Vol 4 (2023) No 1

Jitsi

Jurnal Ilmiah Teknologi Sistem Informasi

URL: http://jurnal-itsi.org ISSN 2722-4619 (Print) ISSN 2722-4600 (Online)

Classifying User Apps Review for Software Evolution: A Preliminary Experiment

Mutia Rahmi Dewi[#], Hidayatul Munawaroh[*], Siti Rochimah[**]

[#]*Jurusan Teknologi Informasi, Politeknik Negeri Padang, Limau Manis, Padang, 25164, Indonesia*
[*]*Departemen Sistem Informasi, Universitas Internasional Semen Indonesia, Sidomoro, Gresik, 61122, Indonesia*
[**]*Departemen Informatika, Institut Teknologi Sepuluh Nopember, Keputih, Surabaya, 60111, Indonesia*
E-mail: mutia@pnp.ac.id, hidayatu.munawaroh@uisi.ac.id, siti@if.its.ac.id

ABSTRACTS

Application Store is a platform where users can download several applications and games. Users also can provide comments about related applications. These comments made as evaluation material for developers, who have not yet developed applications in the future. In previous studies, an application user assessment has been carried out based on existing taxonomies such as feature requests, information provision, information retrieval, and problem discovery by using Natural Language Processing (NLP), Text Analysis (TA) and Sentiment Analysis (SA). In this study, we propose a model using Topic Modelling (TM) and Minority Synthetic Over-Sampling Technique (SMOTE) to improve classification results. Making user reviews that previously ignored can be taken into consideration for developers in conducting software development. Topic modelling will generate list of topics that representing each review and SMOTE method can overcome the amount of imbalanced data on several tables. We also combine methods TA + NLP + SA, TA + NLP + SA + TM, and TA + NLP + SA + TM + SMOTE with J48 classifier. In this study, can be seen the combination of TA+NLP+SA+TM+SMOTE+J48 method gives the highest result with 84.9% precision, 84.3% recall, and 84.6% F-Measure

Manuscript received 19 Nov. 2022; revised 16 Dec. 2022; accepted 22 Mar. 2023 Date of publication 31 Mar. 2023.

Keywords— ***Natural Language Processing; Sentiment Analysis; Text Analysis; Topic Modelling User Apps Review.***

1. INTRODUCTION

Apps Store or Google Play provides various applications that can be used by user. Besides downloading apps, users can also give reviews on related applications. The review given can be in the form of comments and ratings. Developers can evolve and improve the applications and software based on the user needs, by the reviews, which contain praise or advice. From existing reviews, the developer can obtain information such as new features, bug fixes, or improvements of features that already exist in the application.

Application developers need a lot of time and energy in gathering application reviews and finding information relevant to the evolution of related software. Previous studies [1], [2], [3] showed that one-third of the information in a review is useful for developers. However, analysing user reviews to obtain valuable information for developers, this has several challenges. Pagano et al. [3] found that the average application on the Play Store or Google Store receives around 23 reviews per day. In contrast, for popular applications like Facebook, it can reach an average of 4,275 reviews per day. User reviews tend to have unstructured sentences that are difficult to analyse. So, the developer must read most of the app's reviews to find out what the user wants. Besides, user reviews vary greatly, such as giving ideas for new features, bug reports, praise, or complaints.

Previous research has carried out [4] to obtain important information from user reviews that are useful for developers. The user reviews have been classified based on pre-existing taxonomies [4]. The steps are taken, are pre-processing, TF-IDF weighting, Natural Language Processing (NLP), and Sentiment Analysis. And the last step is doing classification using J48 classifier. A combination of Text Analysis, NLP, and sentiment analysis has

done to get the best results. The results of this research were 75.2% precision and 74.2% recall by combining three above-mentioned methods. From these results can be seen, that there are still many reviews that are misclassified. This can make it possible, that information from the review will not reach the developer.

We propose a new model using Topic Modelling and Minority Synthetic Over-Sampling Technique (SMOTE). The model will be an additional method from previous studies [4] so that information from user reviews can be conveyed more to the developer. The process to use is to add a topic modelling approach. Then a combination of Text Analysis, NLP, Sentiment Analysis and topic modelling approach will be carried out. The topic model approach used is Latent Dirichlet Allocation (LDA). From this study, the results obtained 84.9% precision and 84.3% recall.

The rest of the paper organized as follow. Section 2 presents our research methodology. Section 3 present our results and enumerates the threats to validity in our case study. Finally, we present the conclusion and future work in Section 4

2. RESEARCH METHODOLOGY

The purpose of this research is to help developers to group information from reviews, which are relevant to software maintenance and evolution. The stages carried out in this study can be seen in Figure 1

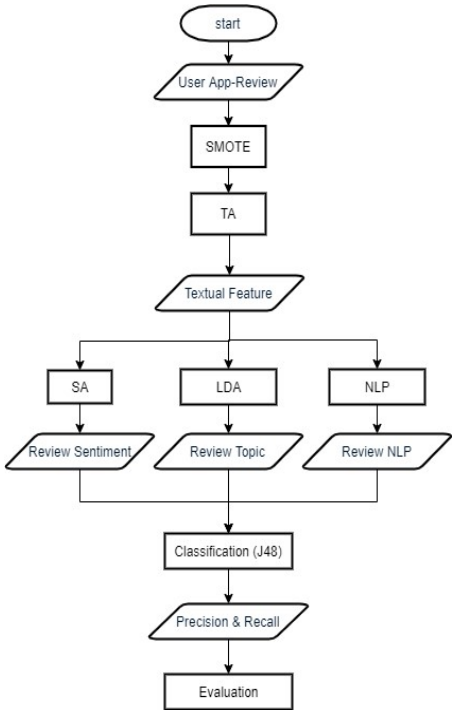

FIG 1. Proposed Method

2.1. Dataset

The datasets used in this study are reviews of the AngryBirds, Dropbox and Evernote application datasets on Apple's App Store and a review of the TripAdvisor, PicsArt, Pinterest and Whatsapp application datasets on Android's Google Play Store. This data was obtained in 2013 by Guzman and Maalej [6]. The number of reviews of each application can be seen in Table 1

TABLE 1. Number of Dataset Review Apps [6]

App	Platform	Category	Reviews
AngryBirds	App Store	Games	1538
Dropbox	App Store	Productivity	2009
Evernote	App Store	Productivity	8878
TripAdvisor	App Store	Travel	3165
PicsArt	Google	Photography	4438
Pinterest	Google	Social	4486
Whatsapp	Google	Communicati	7696

In Table 1, the "App" column describes the name of the application as a sample of the dataset to be tested. The "Platform" column is a platform that houses related applications. The "Category" column describes the type or category of the related application. And the "Reviews" column describes the number of reviews from each related application

2.2. SMOTE

SMOTE (Minority Synthetic Over-Sampling Technique) is good and effective oversampling technique to handle overfitting in the oversampling process to deal with imbalances in the module class that are defective in the minority class (positive) [5].

In this study, we use the Weka tools to perform SMOTE techniques to handle the problem of class imbalance on input data. Figure 2 shows the raw data before doing the SMOTE technique. There are shown 101 data on "Information Seeking" label, 583 data on "Information Giving" label, 218 data on "Feature Request" label, and 488 data on "Problem Discovery" label. The results of the data after the SMOTE technique shown in Figure 3, these are 808 data on the Information Seeking label, 583 on the Information Giving label, 988 on the Problem Discovery label, and 768 on the Feature Request label. With the use of SMOTE labels with less data, it has more data so that the model of training results can better present each label in a review.

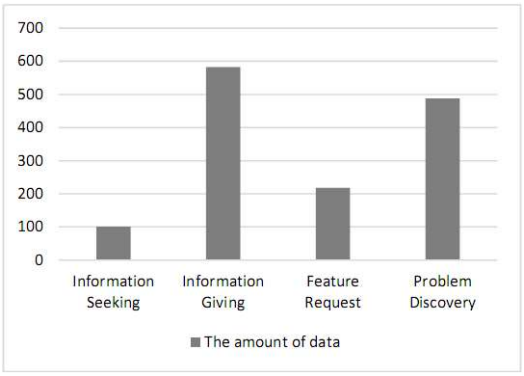

FIG 2. Diagram of Dataset Before SMOTE

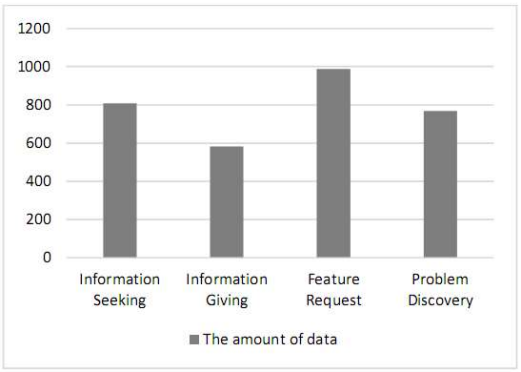

FIG 3. Diagram of Dataset After SMOTE

2.3. Text Analysis

The text analysis was performed to extract features from the app review. There are two steps, tokenization and weighting of text features using Term Frequency.

1. Tokenization

   At this stage only do tokenization instead all preprocessing stages since on the experiment, the use of stemming and stop-word removal caused important words that should have wasted. For example, in the sentence "Make it like better with a giant pig bigger than king pig" labelled "Feature Request", from the example sentence the most important word should be "make", but if done stemming and stop-word removal then the word " make "will be removed, so the main word that will be detected is" like ". With the main word "like" then the review will be more likely to label "Information Giving". While tokenization, user reviews split into tokens. The selection of tokens based on separator such as (space; comma; &; *; !; \ T; \ n; \ r; number).

2. Term Frequency (TF)

   At this stage, the weighting of each term in the application review carried out after tokenization. TF represents the frequency of words (terms) appearing in the document. More frequent a term appears the weight of the term will increase. In this study, one document is equal to an application review.

   The use of IDF in this study make terms that should be important terms, which have a low weight, because many words repeatedly appear in many reviews. Application reviews tend to have words that are almost similar if they included in the same label. For example, the Problem Discovery label will often find the words "bug", "failed", etc. So that the use of TF only for weighting the term in the application review, makes the term with a high frequency of appearance has a high weight as well.

2.4. Natural Language Processing

The use of NLP in this study aims to determine the most important words from a review. In previous studies, there was an assumption that users tend to use repetitive linguistic patterns in writing application reviews [6]. From this assumption, it found that the sentence that is compatible with repetitive linguistic patterns can mapped into categories of existing taxonomies. Therefore, Natural Language Processing (NLP) is needed to identify repetitive linguistic patterns. The application of heuristic NLP allows automatic detection of a sentence that matches a particular structure.

In this study, we use the Stanford Typed Dependencies parse, a python library that can represent dependencies between individual words contained in sentences and label each with a specific grammatical relationship [7]. For example, in the sentence "Make it like better with a giant pig bigger than king pig". NLP will break down each word in a sentence and look for the structure of the sentence.

- "Make" as root
- "it" as an object
- "like" as case
- "better" as obl
- "with" as case
- "a" as det
- "giant" as compound
- "pig" as obl
- "bigger" as amod
- "than" as case
- "king" as compound

From the mapping of word structure in the example sentence, it can be seen that the main word representing the sentence is " Make". "Make" can be grouped as a Feature Request label so that the phrase "Make it like better

with a giant pig is bigger than king pig" can be grouped into the Feature Request label. This NLP will apply to all review sentences used in this study.

2.5. Sentiment Analysis

Textual information can generally divide into fact and opinion information [8]. Facts are objective expressions of an object, its events, and its possession. Opinions can be in the form of subjective expressions, that describe a person's sentiments, judgments, or feelings about an object, event or possession of that object. Peng, et al. explained that sentiment analysis is part of the work that reviews everything related to computational opinions, sentiments, and subjectivity of texts [9]. Sentiment analysis is a tool to process the collection of search results aimed at finding the attributes of a product (quality, features, etc.) and the process of obtaining the results of opinion [10].

Sentiment analysis is used to detect sentiment from an application review. In this study, sentiment class is divided into two, positive and negative. We do sentiment analysis using Azure Machine Learning, which is an Excel add-in. Azure Machine Learning empowers data scientist and developers to transform data into insight using predictive analytics. By making it easier for developers to use the predictive models in end-to-end solutions, Azure Machine Learning enables actionable insights to be gleaned and operationalized easily [11].

2.6. Topic Modelling

Topic modelling is an algorithm for finding major themes that include a large and unstructured set of documents. Topic modelling can arrange documents according to themes that have been found. To large documents, topic modelling algorithms can be applied. Some advancements in this field have enabled the analysis of streaming documents like those in the Web API. Topic modelling algorithms can also be used in various types of data. Another application in the topic modelling algorithm is to find patterns in genetic data, images, and social networks [12].

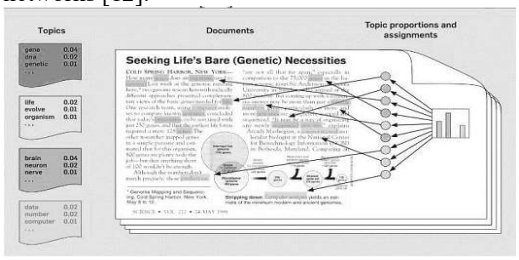

FIG 4. How LDA Works [13]

Topic modelling was done using the Latent Dirichlet Allocation (LDA) rhythm algorithm. LDA is a generative probabilistic model of discrete data collection such as a text corpus. LDA is a Bayesian three-level hierarchy model, where each collection item modelled as a limited mix of a set of topics. Each topic modelled as an infinite mix through a set of underlying topic probabilities. In the context of making text models, topic probabilities provide an explicit representation of a document [13]. Illustration of LDA can be seen in Figure 4

To make an LDA model a corpus containing, a token collection is needed. The first step in making a corpus for the LDA model is making a bigram. The making of bigram aims to make phrases consisting of 2 words that often appear together so that it will give more meaning and helps to identify topics. After a bigram document formed, filtering out the words that appear too many and too few based on the frequency of their appearance in the document. The corpus made from the dictionary of each document, which has been filtered in the form of a bag of words, namely the frequency of occurrence of each word.

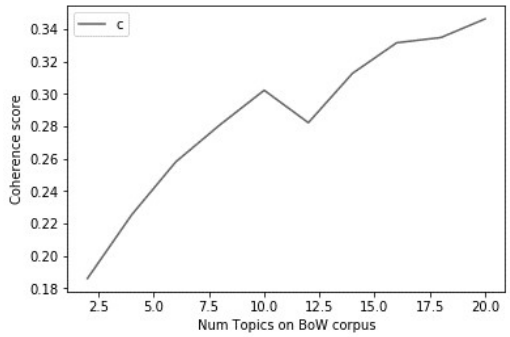

FIG 5. How LDA Test Result

The determination of the optimal number of topics for the LDA model done by calculating the coherence score of the model. If the coherence score is higher, so the topic is better and easier to interpret to the mode, each document will test on the model with the highest value that has made before. The model will provide the probability values of topics relevant to the tested document. The topic selection for the document did by selecting the topic with the highest probability.

In this study, we produced the best LDA model of eight topics with a coherence score of 0.346. Figure 5 shows the number of topics covered by each user review

2.7. Methodology Evaluation

The results of the classification method described in section 2. We make a comparison between the results of the classification with the truth set manually labelled using predetermined metrics. The dataset with the truth set

that we used is a dataset used in previous studies by Panichella and Di Sorbo [4]. The truth set labelling was done by sampling getting 1421 reviews from a total of 7696 reviews (18.46%).

Distribution of labels after labelling truth sets manually can be seen in Table 2. The "Category" column in Table 2 describes the taxonomic results that have carried out in previous studies. However, there are additional "Others", which used to accommodate sentences that do not fit into the categories of available taxonomies. The "Reviews" column in Table 2 describes the number of reviews based on their category. Finally, the "Proportion" column in Table 2 describes the percentage of reviews from each category.

Our evaluation uses precision, recall and F-measure metrics that commonly used in machine learning. For evaluations, a comparison made between truth sets manually traded and the results of the classification. Calculations for evaluations can be seen in equations (1), (2) and (3).

$$P = TP / (TP + FP) \quad (1)$$
$$R = TP / (TP + FN) \quad (2)$$
$$F = (P \times R) / (P + R) \quad (3)$$

Where TP (True Positive) is the number of instances, that correct to classify based on its class. FP (False Positive) is the number of instances that fail classification based on the four categories mentioned in Table 2, FN (False Negative), namely the number of instances who failed to classify into other classes.

TABLE 2. Number of Dataset Review Apps [6]

Category	Review	Proportion
Information Seeking	101	0.071
Information Giving	583	0.410
Feature Request	218	0.153
Problem Discovery	488	0.343
Others	31	0.022
Total	1421	1

3. RESULTS AND DISCUSSION

3.1. Result

Finally, we compared the combination of each method used. Comparison conducted is a combination of TA + NLP + SA, TA + NLP + SA + Topic Modelling (TM), and SMOTE + TA + NLP + SA + TM.

Among several combinations of each method, the combination of SMOTE + TA + NLP + SA + TM gets the best results using the alternating decision tree (AD Tree) classifier J48, with a precision of 84.9% and a recall of 84.3%. While the lowest results with a combination of TA + NLP + SA methods and using the J48 classifier obtained 64.4% precision and 65.8% recall. Details of the experimental results for each combination of SMOTE, TA, NLP, SA, and TM shown in Table 3.

TABLE 3. Classification Result Using J48

Methods Combination	Precision	Recall	F-Measure
TA+NLP+SA	64.4%	65,8%	63,4%
TA+NLP+SA+TM	64,4%	66,3%	63,6%
SMOTE+TA+NLP+SA+TM	84,9%	84,3%	84,6%

In this study, we classify using the J48 classifier, because the performance of the J48 classifier increases with the addition of training data [14]. Classification with the J48 classifier performed using Weka tools with 8-fold cross-validation and confidence factor 0.17.

The combination of TA + NLP + SA + TM shows an increase in the results on the Recall and F-Measure values when compared to the combination of TA + NLP + SA. The use of topic modelling as one of the features for classification review shows, that reviews which has the same topic, tend to have the same label. The combination of SMOTE + TA + NLP + SA + TM shows a significant increase in results. The use of SMOTE for balancing data on each label has shown to provide improvements to the training model.

With the amount of data in each label that has a balance, the model of training results can better represent what features represent each label. The following are the detailed results of the TA and SA method experiments based on the label shown in Table 4. The best precision value obtained on “Problem Discovery” label, the best recall value obtained on “Information Giving” label, and the F-Measure value obtained on “Problem Discovery” label.

TABLE 4. Comparison of Results Each Label

Label	Precision	Recall	F-Measure
Feature Request	87%	84,5%	85,7%
Information Giving	84,6%	71,5%	67,9%
Problem Discovery	86,4%	86,4%	86,4%
Information Seeking	96,2%	91,2%	93,2%
Weighted Avg.	84,9%	84,3%	84,6%

3.2. Discussion

From the results of experiments that have carried out, it can be seen in Table 4, that although it has the highest precision, recall, and f-measure values when analyzed there are still low precision and recall values compared to

other labels. That is because the amount of data on the label is small so that the model formed is still not representative of the label. This can be overcome by, when the process of making truth sets, sampling is done with the number of instances taken from each class having almost the same amount. So, there is no imbalance of the data.

Also, the value of precision and recall decreased when the LDA method added. This is because several instances are categorized into the same topic category even though they have different labels. This results in the features used for classification be underrepresented

4. CONCLUSIONS

In this research, we present an approach using Text Analysis (TA), Sentiment Analysis (SA), and modelling of the topic Latent Dirichlet Allocation (LDA) to detect and classify sentences in-app reviews, which can help developers carry out maintenance and evolution on the application. Classification results obtained from taxonomic results carried out through review analysis and email development in previous research [15]. The best results obtained by using a combination of TA and SA and the alternating decision tree (ADTree) classifier, namely J48 with a precision of 65.1% and a recall of 65.5%.

We find that with TA and SA as a feature for classification, it gives better results than the combination of the TA, SA and LDA methods. This is because several instances in the dataset are categorized into the same topic category even though they have different labels.

From this research, developers can: 1) sort out relevant information from user reviews; 2) decide what maintenance/software evolution should be done next; and 3) more responsive to user requests.

In the future, we plan to develop this research by selecting a dataset that has balanced data. So, the model that results from classification can represent each label.

REFERENSI

[1] N. Chen, J. Lin, S. C. H. Hoi, X. Xiao, and B. Zhang, "AR-miner: mining informative reviews for developers from mobile app marketplace," in Proceedings of the 36th International Conference on Software Engineering, 2014, pp. 767–778.

[2] L. V Galvis Carreño and K. Winbladh, "Analysis of user comments: an approach for software requirements evolution," in Proceedings of the 2013 International Conference on Software Engineering, 2013, pp. 582–591.

[3] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in 2013 21st IEEE international requirements engineering conference (RE), 2013, pp. 125–134.

[4] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can i improve my app? classifying user reviews for software maintenance and evolution," in 2015 IEEE international conference on software maintenance and evolution (ICSME), 2015, pp. 281–290.

[5] A. Fernández, S. Garcia, F. Herrera, and N. V Chawla, "SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," J. Artif. Intell. Res., vol. 61, pp. 863–905, 2018.

[6] E. Guzman and W. Maalej, "How do users like this feature? a fine grained sentiment analysis of app reviews," in 2014 IEEE 22nd international requirements engineering conference (RE), 2014, pp. 153–162.

[7] T. Peng, I. Harris, and Y. Sawa, "Detecting phishing attacks using natural language processing and machine learning," in 2018 ieee 12th international conference on semantic computing (icsc), 2018, pp. 300–301.

[8] P. Mehta and S. Pandya, "A review on sentiment analysis methodologies, practices and applications," Int. J. Sci. Technol. Res., vol. 9, no. 2, pp. 601–609, 2020.

[9] H. Peng, E. Cambria, and A. Hussain, "A review of sentiment analysis research in Chinese language,"Cognit. Comput., vol. 9, no. 4, pp. 423–435, 2017.

[10] S. Shayaa et al., "Sentiment analysis of big data: methods, applications, and open challenges," IEEE Access, vol. 6, pp. 37807–37827, 2018.

[11] S. Rajagopal, K. S. Hareesha, and P. P. Kundapur, "Performance analysis of binary and multiclass models using azure machine learning.," Int. J. Electr. Comput. Eng., vol. 10, no. 1, 2020.

[12] J. A. Khan, L. Liu, and L. Wen, "Requirements knowledge acquisition from online user forums," IET Softw., vol. 14, no. 3, pp. 242–253, 2020, doi: 10.1049/iet-sen.2019.0262.

[13] A. Zaim, J. Ahmad, N. H. Zakaria, G. E. Su, and H. Amnur, "Software Defect Prediction Framework Using Hybrid Software Metric," Int. J. Informatics Vis., vol. 6, no. 4, pp. 921–930, 2022, doi: 10.30630/joiv.6.4.1258

[14] H. Jelodar et al., "Latent Dirichlet allocation (LDA) and topic modeling: models, applications, a survey," Multimed. Tools Appl., vol. 78, no. 11, pp. 15169–15211, 2019.

[15] S. Aljawarneh, M. B. Yassein, and M. Aljundi, "An enhanced J48 classification algorithm for the anomaly intrusion detection systems," Cluster Comput., vol. 22, no. 5, pp. 10549–10565, 2019.

[16] B. G. Glaser and A. L. Strauss, Discovery of grounded theory: Strategies for qualitative research. Routledge, 2017.