

Aspect Extraction using Informative Data from Mobile App Data Review

Budi Eko Prasetyo
Electronics Engineering Program
Politeknik Negeri Malang

Divi Galih Prasetyo Putri
Electrical and Informatics
Engineering Department
Vocational School
Universitas Gadjah Mada

Endang Wahyu Pamungkas
Informatics Department
Faculty of Communication and
Informatics
Universitas Muhammadiyah
Surakarta

ABSTRACT

User review of mobile application is an valuable data that can be used by developer to improve their application or to build similar application. User can give feedback such as reporting errors, asking for new or improved feature, explain their experience of using certain feature and also praise or dispraise. User review or opinion data is very large in amount and difficult to analyze. It is time consuming and labour expensive to do it manually. Recent study has tried to extract product feature using word collocation. In this work, we try to improve the aspect extraction process by using only informative data. We took user opinion of 3 mobile application from application distribution platform. The experiment result indicate that our approach is able to improve the performance of collocation finding method..

General Terms

Information retrieval

Keywords

Aspect extraction; aspect-based sentiment analysis; collocation; naive bayes;

1. INTRODUCTION

Huge amount of mobile device users influence the high number of available application in the Appstore. There are three widely used appstore including Google PlayStore, Apple AppStore, and Windows Phone Store. These platforms provide services to easily find and deploy application for user [1]. In addition, appstore also provide space for user to give feedbacks about the performance of application. They can deliver about their satisfaction, explain about some bugs, and request about new feature [2]. Several previous research stated that user feedback on appstore is a valuable information for the developer [3] [4]. User can give rating and plain text review as a feedback. It can be simple sentiment (e.g “Hate this app!”) or a specific topic that very relevant for software maintenance and evolution [5]. Developer could use these data to make analysis for developing the application based on user needs. There are high competition among developer that build similar apps. Disapointing and ignoring user will cause the developer a loose in the market share [6]. They do not need to conduct some survey that usually very budget and time consuming. Furthermore, these data also can be used as a resource to analyze market for the other developers who want to make similar application.

However, developer may face some limitations and obstacles

in order to retrieve the information from appstore. First, the huge amount of the data will rise some difficulties for the developer to analyze the important feedback one by one [7]. A prior study found that popular apps such as facebook get 4,275 feedbacks from user for each day [1]. Second, the diversity form of the user feedback will produce the other difficulties to build the suitable method to process the data. Some feedbacks state the good and bad point of some features clearly, but some of them explain it implicitly. Then, out of vocabulary (OOV) words are found on some feedbacks that make another challenge to process the data. Third, feedback from user are usually short as they write it from mobile phone. It is shorter than common non-mobile-software product review. A study found that app store user usually write a short but informative review. They analyze 8.7 million review from 13.770 apps [8]. Every application also has many version of release and every user feedbacks is correspond to a specific version [9].

Some previous studies tried to implement aspect based sentiment analysis or aspect based opinion mining [10] [11] [12]. In this type of sentiment analysis, the processes are focused on the sentiment of each aspect or product features. One of the important processes is aspect or feature extraction. Several methods are developed to handle these problems such as collocation finding method [13], linguistics rule method [10], and also the combination of these two methods [14]. Collocation method try to extract the aspect by taking combination of two words which is most frequently used. While, linguistics rule method is done by building some linguistics rules to find the aspect of the feedback. However, these approaches are still lack in accuracy. One of the research explain that these methods still have some disadvantage in distinguishing between word or word combination which one is aspect and which one is non aspect. It happens because there are some opinion data that are not contain aspect are included in the aspect extraction. Some research try to classify user review into several classes to get informative review [15] [16] [17] [18]. Chen et al. uses naive bayes to filter “non-informative” reviews, however the category rules of might not be always true [15]. Another research have tried to classify the opinion data in to 4 different categories including bug report, user experience, feature request, and rating [16]. This research try to compare different methods for opinion classification. They found that naïve bayes was the most powerful method to classify this data. This study also stated that rating data that consist of un-relevant criticism, praise and also dispraise are not useful because they are not informative.

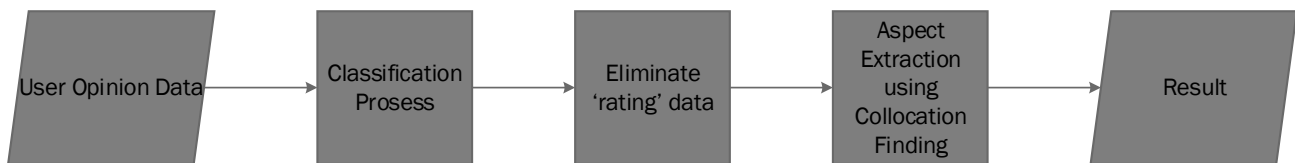


Figure 1. Proposed Method

In this research, we try to combine opinion data type classification and aspect extraction. It will use collocation finding to extract the aspect. Before extracting the aspect, opinion data is classified into bug, user experience, feature request, and rating by using naive bayes method. Aspect extraction is implemented only for the data that is included in bug, user experience, and feature request. By excluding rating data is expected to improve the performance of aspect extraction process.

The remainder of this paper is structured as follows: Section 2 explains works related to aspect extraction topics, section 3 describes an overview of our proposed method, section 4 presents the performance of our proposed method and some findings from the result, section 5 concludes our research and ideas for the future works.

2. RELATED WORK

Several research has been conducted that focused on the methods for extracting aspects from user opinion data. Prior study tried to implement association mining to mine opinion feature or aspect from product reviews of electronic product [19]. Ding et al. try to improve Hu and Liu's method by using holistic lexicon-based method [20]. Zhang et al. introduced a developed double propagation method to extract product features. This research uses dataset from 4 non-software product [21] and able to calculate feature relevance. Other method try to make use of the opinion word to extract features [22]. Soujanya et al. utilize a rule-based approach to extract aspect. It is focused on extracting explicit and implicit aspect [10]. Beside unsupervised methods, supervised methods can be used to handle product aspect extraction. One of them used lexicalized Hidden Markov Model [23]. In addition, research on semi-supervised modelling also has been conducted. It utilized some seed words for few aspect categories and use it to cluster aspect terms [24].

However, our research is focused on user opinion data from application distribution platform (*app store*) as a dataset. Data from app store is tend to be shorter than non-software product review. There are many types of user review from app store. One research focused on review that contains user request for new features. It uses linguistic rules to extract requested features from app-store data [3]. While others try to extract user complains and its relationship with the ratings [25]. Guzman et al. proposed collocation finding method to extract all of the software feature. It is based on the most frequently mentioned words in the opinion data [13]. Other study, use the data from appstore to find out features that are requested by the user [3].

3. RESEARCH METHODOLOGY

Error! Reference source not found. shows the flow of the proposed method. The aim is to improve the performance of aspect extraction process. There are two main process, data

classification and aspect extraction. The opinion data is first classified into classes. We eliminate several data that is not used in the aspect extraction process. After that, we use collocation finding method to perform aspect extraction process.

3.1 Opinion Classification

The data classification process is utilizing naive bayes method. Previous study said that naive bayes is the best method to classify user opinion data. Opinion data is classified into 4 classes, bug report, feature request, user experiences and rating. Opinion data that contains user report of an error, performance issue or system crash are classified as *bug report*. *Feature request* is when user ask for a new feature or upgrade on a existing feature. Data that describe user experience on a product feature is called *user experience*. *Rating* contains user explanation of the star that they give.

3.2 Eliminate Data Rating

Previous study stated that opinion data can be classified into 4 types. One of them is data rating. Data rating contains user statement that explain star rating given by the user. It is about praise, dispraise, and distractive critique. This type of data is not containing much information about the product. Below are the example of user opinion that classified as rating:

- *I love this app, but since iOS 9 came out it s baaaad*
- *I just felt like it to put a 1 star.*
- *Very bad*

Therefore, in order to extract more relevant information from the opinion data, we eliminate rating data. We only use bug, feature request and also user experiences data in the aspect extraction process.

3.3 Aspect Extraction

After data rating are eliminated from the overall data, we do aspect extraction for the remaining data. We use collocation finding method to extract aspects. However, there are some additional preprocess steps such as POS Tagging, Stopword Removal, Spelling Correction, and Lemmatization. The proses flow is as seen in Fig 2.

Previous study stated that only the nouns, verbs and adjectives usualy describe product features [2]. We use POS Tagging to define the Part-of-Speech (POS) of each word in the opinion sentences. Then, we eliminate words that belong to other POS like adverbs or quantifiers. Common english terms such as "this" or "and" are also removed from the data. We use Enchant API to check and revise words that are spelled wrongly. In addition, WordNet Lemmatization is used to group words with the same meaning.

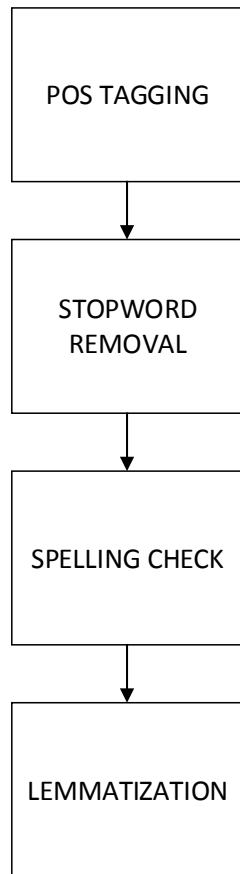


Figure 2. Preprocess Flow

Next, aspect extraction process is done by using collocation finding method. Collocation defines two or more words that usually used together to explain things. In this experiment, we use two word collocation or called bigram. This two word can be separated by several words. Generally, collocation can be used to describe aspect because aspect contains terms that used to explain a specific meaning. In collocation finding method, we calculate the frequency of each word pair. Likelihood ratio test is used to help finding the collocation. We only take word pair that used at least three times in reviews.

4. RESULT AND DISCUSSION

4.1 Dataset and Evaluation

We used user opinion data as dataset. The dataset was taken from 3 different domain application from appstore. From each application, we took 250 opinion data. So, we use a total of 750 user opinion data. We established a gold standart that contains the opinion data and the aspect stated in the opinion. We used this standart to evaluate our proposed method. The result is compared to collocation finding method. We calculated the precision, recall and f-measure value of each application. For each user opinion, we defined the value of True Positive (TP), False Positive (FP) and False Negative (FN). We gives TP when the feature that extracted from our proposed method is also written in the gold standart. FP are features extracted by our proposed method but not in the gold standart for that specific user opinion. Lastly, a feature is a FN if it is manually extracted in the gold standart but not by our proposed method

4.2 Result

We implemented our proposed method using Python and

NLTK [26]. **Error! Reference source not found.** is the result of our experiment. The result shows that the proposed method is able to get higher value of precision and recall compared to the collocation method. The overall f-measure value comparison from collocation and our proposed method is as seen at Fig 3. F-Measure value presents the balance between precision and recall value. In conclusion, removing data that do not contain important information could improve the performance of aspect extraction process. It happens because collocation method takes word pair with highest frequency as aspects. If we include the data rating in the extraction process, it will affect the frequency of word pairs. Many irrelevant aspect tend to be generated from it. The proposed method is able to eliminate irrelevant aspects and extract some additional aspects. The collocation extracts approximately 370 aspects. On the other hand, the proposed method extract 350 aspects, 20 aspects less than the collocation method. There are some aspect that is extracted by collocation method but not by our proposed method such as “*everything need*”, “*get bad*” and “*great love*”. These aspects are not relevant as application features and affect the performance of the extraction process.

Table 1. Experiment Result

	Proposed Method			Collocation		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
Line	0,532	0,573	0,551	0,432	0,549	0,484
ibook	0,514	0,487	0,500	0,442	0,411	0,426
Evernote	0,506	0,506	0,506	0,463	0,481	0,472

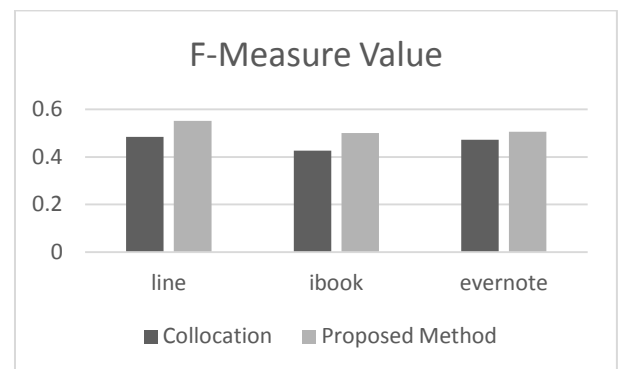


Figure 3. F-Measure Value Graph

However, we can see from the table that the performance of our proposed method is still far from perfection. There are se that affect it. First, the performance of the classifier. There are some data that misclassified. Some that contain bug or important information are classified as rating. These data are eliminated and not being processed in the aspect extraction process. For example the sentences “*i wish i can use timeline function in ipad version....PLEASE*” is classified as rating. But, we can see that this opinion is about asking for a new feature or should be classified as feature request. There are also some rating review that misclassified as other class.

Second, our proposed method is not able to handle aspect that is not frequently mentioned. We only consider word pair that is mentioned at least three times as an aspect. However, there

are aspect that is not mentioned by many people such as when people share their experience on specific features or request a new feature. For example, in the sentence “Doesn't have block/ignore button.” the aspect is block/ignore button. Only one person asked for this new feature. So that, frequency of the word pair will be low and not considered as aspect.

5. CONCLUSION AND FUTURE WORK

Features or aspects extraction is an important process in aspect-based sentiment analysis. This research proposed a method that combine opinion classification and aspect extraction to improve the performance. Online review for mobile application was used to extract product aspects. Opinion data is classified into several classes. Proposed study only utilizes data that is informative based on the classification result. Aspect extraction is done by using collocation finding method. In the experiment, we compare the precision, recall and f-measure value of our proposed method to the performance of collocation finding method itself. It can be seen from the result that eliminating uninformative data from the dataset could improve the performance. There are several findings from the experiment. The performance of the classifier as the first problem. Second, our proposed method is not able to extract aspects that are not frequently mentioned in the data. The future plan is to improve the classification performance in order to get a better accuracy of aspect extraction.

6. REFERENCES

- [1] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in In 2013 21st IEEE international requirements engineering conference (RE), 2013.
- [2] S. Panichella, A. Di Sorbo, E. Guzman and C. A. Visaggio, "How can i improve my app? classifying user reviews for software maintenance and evolution," in IEEE International Conference on Software Maintenance and Evolution (ICSME), 2015.
- [3] C. Iacob and R. Harrison, "Retrieving and analyzing mobile apps feature requests from online reviews," in 10th IEEE Working Conference on Mining Software Repositories (MSR), 2013 , 2013.
- [4] L. V. Galvis Carreno and K. Winbladh, "Analysis of user comments: an approach for software requirements evolution," in In Proceedings of the 2013 International Conference on Software Engineering, 2013.
- [5] A. Di Sorbo, S. Panichella, C. V. Alexandru, J. Shimagaki, C. A. Visaggio, G. Canfora and H. Gall, "What would users change in my app? summarizing app reviews for recommending software changes," in In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering, November.
- [6] L. Villarroel, G. Bavota and B. Russo, "Release Planning of Mobile Apps Based on User Reviews," in ICSE 16th Proceeding of the 38th International Conference of Software Engineering, 2016.
- [7] N. Genc-Nayebi and A. Abran, "A Systematic Literature Review: Opinion Mining Studies from Mobile App Store User Reviews.," Journal of Systems and Software., pp. 207-219, 2016.
- [8] R. Vasa, L. Hoon, K. Mouzakis and A. Noguchi, "A preliminary analysis of mobile app user reviews," in OzCHI '12 Proceedings of the 24th Australian Computer-Human Interaction Conference, 2012.
- [9] B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong and N. Sadeh, "Why people hate your app: Making sense of user feedback in a mobile app store," in In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, 2013.
- [10] S. Poria, E. Cambria, C. Gui, A. Gelbukh and L. W. Ku, "A Rule-Based Approach to Aspect Extraction from Product Reviews," in Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP), 2014.
- [11] Y. Jo and A. H. Oh, "Aspect and sentiment unification model for online review analysis," in In Proceedings of the fourth ACM international conference on Web search and data mining, 2011.
- [12] R. Feldman, "Techniques and applications for sentiment analysis," Communications of the ACM, vol. 56, no. 4, pp. 82-89, 2013.
- [13] E. Guzman and W. Maleej, "How Do Users Like This Feature? A Fine Grained Sentiment Analysis of App Reviews," in Requirements Engineering Conference (RE), 2014 IEEE 22nd International, 2014.
- [14] D. G. Putri and D. O. Siahaan, "Software Feature Extraction using Infrequent Feature," in The 2016 International Annual Engineering Seminar (INAES), 2016.
- [15] N. Chen, J. Lin, S. C. Hoi, X. Xiao and B. Zhang, "AR-miner: mining informative reviews for developers from mobile app marketplace," in In Proceedings of the 36th International Conference on Software Engineering ACM, 2014.
- [16] W. Maleej and H. Nabil, "Bug Report, Feature Request, or Simply Praise? On Automatically Classifying App Reviews," Requirements Engineering (RE'15)., 2015.
- [17] J. Oh, D. Kim, U. Lee, J. G. Lee and J. Song, "Facilitating developer-user interactions with mobile app review digests," in In CHI'13 Extended Abstracts on Human Factors in Computing Systems, 2013.
- [18] S. McIlroy, N. Ali, H. Khalid and A. E. Hassan, "Analyzing and automatically labelling the types of user issues that are raised in mobile app reviews," Empirical Software Engineering, vol. 21, no. 3, pp. 1067-1106, 2016.
- [19] M. Hu and B. Liu, "Mining opinion features in customer reviews," in Proceedings of the International Conference on Knowledge Discovery, 2004.
- [20] X. Ding and B. Liu, "A holistic lexicon-based approach to opinion mining.," in In Proceedings of the 2008 international conference on web search and data mining, 2008.
- [21] L. Zhang and L. Bing, "Extracting and ranking product features in opinion documents," in Proceedings of the 23rd International Conference on Computational Linguistics: Posters, 2010.
- [22] W. Hu and Z. Gong, "Mining Product Features from Online Reviews," in IEEE International Conference on E-Business Engineering, 2010.

- [23] W. Jin, H. H. Ho and R. Srihari, "OpinionMiner. A novel machine learning system for web opinion mining and extraction," in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009.
- [24] A. Mukherjee and B. Liu, "Aspect extraction through semi-supervised modeling," in In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, 2012.
- [25] Khalid, E. Shihab, M. Nagappan , A. Hassan and E. Ahmed, "What do mobile app users complain about?," IEEE Software, vol. 32, no. 3, pp. 70-77, 2015.
- [26] S. Bird and E. Klein, Natural language processing with Python, O'Reilly Media, Inc, 2009.