App Review Mining and Summarization

Article *in* International Journal of Computer Applications · April 2018

DOI: 10.5120/ijca2018916918

Rabeya Sultana
Prime University Bangladesh

App Review Mining and Summarization

Rabeya Sultana
Ahsanullah University of Science and Technology
Dhaka, Bangladesh

Sujan Sarker
Ahsanullah University of Science and Technology
Dhaka, Bangladesh

ABSTRACT

With the development of the web, online reviews are more important and essential information resource for people. Opinion mining and summarizing aims at extracting features and opinions and classify them as positive or negative. In this work, we develop a review mining and summarization technique and apply it to summarize the reviews of apps from Google Play App Store. Different from traditional text summarization, the features of apps are extracted based on customers opinions, classified them as positive or negative and ranked the apps based on the ranking of each feature. We propose two approaches, SentiWordNet 3.0 based and Naïve Bayes algorithm to classify opinions and find scores. The result of two approaches is quite similar. The experimental results show the effectiveness of the proposed approach in app review mining and summarizing.

General Terms

Supervised algorithm, Naïve Bayes, SentiWordNet 3.0.

Keywords

Opinion Mining, Sentiment Analysis, Summarization, App Review Analysis.

1. INTRODUCTION

Most of the peoples nowadays would like to share their views, emotions, feelings and opinions on the web. People usually use forums, blogs and social networking platforms such as Facebook, Twitter etc. to express their opinions and views. The World Wide Web plays a vital role in assembling public opinions. These opinions are very beneficial for business organization as they would know the user's opinions about the products and also beneficial for customers to taking decisions. Large amount of user content data is generated on the web every day, thus mining the data and identifying user opinions, likes and dislikes is one of the essential tasks. Opinion mining is a relatively recent discipline that studies the extraction of opinions using information retrieval, artificial intelligence and natural language processing techniques. More informally, it's about extracting the opinions or sentiments given in a piece of text [1, 2, 3]. ***It's about finding out what other people think.*** "*What other people think*" has always been an important piece of information for most of us during the decision-making process. Opinion mining mines the information from various text forms such as reviews, news and blogs and classifies them on the basis of their polarity as positive, negative or neutral. It focuses on categorizing the text at the level of subjective and objective nature. An objective sentence represents some factual information without any opinion content, while a subjective sentence expresses some personal feelings or beliefs and indicates that the text contains opinion.

In this paper, we propose an approach that automatically extracts app features referred in the reviews together with the user opinions about them. We refer to a feature as a prominent or distinctive visible characteristic or quality of an app [4]. It can be any description of specific app functionality visible to the user (e.g., "video calling" or "sending a message"), a specific screen of the app (e.g., "configuration screen"), a general quality of the app (e.g., "load time", "size of storage", or "price"), as well as technical characteristics (e.g., "encryption technology"). Our approach produces a list of features mentioned in the reviews. It then extracts the user sentiments of the identified features and gives them a general score across all reviews. Finally, we suggest the developers and the users with these features importance, lacking, bug fixing etc.

2. RELATED WORK

Sentiment analysis of natural language texts is a large and growing field. Previous work particularly relevant to our task falls naturally in two groups. The first relates to techniques to automatically generate sentiment lexicons. The second relates to systems that analyze sentiment (on a global or local basis) for entire documents.

Kim and Hovy [5] evaluate the sentiment of an opinion holder (entity) using WordNet to generate lists of positive and negative words by expanding seed lists. They assume that synonyms (antonyms) of a word have the same (opposite) polarity. The percentage of a word's synonyms belonging to lists of either polarity was used as a measure of its polarity strength, while those below a threshold were deemed neutral or ambiguous. Their best results were achieved when the topic neighborhood consisted of words between the topics up to the end of the sentence.

Pang, Lee and Vaithyanathan [6] perform sentiment analysis of movie reviews. Their results show that the machine learning techniques perform better than simple counting methods. They achieve an accuracy of polarity classification of roughly 83%. In [7], they identify which sentences in a review are of subjective character to improve sentiment analysis. Nasukawa and Yi [8] identify local sentiment as being more reliable than global document sentiment, since human evaluators often fail to agree on the global sentiment of a document.

In [9], they follow up by employing a feature-term extractor. For a given item, the feature extractor identifies parts or attributes of that item. e.g., battery and lens are features of a camera. There are a number of articles has been published on the sentiment analysis of movie [10] and product [11] reviews etc. with elaborate consideration towards natural language processing and understanding, subjectivity detection and opinion identification, feature selection and extraction, classification, language models etc. Pang and Lee [12] presented survey on sentiment analysis and opinion mining where they explained opinion oriented information access, challenges, opinion classification and summarization. Like them MikalaiTsytsarau, Themis Palpanas [13] also have presented Survey on opinion mining. In that survey author explained opinion mining, opinion aggregation and subjectivity analysis. And their study also mentioned different work performed on this issue and their comparisons. Earlier sentiment analysis performed on different domain data such as Movie [14], Products [15] and Travel [5] etc.

Many authors applied opining mining concept to social network data. In [6] author worked on sentiment analysis of Facebook data from messages written by users. Many researches [7] developed sentiment analysis applications on twitter data. Other issues in opinion mining are emotion recognition, opinion spam detection [13]. In [5] author proposed different methods such as machine learning, machine translation and dictionary for sentiment analysis of text in Hindi language. Many researchers used machine learning methods for sentiment analysis [6, 8, 9, 16, 17] that involve training of classifier on datasets and use the trained model for new document classification. Some authors suggested another method such as dictionary of word lexicons [18, 19]. The Dictionary approach is based on a prebuilt dictionary that contains opinion polarity values of words. Many resources consider the data which contain subjective data but not the objective data. Reviews are retrieved by query based information retrieval techniques [18]. Siddhi Patni, AvinashWadhe [20] in this paper major tasks, various challenges, and applications of sentiment analysis. Their most of the work has been done on product reviews like documents that have a definite topic. Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, and Emery Jou [21] in this paper sentiment classification are applied to the movie reviews, and rating information is based on sentiment-classification results. Changbo Wang, Zhao Xiao, Yuhua Liu, YanruXu, Aoying Zhou, and Kang Zhang [22] in this paper a text-based sentiment mining method and a model driven prediction approach is used to analyze the public sentiments on hot topic. SentiView can be used to analyze and visualize mass Web information effectively in many applications. SentiView builds upon and extend several ideas from state-of-the-art techniques to enable advanced visual analysis of public sentiments on popular topics on the Internet. Three system components designed in SentiView are showing rich information from different aspects at once and provide flexibility for varying task. LisetteGarcía-Moya, Henry Anaya-Sánchez, and Rafael Berlanga-Llavor [23] in this paper a new methodology for the retrieval of product features from a collection of customer reviews about a product or service is that it doesn’t require any training set of product features, and over several collections of customer reviews in English.

3. PROPOSED SECTION

This section represents the proposed approach for performing opinion mining and techniques and algorithms are used to perform Opinion Mining and Sentiment Analysis for getting useful information from online app reviews.

In this era of mobile technology mobile applications commonly known as mobile apps has become a very important thing in our life. And Android is the largest mobile application platform holding around 78% market share [24] of mobile apps platform. Android apps distribution platform Google Play allow users to search, buy and install apps from thousands of apps uploaded by the developers. Users also can give their opinions about apps in form of textual review and star rating.

Google Play Store user reviews contain a unique author ID, Review Creation Time, Rating (ranging from 1 to 5), Review Text. Star rating reflects the overall evaluation of user’s experiences using a numeric scale of one to five star but textual comments are more capable of giving more information about the apps that the overall ratings cannot. After few months of a new app is launched in the market, there could be over ten thousand textual comments from users. It is very challenging for a potential user to read all the comments one by one. For example, very popular apps such as Facebook get more than 4000 reviews per day. The review quality also varies user to user. They may contain helpful advice, innovative ideas about features as well as insulting comments. Third, a textual review generally holds a mixed sentiment. So it is very difficult to filter out the positive and negative feedback or retrieve the feedback for specific features. This can be made easier with sentiment analysis.

As App stores like Google Play allow users to submit feedback for downloaded apps in form of star ratings and text comments. User reviews serves as a valuable source of information for evaluating a mobile app, for both new users and developers. Interactive systems can be developed to summarize and visualize these reviews which will be beneficial for both the parties.

This system is to summarize the App review according to App features and the methodology consists of following steps:

I. Identify and extract object features that have been commented on by an opinion holder.

II. Determine whether the opinions on the features are positive, negative or neutral.

III. Identify the top features of the opinion.

IV. Identify the best apps for each feature.

The system will be beneficial for both developers and users. As a developer, it is essential to “stay on top of your app”, i.e., keep app updated with the most requested features. Consequently, it is difficult to identify why people like or dislike a particular. The motive of the work is to solve this problem.

3.1 Implementation

For implementation, we use Python 2.7.8 and NLTK 2.0. As NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, etc. [25].

There are many machine learning methods to implement the system. In this work, two machine learning methods are considered to implement, semi-supervised and supervised learning method. For semi-supervised learning method, SentiWordNet 3.0 is used to classify the opinion and also for find the score of that opinion.

SentiWerdNet 3.0 is an improved version of *SentiWerdNet 1.0*. SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity [26]. Each of the three scores ranges in the interval [0.0 – 1.0], and their sum is 1.0 for each synset.

For supervised learning method, there are many algorithm Decision trees, Support Vector Machines (SVM), Naive Bayes, Nearest neighbor, Ensembles etc. In this work, Naïve Bayes algorithm is used to classify the opinion and also for find the score of that opinion.

The Naïve Bayes classification is a supervised learning technique as well as a statistical technique for classification. This method assumes an underlying probabilistic model and it allows for the capture of uncertainty about the model, in a principled way by determining probabilities of the outcomes [27].

Naïve Bayes classifier formula [28]:

$$\mathrm{P}(\mathrm{Sentiment} \mid \mathrm{Sentence}) = \frac{\mathrm{P}(\mathrm{Sentiment}) \times \mathrm{P}(\mathrm{Sentence} \mid \mathrm{Sentiment})}{\mathrm{P}(\mathrm{Sentence})}$$

The Bayes Theorem:

$$P(h \mid D) = \frac{P(D \mid h) \times P(h)}{P(D)}$$

P(h) : Prior probability of hypothesis h

P(D) : Prior probability of training data D

P(h/D) : Probability of h given D

P(D/h) : Probability of D given h

3.2 Components of the Model

This section presents the architecture and functional details of the proposed system to identify features and opinions of that features for summarizing and ranking. Figure 1 presents the complete architecture of the proposed system, which consists of seven different functional components – review documents, split into sentences, preprocessing, feature and opinion learner, defining polarity and score for opinion phrase, feature scoring, summarizing and ranking. Further details about these modules are presented in the following sub-sections.

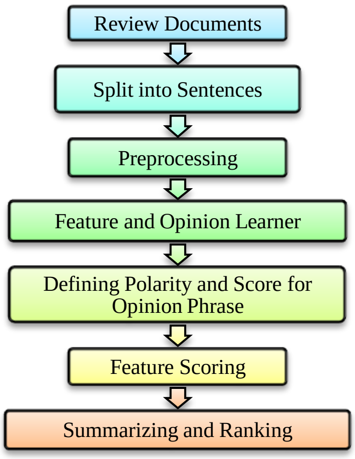


Figure 1: Proposed System Architecture

For google play app store site, user reviews have retrieved from the site [29]. The review comments will be proceeding for more processing steps. It has been found that noisy reviews that are used to increase or decrease the popularity of the Android Apps. To remove noise from the texts, the input texts need to be preprocessed. Before preprocessing the text, reviews need to be retrieved from the site [4]. To retrieve texts from the site we used a python library Beautiful Soup.

Preprocessing is an important task and critical step in Opinion Mining. In the area of Opinion Mining, text preprocessing is used to remove noise from unstructured text data. There are some activities in text preprocessing like tokenization, punctuation removal, stop word removal, normalization which will be applied to get a structure of the noisy text. Because the text data often contains some special formats like number formats, date formats and the most common words that unlikely to help Text mining such as prepositions, articles and pronouns can be eliminated.

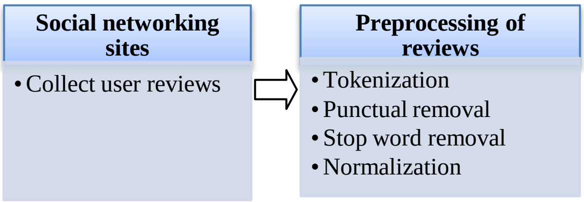


Figure 2: Steps of Text Preprocessing

Tokenizationis the process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens. The aim of the tokenization is the exploration of the words in a sentence. The list of tokens becomes input for further processing such as punctuation and stop word removal.

Punctuation removal is used to remove any unnecessary symbol from the text. Unnecessary symbols are !, @, #, $, %, ^, *, :, (,) etc.

Many words in document are used to join words together in a sentence is called stop word removal. But they are essentially meaningless. Stop words are very frequently used common words like ‘and’, ‘are’, ‘this’ etc. They are not useful in classification of documents. So we can remove stop word from text from classification.

Case Normalization is a process that has English texts to be published in both higher and lowercase characters and turns the entire document or sentences into lowercase/uppercase.

Complete preprocessing steps:

Example:

Input text: "The App has Fantastic Sound Quality and It's Video is Awesome :) :).

Table 1: Preprocessing steps for a sentence

Tokenization	['The', 'App', 'has', 'Fantastic', 'Sound', 'Quality', 'and', 'It', "'s", 'Video', 'is', 'Awesome', ':', ')', ':', ')', '.']
Punctuation Removal	['The', 'App', 'has', 'Fantastic', 'Sound', 'Quality', 'and', 'It', 's', 'Video', 'is', 'Awesome']
Stop Word Removal	['App', 'Fantastic', 'Sound', 'Quality', 'It', 'Video', 'Awesome']
Case Normalization (Lower Case)	app fantastic sound quality and it video awesome

Part of Speech (pos) tagging is the problem of assigning each word in a sentence the part of speech that it assumes in that sentence. Tags for parts of speech are nouns, verbs, adverbs, adjectives, articles etc. Sub tagging is nouns can be singular or plural, verbs have tenses.

Example:

Input text: "Viber is a great app and easy to use"

Output: [('Viber', 'NNP'), ('is', 'VBZ'), ('a', 'DT'), ('great', 'JJ'), ('app', 'NN'), ('and', 'CC'), ('easy', 'JJ'), ('to', 'TO'), ('use', 'VB')]

Table 2: POS tag of above example

Word	Tag
Viber	Noun
is	Verb
a	Determiner
great	Adjective
app	Noun
and	Conjunction
easy	Adjective
to	To
use	Verb

3.3 Feature Extraction

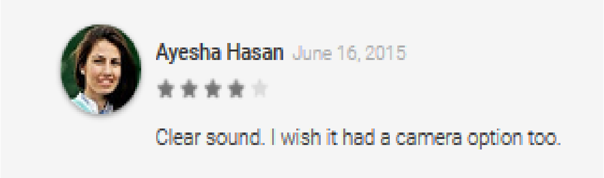
Features are a set of words that represents the quality of an app. The feature set is identified by retrieving the noun POS tagger. We assume that features can only be singular, plural or common nouns (NN, NNS). Thus, all the words with POS tags NN or NNS are considered in our feature set. Those NNPs (proper nouns) are ignored which can never occur as a common noun (singular or plural) as they are usually names like John, Steve etc. which are not features. After extract nouns (NN, NNS), those words are checked from the dictionary *SentiWordnet 3.0* to remove the unnecessary or wrong words from the feature set.

Algorithm for feature extraction is shown below-

Algorithm 1: (Feature Extraction)

```
for each sentence in the review documents
  if (it contains noun (NN, NNS) extract all the word from the
      sentence)
      for each word (NN, NNS) in the sentence
          check word from dictionary and record the verified
          word as a feature
```

Example:



Ayesha Hasan June 16, 2015

★★★★☆

Clear sound. I wish it had a camera option too.

Figure 3: A review example of Viber app [29]

In the example, sentence is "clear sound. I wish it had a camera option too."

After POS tag, tags of each word will be-

[('clear', 'JJ'), ('sound', 'NN'), ('.', '.'), ('I', 'PRP'), ('wish', 'VBP'), ('it', 'PRP'), ('had', 'VBD'), ('a', 'DT'), ('camera', 'NN'), ('option', 'NN'), ('too', 'RB')]

Here, sound and camera option are noun (NN). So, from above review, we can identify the features **Sound** and **Camera option**.

Feature Set = {Sound Quality, Camera Option}

Let, f_j denotes the feature set extracted from a jth review-

f_j = {Word$_i$ ϵ Review$_j$: Word$_i$ is a noun}

Extracting Opinion Value for a Particular Feature:

After identify the feature set, it is need to identify the opinion words. Opinion words are those words that user use to express positive or negative opinion. Adjectives are useful for determining positive, negative or neutral opinion. So adjectives are used as opinion words and also opinion words from sentences are extracted having at least one feature.

Algorithm for opinion word extraction is shown below-

Algorithm 2: (Opinion Word Extraction)

```
for each sentence in the review documents
  if (it contains a frequent feature, extract all the
      adjective words as opinion)
       for each feature in the sentence
           the nearby adjective is recorded as its effective
opinion
```

In the example [figure 3], feature is "Sound" and the nearby adjective of that feature is "**Clear**". So, **clear** is an opinion of the feature sound and the opinion is positive opinion.

Opinion Classification using SentiWordNet:

The aim of this work is to classifying the reviews as positive or negative using the SentiWordNet. This proposed technique consists of two phases. The first is to assign scores (positive and negative) to the words. The second phase applies various calculations on reviews to classify them as positive or negative using words scores got from the first phase.

SentiWordNet is a lexical resource for opinion mining. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity [30]. All words are grouped into set of synonyms called *synset*. In a SentiWordNet 3.0, every English word has a POS tag, an id number, positive score, negative score, synonyms and corresponding example.

Example:

Text: "Sound quality is awesome."

Here, "Sound Quality" is feature and awesome is opinion. The opinion value can be extracted from SentiWordNet. Opinion Score is-

Table 3: SentiWordNet structure for a word "awesome" [31]

A	01282510	0.875	0	awing#1 awful#6 awesome#1 awe-inspiring#1amazing#2	"the awesome complexity of the universe"

Here from the table 3,"A" stands for adjective, 01282510 is the id number, 0.875 is positive score, 0 is negative score, "awing", "awful", "amazing" are the synonyms of the word "awesome" and "the awesome complexity of the universe" is the use or example of that word "awesome".

So, from the table 3, the score of the opinion "awesome" can be extracted and the review is classified as positive.

Let, *Score$_i$* denotes score of *i*th opinion (o_i) and assign *Score$_i$*=0.0. Opinion can be positive, negative or neutral. Let, *P* denotes positive score, *N* denotes negative score and *O* denotes objective score.

$(P, N, O) = \oint (o_i)$

$$Score_i = \begin{cases} Score_i + P \; if \; P > N \\ Score_i - N \; if \; N > p \end{cases}$$

Total Scores of n reviews will be-

$$tScore = \sum_{i=1}^{n} Score_i$$

After identify the total score of every feature, we rank the features of every app. Final score of a feature will be-

$$Score = \frac{tScore}{N} \times 100$$

Where N is number of reviews contains the opinion of that feature.

Identify the Top Features of the App:

Using the above formula we identify the score of each feature and rank those features according to score.

Example: *Feature ranking for "Messenger" app:*

Table 4: Feature ranking for messenger app

Feature	Score
Voice Quality	137.5
Fun Sticker	87.5
Free Calling	41.665
Video Quality	3.895
Sound Quality	-25
Free Messages	-158.333

For messenger app, we take six features and the features are free calling, free messages, video quality, voice quality, sound quality, fun sticker. Among the features, voice quality is best for the app.

Identify the Best Apps for each Feature:

After identifying the scores of each feature of some app, the top apps of each feature are identified.

Example:

Best apps for the feature "Free Calling":

Table 5: Best apps for the feature free calling

App	Score
WeChat	156.25
Imo	100
Line	54.167
Messenger	41.667
Whatsapp	40.667
Viber	-5.556
Skype	-15.909

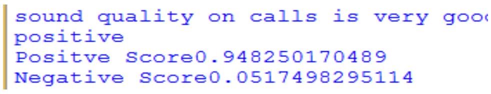
From the ranking, WeChat and Imo is the top app for calling. Line, Messenger and Whatsapp have also good score from the user's view.

Opinion Classification using Naïve Bayes:

Naïve Bayes classifier is trained by giving training set that was created. Once the classifier has been trained, the polarity of any sentence can be identified. The aim of this work is to classifying the reviews as positive or negative using Naïve Bayes.

A train dataset is used which consists of 470 data where 178 data is positive and 292 data is negative. After the classifier has been trained, a provided text can be classified.

Example:



```
sound quality on calls is very goo
positive
Positve Score0.948250170489
Negative Score0.0517498295114
```

Figure 4: the sentence classified as a positive sentence as positive score is greater than negative score

If positive score is maximum then the sentence is positive and if negative score is maximum then the sentence is negative.

Let, *Score$_i$* denotes score of *i*th opinion (o_i) and assign *Score$_i$* = 0.0

Opinion can be positive, negative. Let, *P* denotes positive score, *N* denotes negative score and *O* denotes objective score.

$(P, N, O) = \oint (o_i)$

$$\mathrm{Score_i} = \begin{cases} \mathrm{Score_i + P \; if \; P > N} \\ \mathrm{Score_i - N \; if \; N > p} \end{cases}$$

Total Scores of n reviews will be-

$$\mathrm{tScore} = \sum_{\mathrm{i=1}}^{\mathrm{n}} \mathrm{Score_i}$$

After identify the total score of every feature, the features of every app are ranked. Final score of a feature will be-

$$Score = \frac{tScore}{N} \times 100$$

Where N is number of reviews contains the opinion of that feature.

Identify the Top Features of the App

Example:

Feature ranking for "Messenger" app:

Table 6: Feature ranking for messenger app

Feature	Score
Fun Sticker	69.07
Free Calling	-22.896
Free Messages	-38.575
Voice Quality	-83.171
Video Quality	-84.445
Sound Quality	-99.148

For messenger app, six features are considered and the features are free calling, free messages, video quality, voice quality, sound quality, fun sticker. Messanger have nice stickers, free calling and messages have negative score but better than other features.

Identify the Best Apps for each Feature:

Example:

Best apps for the feature “Free Calling”:

Table 7: Best apps for the feature free calling

App	Score
Imo	77.712
WeChat	-0.512
Messenger	-22.896
Line	-33.585
Viber	-47.464
Whatsapp	-63.442
Skype	-84.559

From the ranking, Imo is the top app for calling. Wechat and whatsapp have negative score but better than other apps.

4. RESULT ANALYSIS

To benchmark our approach, we use available sets of reviews for seven apps (Viber, Skype, Imo, Messenger, Line, WeChat, Whatsapp) from [4]. The number of reviews for each of the apps, Viber, Skype, Imo, Messenger, Line, WeChat, Whatsapp are 41, 40, 40, 39, 40, 33, 40 respectively. For the classification of opinion, we use two methods naïve bayes and SentiWordNet 3.0. In this section, we will discuss the results of both methods and comparison between those results.

Figure 6 represents the comparison of total number of positive and negative reviews using SentiWordNet and Naïve Bayes.

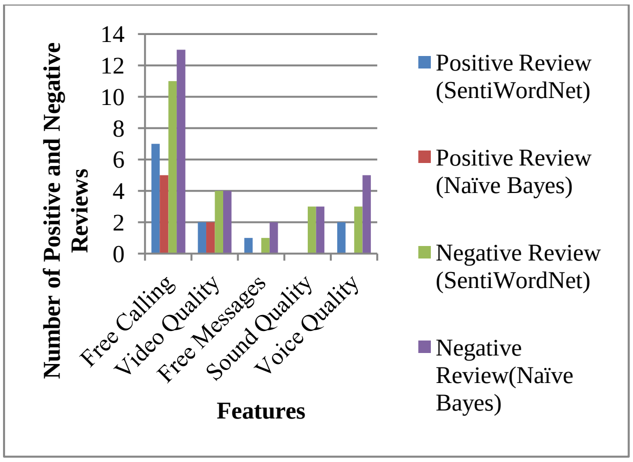


Figure 5: Comparison of total number of positive and negative reviews of “Viber” app using SentiWordNet and Naïve Bayes

From the figure, there are total 18 reviews for “free calling” feature and number of positive review is 7 and 5 using SentiWordNet and Naïve Bayes respectively. Number of negative review is 11 and 13 using SentiWordNet and Naïve Bayes respectively. For “video quality” feature positive review is same for both algorithms. So, some features have same result but some have different.

For getting more clear result, track initiation will be simulated firstly in order to test its distinguishing ability and fast ability in clutter environment. At last, set environment to test the final effect of track processing system as a whole.

Top Features of the App using SentiWordNet and Naïve Bayes:

Figure 7 compares the top features of the “Whatsapp” app using two different methods.

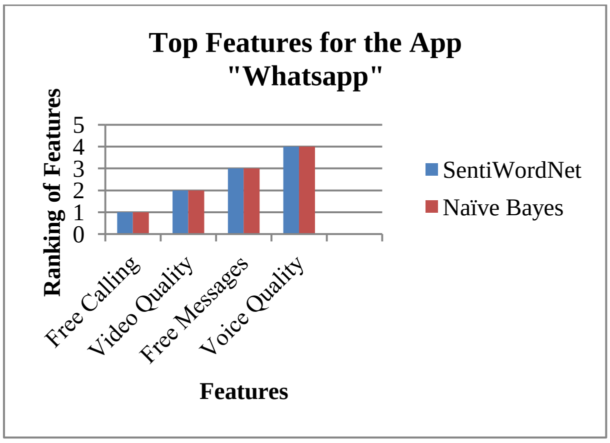


Figure 7: Comparison of top features of “Whatsapp” app using SentiWordNet and Naïve Bayes

From the figure, there are no dissimilarity for the top features of the app “Messenger”. For both methods, the result is same.

Best Apps of each Feature using SentiWordNet and Naïve Bayes:

Figure 8 compares the top apps for the feature “Free Calling” using two different methods, SentiWordNet and Naïve bayes.

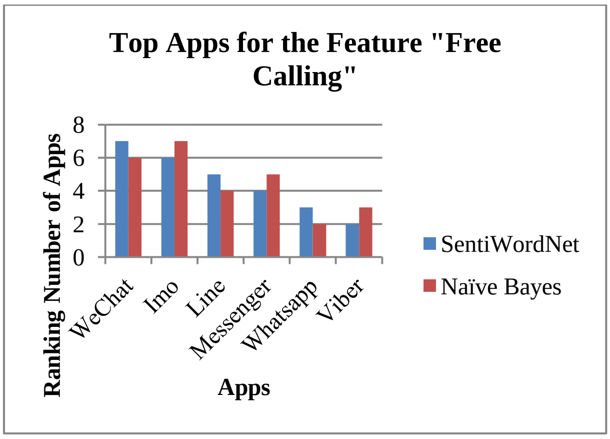


Figure 8: Comparison between top apps of the feature “Free Calling” using SentiWordNet and Naïve Bayes

From the figure, there are no huge differences between two ranking processes. So, top ranking for the feature “free calling” is quite similar.

Total Execution Time for SentiWordNet and Naïve Bayes

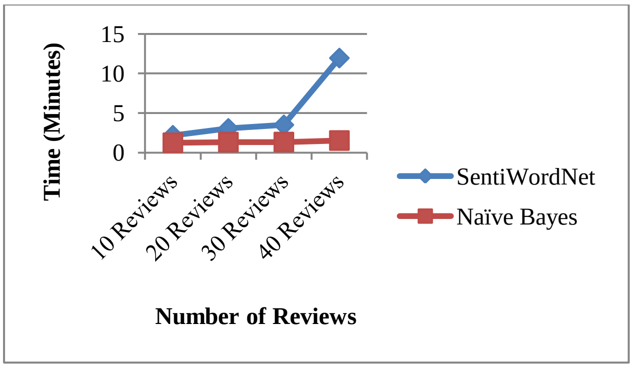


Figure 9: Comparison of execution time of two algorithms

From the chart, it is proved that Naïve Bayes is much faster (even including training time), than SentiWordNet.

For Naïve Bayes algorithm we use train dataset which contains total 470 data (178 data is classified as positive sentence and 292 data is classified as negative sentence). For accuracy measurement we give some test data and the result is shown in the table 6.

Table 6: Accuracy measurement for naïve bayes algorithm

Total Test Data	Positive Data	Negative Data	Accuracy (in Percentage)
50	24	26	72%
125	29	46	75%
100	35	65	79%
125	58	67	82.4%
150	78	72	84.6%
175	95	80	85.1%
200	109	91	84%

So, we find 84% accuracy which is enough to classify a sentence.

5. CONCLUSION

Due to web and social network, large amount of data are generated on Internet every day. This web data can be mined and useful knowledge information can be fetched through opinion mining process. This paper discussed opinion summarization based on features of app reviews.

Here, an approach is presented to summarize individual feature scores from app reviews. In order to provide greater precision than that in existing methods, the sentiment polarity of a feature is analyzed by considering the review's context-sensitive opinion words. Subsequently, feature scores are obtained by adjusting the original user scores by the sentiment polarities of the opinion words. Through experiments with actual reviews, that the method can accurately perform app review summarization at the app's feature-level. In addition, the method resulted in higher precision than existing methods.

In future, Opinion Mining can be carried out on a set of reviews and set of discovered feature expressions extracted from reviews. The state-of-art for current methods, useful for producing better summary based on feature based opinions as positive, negative or neutral is the Expectation Maximization algorithm based on Naive Bayes is the most efficient method. The efficiency of EM algorithm can be increased by augmenting it, to reassign classes of the labeled set.

6. REFERENCES

[1] Opinion Mining: Exploiting the Sentiment of the Crowd,Diana Maynard, Adam Funk, Kalina Bontcheva. University of Sheffield, UK

[2] Sun, Shiliang, Chen Luo, and Junyu Chen. "A review of natural language processing techniques for opinion mining systems." *Information Fusion* 36 (2017): 10-25.

[3] Kumar, KM Anil, et al. "Effective Approaches for Classification and Rating of Users Reviews." Proceedings of International Conference on Cognition and Recognition. Springer, Singapore, 2018.

[4] https://play.google.com/store

[5] Kim, S.M., Hovy, E.: Determining the sentiment of opinions. In: Proceedings of the Coling Conference. (2004)

[6] Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP). (2002)

[7] Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the ACL. (2004)

[8] Nasukawa, T., Yi, J.: Sentiment analysis: Capturing favorability using natural language processing. In: The Second International Conferences on Knowledge Capture. (2003)

[9] J. Yi, T. Nasukawa, R.B., Niblack, W.: Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In: 3rd IEEE Conf. on Data Mining (ICDM'03). (2003).

[10] L. Zhuang, F. Jing and X.-Y. Zhu, "Movie review mining and summarization," in Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM '06), ACM, New York, NY, USA, 2006.

[11] H. Tang, S. Tan and X. Cheng, "A survey on sentiment detection of reviews," Expert Systems with Applications, vol. 36, no. 7, pp. 10760-10773, 2009.

[12] Bo Pang, Lillian Lee, "Opinion Mining and Sentiment Analysis", Foundations and Trends in Information Retrieval Vol. 2, Nos. 1–2 (2008).

[13] MikalaiTsytsarau, Themis Palpanas "Survey on mining subjective data on the web", Data Mining Knowledge Discovery, Springer 2012, pp.478-514.

[14] Pang B, Lee L, Vaithyanathan S. "Thumbs up? Sentiment classification using machine learning techniques". Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2002.

[15] Dave K, Lawrence S, Pennock D. "Mining the peanut gallery: opinion extraction and semantic classification of product reviews". Proceedings of the 12th international conference on World Wide Web, ACM, New York, NY, USA, WWW'03.

[16] Bhatia, Surbhi, Manisha Sharma, and Komal Kumar Bhatia. "Sentiment Analysis and Mining of Opinions." *Internet of Things and Big Data Analytics*

Toward Next-Generation Intelligence. Springer, Cham, 2018.503-523.

[17] M. Dragoni, C. da Costa Pereira, A.G.B. Tettamanzi, S. Villata, Smack: an argumentation framework for opinion mining, in: S. Kambhampati (Ed.), Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016, IJCAI/AAAI Press, 2016, pp. 4242–4243

[18] Fellbaum, C. (1998). WordNet: An electronic lexical database. The MIT Press, 1998.

[19] AlexandraBalahur, MijailKabadjov, Josef Steinberger, Ralf Steinberger, Andrés Montoyo, "Challenges and solutions in the opinion summarization", Journal of Intelligent Information Systems Springer 2012.

[20] Siddhi Patni ,AvinashWadhe , "Review paper on sentiment analysis using web 2.0 by classification method ",IJARCS 2012.

[21] Chien-Liang Liu, Wen-Hoar Hsaio, Chia-Hoang Lee, Gen-Chi Lu, and Emery Jou" Movie Rating and Review Summarization in Mobile Environment".

[22] Changbo Wang, Zhao Xiao, Yuhua Liu, YanruXu, Aoying Zhou, and Kang Zhang,"SentiView: Sentiment Analysis and Visualization for Internet Popular

[23] LisetteGarcía-Moya, Henry Anaya-Sánchez, and Rafael Berlanga-Llavori, "Retrieving Product Features and Opinions from Customer Reviews", IEEE 2013.

[24] http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[25] http://www.nltk.org/

[26] http://sentiwordnet.isti.cnr.it/

[27] Opinion Mining For Text Classification AnandMahendran, Anjali Duraiswamy, Amulya Reddy, Clayton Gonsalves School of Computing Science & Engineering, VIT University, Vellore, Tamilnadu, India.

[28] https://technowiki.wordpress.com/2011/08/28/bayesian-opinion-mining/

[29] https://play.google.com/store/apps/details?id=com.viber.voip

[30] http://sentiwordnet.isti.cnr.it/

[31] http://sentiwordnet.isti.cnr.it/search.php?q=awesome