which product to buy. These sites contain large number of reviews and reading through them is tedious. In this work, we propose building a product review summarizer which will process all the reviews for a product and present them in an easy to read manner. The generated summaries show a list of product features or aspects and their corresponding rating, allowing users in comparing between different products easily. Our system first makes use of an aspect/sentiment extractor to extract the list of aspects and their sentiment words. Sentiment classification is then performed to obtain the polarity of aspects. Finally, these aspects are combined and assigned a rating to form the final summary. The experimental results on various domains have shown that our system is promising.

1 INTRODUCTION

In recent years, consumers are turning to product review websites to aid them in making informed decisions before making purchases. Consumers can also write product reviews on these sites, which in turn benefit other consumers. A typical review can contain general opinions about a product, such as *"I like the product"*, or can mention specifically about a particular product feature or aspect, such as *"The battery life is good"*. The latter provides valuable information for the consumers to understand the strengths and weaknesses of each product. Since there are many reviews for each product, and users can have differing opinions, it would be more credible to determine an overall opinion, rather than basing one's judgement on a single review.

In this work, we propose building a product review summarizer which will process all the reviews of a product, and summarize them so that users do not have to read all the reviews, but still benefit from this massive collections of reviews. Unlike conventional text summarizers which will extract key sentences/phrases, for product reviews, users are interested in the key product features or aspects and their corresponding rating so they can make comparison with other products easily.

Our system consists of a crawler which will crawl product review websites and aggregate the reviews of a product. Using an aspect/sentiment extractor that we devised, a list of aspects with the corresponding sentiments is extracted from each review. These sentiments are then classified using an aspect and domain sensitive sentiment lexicon to determine the sentiment polarity (whether positive or negative) of an aspect. Finally, these information are combined to form a summary of aspects with their individual ratings.

The rest of the paper is organized as follows. In section 2, we discuss the related work, specifically to aspect identification or extraction, and sentiment classification. In section 3 we describe our proposed system and its various components. We then present the experimental results in section 4. Finally in section 5, we conclude the paper.

2 RELATED WORK

The two essential tasks of a product review summarizer are aspect/sentiment word extraction and sentiment classification. For aspect/sentiment word extraction, some (Jo and Oh, 2011; Moghaddam and Ester, 2011) make use of unsupervised topic modelling approaches like Latent Dirichlet Allocation (LDA). Even though these approaches are unsupervised and able to detect latent aspects, most of them still require a domain expert to assign a set of fixed aspects to different topics before they can be usable in an application.

Linguistic approaches have also been proposed to identify aspects and sentiment words. Many (Hu

Hui Lek H. and C. C. Poo D..
Aspect-based Product Review Summarizer.
DOI: 10.5220/0004170302900295
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development* (KEOD-2012), pages 290-295
ISBN: 978-989-8565-30-3

and Liu, 2004; Zhang and Liu, 2011) have extracted noun phrases as aspects and adjectives as sentiment words. Similarly, we adopt a linguistic approach but instead of identifying noun phrases/adjectives, we use the results of a dependency parser and combine the dependency parse nodes into a structure that allows us to determine the aspect and their corresponding sentiments. Using dependency parse allows us to more accurately capture relationships between different tokens. Unlike work like Zhu et al. (2012) which considers fixed set of aspects, this work does not assume a fixed set of aspects.

For sentiment classification, many of the previous works have used general-purpose sentiment lexicon like General Inquirer (Stone, 1966), SentiWordNet (Esuli and Sebastiani, 2006) or Subjectivity Lexicon (Wilson et al., 2005) which assign a fixed polarity to a word. However, it has been observed that the polarity of words depends on the domain (Fahrni and Klenner, 2008; Pang and Lee, 2008). Although domain-specific lexicons like those generated in Du et al. (2010) and Lau et al. (2011) can better model the sentiment orientation of words, it is important to note that the sentiment of words may differ depending on how they are used even within a single domain. Consider the mobile phone domain, the word "cheap" is positive for the *price* product aspect but is negative for the *design* aspect. Thus, in order to accurately determine the sentiment of a word, we have to consider both its aspect and domain. In this work, we make use of an aspect and domain sensitive sentiment lexicon to determine the polarity of sentiments.

3 THE PROPOSED SYSTEM

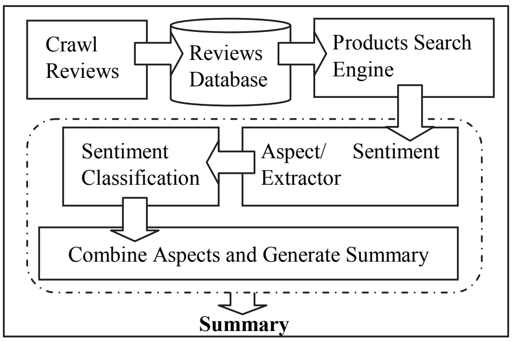

Figure 1: System Architecture.

3.1 System Architecture

Figure 1 shows a high level overview of the system. The system first crawl reviews from various reviews websites and store these reviews inside a reviews database. A user can then search for a product on the products search engine which is basically a website. On the website, the user can also ask for the summary of a product. The system will then send all the reviews of the product to the aspect/sentiment extractor which will extract a list of aspects and their corresponding sentiment words.

Thereafter, the system will perform sentiment classification to these words using an aspect and domain sensitive sentiment lexicon. Finally all the aspects/sentiments are combined together and a list of aspects with their corresponding rating (in terms of stars) is displayed to the user.

3.2 Aspect/Sentiment Extractor

It has been observed that there are relations between product features or aspects, and opinion words (Popescu & Etzioni, 2005; Qiu et al., 2009). As such, we describe a method to construct an aspect/sentiment extractor based on the results from Stanford Dependency Parser (Marneffe et al., 2006) and adopt their notation in this discussion. This extractor takes in product reviews text and extracts a list of tuple of this form: *[aspect, sentiment words]*.

Since English text usually follows the Subject-Verb-Object (SVO) format, we propose combining the results from the dependency parser which consist of a list of dependency paths between two tokens in the sentence, into a SVO structure (See Figure 2). Each node in our structure consists of the subject/verb/object, a list of modifiers and a list of negation flags associated with these modifiers. These modifiers are typically made up of adjectives or adverbs. The verbs are identified using the OpenNLP POS tagger (http://opennlp.apache.org/).

Subject and Object Identification: The subject associated with a verb can be obtained by looking for nodes with the following dependency relation: {nsubj, nsubjpass, nn, dep} that are having the verb as the head token. The object on the other hand, can be obtained from nodes with {dobj, iobj, attr} as dependency relation and the verb as the head token. Since Stanford dependency node only contains single tokens but the subject/object can be made up of multiple tokens, we have to combine nodes that are having these dependency relations: {prep, pobj, pcomp, nn, nsubj, dep}. {prep, pobj} are used to handle the part-of relations like in Girju et al.(2006).

Table 1: Types of phrases.

Phrase Pattern	Example
SVO	The player has good battery life
S	Large storage space
SV	The player works great
VO	Love the touchscreen
V	Works great

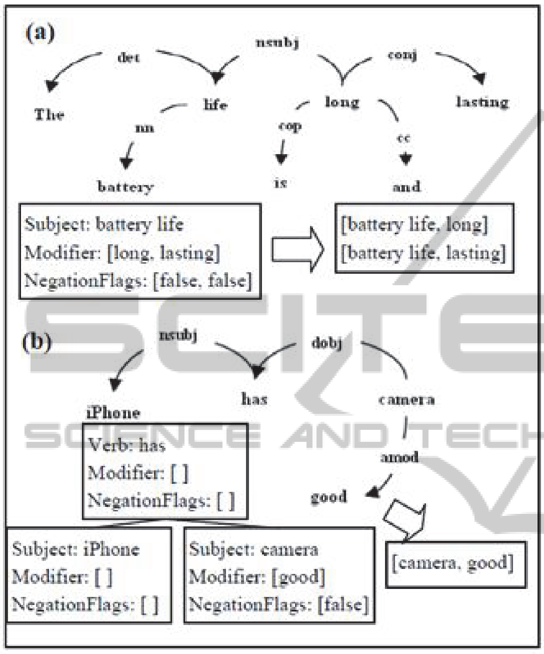

Figure 2: Example how SVO structures are generated.

The modifiers of the subject/verb/object are then obtained from node with these relations: {advmod, amod, xcomp}. Nodes with {conj} as dependency relation indicate the presence of conjunction cases like "and", "but", etc (See Figure 2a) which allows us to have more than one modifier. The boolean flag (NegationFlags) associated with each modifier is set to true whenever we detect the presence of the {neg} node, or for handling the "but" cases. We also capture the prepositions in order to handle the passive voice situations associated with the verb.

3.2.1 Determining the Aspect and Sentiment

After identifying the subject/verb/object, we can obtain different kinds of structures. Table 1 shows the different kinds of phrases/sentences. The next step is to determine the aspect and its corresponding sentiment words from these structures. For example, we want to get *[battery life, long]* and *[battery life, lasting]* in Figure 2a, and *[camera, good]* in Figure 2b.

The S case happens when a verb is not identified and the set of nodes usually consists of {advmod, amod} and {nn, dep}. The subject will be the aspect and the sentiment words will be the associated modifiers, which are mainly adjectives. For example, consider the phrase "good battery life", *battery life* is the aspect and *good* is the corresponding sentiment words.

The V case happens when we cannot identify any associated subject or object. In which case, the verb will be the aspect and the associated modifiers will be the sentiment words. For example, consider the phrase "work well", *work* is the aspect and *well* is the corresponding sentiment words.

For the SV case, the subject will be the aspect and the modifiers of the verb will be the corresponding sentiment words. There are situations where there are no modifiers associated with the verb. In such situations, the verb can be assigned as the sentiment words. For example, consider the phrase "the battery lasts", the *battery* is the aspect and *lasts* is the corresponding sentiment words. Similarly for the VO case, the subject implicitly refers to the reviewer, thus the object is the aspect and the verb is the corresponding sentiment words. The presence of "is" allows us to convert a SV case into an S case with modifiers (See Figure 2a). However, there are some situations for the SV case where we cannot just assign the subject as the aspect because the statement is just making a general comment of reviewer's sentiment about the product. Specifically, when we know that the subject of the statement is referring to the product instead, the aspect is set to be *general*. We check whether the subject matches the product title or is pronouns like "it", "they", etc. Verbs like "seem", "feel", etc also provide some clues to the possibility of this scenario.

For the SVO case, we further classify it into the "has" relation case, the "person reference" relation case and the "others" case.

Phrases/sentences classified as the "has" relation case take on the form "S has O". For example, "iPhone 4 has good camera" suggests that the product has the *camera* aspect. The object in this case would then be the aspect, and the modifier of the object would be the corresponding sentiment words (See Figure 2b). For situation where the object has no modifier, the subject is chosen to be the aspect and the object as the corresponding sentiment words.

Phrases/sentences classified as the "person reference" relation case take on the form "PERSON V O" where the subject is a person (I, he, she, etc), talking about a particular aspect. Therefore, the subject in this case should not be considered as the aspect. Instead the object should be the aspect, and the corresponding sentiment words are the modifiers of the objects, or the verb if there are no modifiers.

The “others” case refers to any other situation. It is possible that the verb is an aspect. Consider this example “The mp3 player works great for him”, *works* becomes the aspect and *great* is the corresponding sentiment words.

3.3 Sentiment Classification

The next step is to determine the polarity of the sentiment words of each aspect-sentiment pair generated from the aspect/sentiment extractor. In order to classify the sentiments, we make use of an aspect and domain sensitive sentiment lexicon generated in our previous work. This lexicon is unique in the sense that it is both domain and aspect sensitive. Each entry in a particular domain consists of a triple *[aspect, sentiment word, polarity]*. For example, a triple for the mp3-player domain is *[volume, low, negative]*. There are (as of now) about 58000 entries spanning 260 product domains.

To determine the orientation of an aspect-sentiment pair, we look up the lexicon using the domain, the aspect, and the sentiment word. The experimental results of this lexicon will be presented in the next section.

3.4 Combining Aspect and Generating Summary

3.4.1 Identifying Implicit Aspect

Implicit aspects are identified before combining the list of aspects to form the product summary. Implicit aspects (as opposed to explicit aspects) are defined as aspects which are not explicitly written in the review text. Nevertheless, they can be inferred based on the sentiment words. For example, the sentiment word *small* indicates that the reviewer could be talking about the *size* aspect.

Since our lexicon contains a list of aspects with their corresponding sentiment word, it can be used to infer implicit aspects. In order to identify implicit aspects (for pairs having *general* as aspect), we make use of equation 1 to determine the most probable aspect given sentiment word and product domain. Equation 1 is made up of two components. The first component determines whether it is possible for a word to be found in an aspect. Assuming uniform distribution, each possible aspect is set to be 1 / N where N is the total number of aspects in lexicon that can contain this sentiment word w in domain d. The second component gives more weight to aspects with fewer number of sentiment words. The idea is similar to the inverse document frequency (idf) in information retrieval. The rationale is as such: if an aspect is only described by a few sentiment words, when we see one of the few sentiment words, it is more likely to uniquely imply this aspect rather than a common aspect which has many words describing it.

$$\arg\max_i P(aspect_i \mid w,d)\left[\frac{1}{P(aspect_i,d)}\right] \quad (1)$$

$$P(aspect_i \mid w,d) = \begin{cases} \frac{1}{N} & \text{if } N > 0 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$P(aspect_i,d) = \frac{\sum_j freq(w_j, aspect_i, d)}{\sum_j \sum_k freq(w_j, aspect_k, d)} \quad (3)$$

where N =total number of aspects that can contain this sentiment word w in domain d, w = sentiment word, d = domain

3.4.2 Combining Aspects and Generating Summary

Since there are many ways of describing the same aspect, some aspects need to be merged together so that the final summary is more compressed. For example, “picture quality”, “picture”, “photo” and “image” should be merged together. To generalize aspects, we make use of WordNet (Fellbaum, 1998) and a version of Pointwise Mutual Information (PMI) similar to Turney and Littman (2003). WordNet is used to generalize single-token aspects. Aspects having the same synsets are merged together. Multi-token aspects are handled by PMI (equation 4), aspects pairs are merged together if their PMI value exceeds a certain threshold.

$$PMI(aspect_i, aspect_j) = \log_2\left(\frac{\text{hits}(\text{"}aspect_i\text{" AND "}aspect_j\text{"})}{\text{hits}(\text{"}aspect_i\text{"}).\text{hits}(\text{"}aspect_j\text{"})}\right) \quad (4)$$

where hits(query) = number of hits from Google search engine with supplied query

After merging similar aspect, the next step is to identify prominent aspects and filter away aspects which are not useful. An aspect is selected if it fulfils equation 5 and 6. Specifically, an $aspect_i$ is selected if there are at least $THRESHOLD_{cross}$ reviews having this aspect. This is to model “cross reviews” aspects where multiple reviews affirm that this $aspect_i$ is significant. Equation 6 is just to check that there are at least *THRESHOLD* number of $aspect_i$*-sentiment word* extractions before choosing this aspect. In our experiments, $THRESHOLD_{cross}$ is empirically set to 3 and *THRESHOLD* is set to 10.

$$freq(reviews\_with\_aspect_i) > THRESHOLD_{cross} \quad (5)$$

$$freq(aspect_i) > THRESHOLD \quad (6)$$

Rating stars are then assigned accordingly to this final list of aspects using equation 7 and 8.

$$percentage = \frac{freq(positive)}{freq(positive) + freq(negative)} \quad (7)$$

$$rating_i = \begin{cases} 1 \text{ star} & \text{if } 0 >= \text{percentage} < 0.2 \\ 2 \text{ stars} & \text{if } 0.2 >= \text{percentage} < 0.4 \\ 3 \text{ stars} & \text{if } 0.4 >= \text{percentage} < 0.6 \\ 4 \text{ stars} & \text{if } 0.6 >= \text{percentage} < 0.9 \\ 5 \text{ stars} & \text{if percentage} >= 0.9 \end{cases} \quad (8)$$

4 EXPERIMENTAL RESULTS

4.1 Qualitative Evaluation

Due to space constraint, an extract of the generated summary for two products (iPod and iPhone) is shown in Figure 3. We can see that the generated aspects are good representations of the important aspects of the products. Users can click on an aspect to see list of sentiment words that make up the aspect. "^" preceding a sentiment word indicates that it is a negation case.

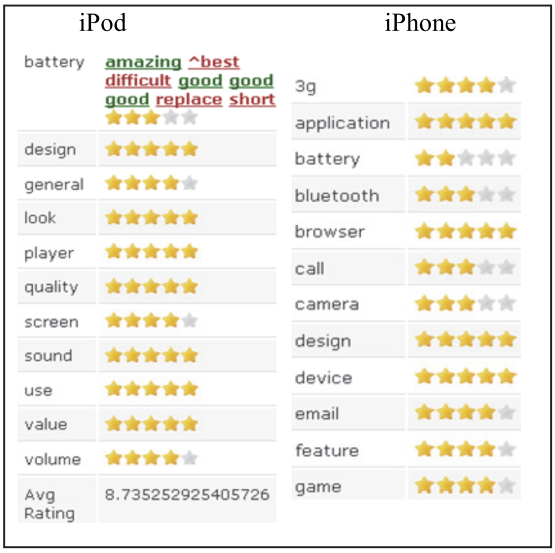

Figure 3: Extract of the Generated Summaries.

4.2 Quantitative Evaluation

We evaluate our system based on the sentiment classification performance and the final summary generated. Table 2 shows the sentiment classification performance (in terms of Precision/Recall/F1-measure) of our lexicon (Sentix) for three different product domains in comparison with two commonly used sentiment lexicons: SentiWordNet (Esuli and Sebastiani, 2006) and SubjLex (Wilson et al., 2005). We see that Sentix significantly outperform the other lexicons.

To evaluate the performance of the summary, we make use of product reviews from Reevoo (http://www.reevoo.com) in various product domains. For each domain, we selected the top ten products with the most reviews. Since Reevoo provides a rating (on a scale of 1 to 10), for each product, we determine the average rating from these reviews and compare it with the system generated average. The system generated average is computed by calculating the average of all the aspects of a product. Mean-squared error (MSE) (Equation 9) is used to measure the accuracy of the generated average. Table 3 shows MSE in thirteen different domains. We see that the MSE is usually less than 1 suggesting that the generated average rating by the system is very close to the actual average.

$$MSE_{domain} = \frac{1}{|p|}\sum_p (average_{review,p} - average_{system,p})^2 \quad (9)$$

where p = a product in this domain (10 in our case),
$average_{review,p}$ = average rating of product based on reviews,
$average_{system,p}$ = average rating of product generated by the system,

Table 2: Sentiment classification results.

Domain	Lexicon	P	R	F1
Mp3 Player	Senti-WordNet	0.7500	0.5436	0.6303
	SubjLex	0.9223	0.6620	0.7708
	Sentix	0.9357	0.7108	0.8079
Digital Camera	Senti-WordNet	0.7640	0.5037	0.6071
	SubjLex	0.8910	0.6666	0.7627
	Sentix	0.9279	0.7629	0.8373
Mobile Phone	Senti-WordNet	0.8333	0.6884	0.7539
	SubjLex	0.9633	0.7600	0.8502
	Sentix	0.9512	0.8478	0.8965

Table 3: Generated average rating compared to actual average rating (scale of 1 to 10).

Domain	MSE
Mp3 Player	0.4722
Vacuum Cleaner	1.0230
Digital Camera	0.4327
Printer	0.5029
Television	1.0427
Baby Products	0.1623
Washing Machine	0.6845
Fridge-Freezer	0.8149
Software	0.5195
Cooker	1.0354
Laptop	0.9737
Mobile Phones	0.8148
Toys	0.9229

5 CONCLUSIONS

In this paper, we propose building a product review summarizer which will process all the reviews of a product and summarize them in a manner that is easy for reading and comparison. The summarizer first extracts a list of aspects along with their corresponding sentiment words. After classifying the polarity of these sentiment words, we can determine the polarity associated with these aspects. It then combines different aspects together to form a summary consisting of a compressed list of aspects and their ratings. The experimental results demonstrate that the summarizer is accurate and promising. Our future work will focus on enhancing the aspect/sentiment extractor to learn extraction rules automatically. We are also looking into better visualization and product comparison mechanisms.

REFERENCES

Du, W., Tan, S., Cheng, X., & Yun, X., 2010. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proceedings of the third ACM international conference on Web search and data mining. WSDM'10*. pp. 111–120.

Esuli, A. & Sebastiani, F., 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC-06*.

Fahrni, A. & Klenner, M., 2008. Old wine or warm beer: target-specific sentiment analysis of adjectives. In *Symposium on Affective Language in Human and Machine, AISB 2008 Convention*. pp. 60–63.

Fellbaum, C. ed., 1998. *WordNet: An Electronic Lexical Database*, MIT Press.

Girju, R., Badulescu, A. & Moldovan, D., 2006. Automatic Discovery of Part-Whole Relations. *Comput. Linguist.*, 32(1), p.83–135.

Hu, M. & Liu, B., 2004. Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Seattle, Washington, pp. 168–177.

Jo, Y. & Oh, A.H., 2011. Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining. WSDM'11*. pp. 815–824.

Lau, R. Y. K., Zhang, W., Bruza, P. D., & Wong, K. F., 2011. Learning Domain-Specific Sentiment Lexicons for Predicting Product Sales. In *2011 IEEE 8th International Conference on e-Business Engineering (ICEBE)*. pp. 131 –138.

Marneffe, M., Maccartney, B. & Manning, C., 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In *Proceedings of LREC-06*. pp. 449–454.

Moghaddam, S. & Ester, M., 2011. ILDA: interdependent LDA model for learning latent aspects and their ratings from online product reviews. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. SIGIR'11*. pp. 665–674.

Pang, B. & Lee, L., 2008. *Opinion mining and sentiment analysis*, Now Publishers.

Popescu, A.-M. & Etzioni, O., 2005. Extracting Product Features and Opinions from Reviews. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*.

Qiu, G., Liu, B., Bu, J., & Chen, C., 2009. Expanding Domain Sentiment Lexicon through Double Propagation. In *International Joint Conference on Artificial Intelligence*. pp. 1199–1204.

Stone, P. J., 1966. *The General Inquirer: A Computer Approach to Content Analysis*, The MIT Press.

Turney, P. D. & Littman, M. L., 2003. Measuring Praise and Criticism: Inference of Semantic Orientation from Association. *ACM Transactions on Information Systems (TOIS)*, 21(4), pp.315–346.

Wilson, T., Wiebe, J. & Hoffmann, P., 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*. pp. 347–354.

Zhang, L. & Liu, B., 2011. Identifying noun product features that imply opinions. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers - Volume 2*. HLT'11. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 575–580.

Zhu, J., Zhang, C. & Ma, M., 2012. Multi-Aspect Rating Inference with Aspect-based Segmentation. *Affective Computing, IEEE Transactions on*, PP(99), p.1.