Analysis of Online Reviews of Emergency apps using NLP and Technology Acceptance Models

Project on GitHub : https://github.com/kalpanibhagya/EmergencyAppsReviews

Sasini Mahadura, Kalpani Kammangoda Mudalige, Nazanin Nakhaie Ahooie, and Rishikesh Kesari

Department of Computer Science and Engineering, University of Oulu

***Abstract*—Application reviews are great resources for getting feedback on applications, finding bugs, or seeing what features a platform needs. Many studies suggested that we can extract and analyse these data, using Natural Language Processing techniques and Technology Acceptance Models. In this project, we use the First Aid and Emergency Plus application and extract the reviews from the two application stores to find information on the performance and user sentiments towards these applications. Using Latent Dirichlet Allocation, Machine Learning and Technology acceptance model techniques, our results shows that the different versions of these applications for google play and apple stores have different adaption rates among users. However, in general, users found these applications handy and useful.**

***Index Terms*—Sentiment Analysis, Latent Dirichlet Allocation, Technology Acceptance Model**

I. Introduction

Application stores are platforms that allow users to download other applications and rate them or leave reviews. These platforms include Google Play for Android and the Apple Store. Studies have proven that user reviews of applications are great sources of determining bugs in the applications or feature requests [1]. But these information are often unorganized and occasionally incomplete. Therefore, it is essential to find solutions to this problem so that we can classify the reviews, extract data, and assist with the maintenance and development of software and platforms.

In this project, we went through the reviews of the two emergency applications, called First Aid: American Red Cross [2] and Emergency Plus [3]. These two applications are available at both application stores mentioned above and are developed by American Red Cross and National Triple Zero Awareness Work Group respectively. With more than a few million downloads over application stores, First Aid contains videos, step by step advice and contents that are handy in case of an emergency and Emergency Plus allows users to call emergency services and let them know of their location and situation.

The purpose of this study is to analysis the reviews from these applications and extract meaningful data, regarding the application features and performance and acceptance rate among users. We have extracted the reviews with their meta data such as users' id, ratings, review date , version data etc. from both application stores. As the reviews were not all in English, they needed translation. Several preprocessing techniques were used before analyze tasks.

Moving forward, regardless of reviews having star ratings, sentiment analysis was performed to classify them as positive, negative, or neutral. Sentiment analysis, often known as opinion mining, is a natural language processing method for identifying the passivity, negativity, or neutrality of data. In addition, user ratings and the results of the sentiment analysis over time have been observed to recognize user behavior towards the application after each release.

To learn more about user behavior, topic modeling was performed on review data using Latent Dirichlet Allocation (LDA) which is a technique in NLP that allows us to categorize texts into certain topics. From that most discussed topics by the user were identified and it gave an abstract idea on users opinion about these two apps. In our study, Random Forest Classifier is used as the machine learning technique which is trained using positive and negative sentiment review. From the trained model we were able to have a good overall picture about the feature requests that the user expected, pros and cons that users are concerned etc.

Technology Acceptance Model (TAM) is one of the most often used models for studies on the acceptance of new technology in Information Systems theory, which predicts whether users would accept or reject a new technology. The model provides a traditional view point of acceptance of technology from the user's aspects. Moreover, in our work, the TAM used to assess how end users have responded for the new releases of the applications. For instance, we have analyzed the level of satisfaction (S), attitude (A), users' behavior intention (BI), perceived ease of use (PE) and perceived usefulness (PE) over time. Also correlations between the TAM indicators were identified.

The remainder of the report is structured as follows. A brief review of related work is presented in Section 2. Details on the implementation are addressed in Section 3. Section 4 explains the results we obtained. Finally, in section 5, we conclude with a summary of the work and the results.

II. Related work

A. Sentiment Analysis

Sentiment analysis or opinion mining is a discipline that studies people's sentiments, attitudes, or emotions towards

certain entities [4], posited that One fundamental problem in sentiment analysis is the categorization of sentiment polarity. The issue is to categorize the text into one specific sentiment polarity in a given piece of written text, positive or negative (or neutral). According to them, The process of sentiment polarity categorization is two-way i.e., sentence-level categorization and review-level categorization. Furthermore, the objective of sentence-level categorization is to assign a sentence a positive or negative sentiment based on the sentiment it communicates. Ground truth tags that indicate whether a specific sentence is positive or negative are needed for the training data for this categorization technique. Finally, they performed experiments for sentence-level categorization and review-level categorization with promising outcomes.

B. Latent Dirichlet Allocation model and topic modeling

A statistical technique called topic modeling can be used to discover the underlying semantic structure of a large collection of document sets. Topic modeling is a technique that comes with a group of algorithms that reveal, discover, and annotate thematic structure in collection of documents [5]. Kherwa and Bansal explained a detailed discussion of the challenges of topic modeling, as well as its popularity of usage [6]. According to them, in the fields of machine learning and natural language processing, topic modeling utilizing latent Dirichlet allocation (LDA) is widely used to handle vast amounts of unstructured data and annotate these data with themes and topics. Therefore, we have also used the LDA method for topic modeling related to the review data of the two chosen emergency apps. Also in their article, the Latent Dirichlet Allocation model gives more coherent topics than other topic modeling techniques such as latent semantic analysis.

C. Technology Acceptance Model

In many contexts, there has been a wide utilization of TAM to identify the determinants of technology acceptance, more significantly to predict people's acceptance of information technology. TAM has been continuously studied and has expanded two major theories, TAM2 [7] and UTAUT [8].

TAM theory is strongly supported by studies to understand user acceptance of mobile library applications. The usefulness, interactivity, and ease of use have notable effects on the attitude and intention of the user to use the applications of the mobile library [9].

Within the field of innovation adoption for new technologies, the Technology Acceptance Model and the Theory of Planned Behavior have shown pioneering research efforts. It is suggested how these frameworks applied to the setting of newly developed information-age technology. After thorough analysis of the literature, it is concluded that both frameworks are extensively utilized, suitable to a variety of developing technologies, and still helpful in the field of innovation adoption research. According to the meta-analysis, the summary of 15 years of studies on TAM revealed a high correlation for the 'field setting' between Privacy Understanding (PU), Perceived ease of use PEOU, and the intention to use various technologies [10].

There is a lack of genuine measurement scales that are reliable for predicting the user acceptance of the computer. The majority of practical subjective measurements are unfounded, as well as how they relate to the system usage is obscure. This is where two behavioral factor comes into play. In the previous study, two new scales were created and validated for two distinct variables—perceived usefulness and perceived ease of use—that are hypothesized to be the primary predictors of user acceptance. In the case of perceived ease of use, an application which is perceived to be easier to use than another is more likely to be accepted by users, whereas a system high in perceived usefulness, in turn, is one for which a user believes in the existence of a positive use-performance relationship [11].

III. Methodology

A. Data Extraction

Data gathering is a significant step to be followed up in natural language processes to obtain better results. The accuracy of the final output depends on the quality and accuracy of the data set. As the first step, the required data were collected from the Google Play store and the Apple store for both apps, Emergency Plus and Red Cross First aid. For retrieving data from the google play store, the python package, Google-Play-Scraper was used while App-Store-Scraper was used to retrieve reviews from the apple store. The next step was to translate the reviews collected into English. Though there are several python packages for translations, the limitations of these packages made it a bit difficult to handle the translations with this large data set. During the investigation process to find a suitable Python or Python library or API for our scenario, the behavior of Google Trans, Google Translator API, and Deep Translator was observed. Using the Google Translator API requires special authentication. When using the google-trans library, the request limit per second and the length of the content to be translated were the limitations faced. Although the content could be minimized by splitting, it resulted in exceeding the maximum request limit. Hence the selection was the deep translator. As there is also a character limit per request, the data set was prepared by splitting it into several chunks and concatenating reviews by passing through a simple algorithm that ensures the optimal number of requests to the translator. This logic was written generically in a way that it could be used for some different content than our data set.

B. Sentiment Analysis using VADER

Sentiment analysis allows product owners to understand the sentimental value in customer reviews and find out where an issue or positive experience lies in the product or the process. By executing a sentiment analysis on reviews, we can find out what really are the customer insights about the chosen emergency applications.

To perform the sentiment analysis of reviews, Valence Aware Dictionary and Sentiment Reasoner (VADER) [12]

was used. It is a fully open-source, lexicon-based, and rule-based sentiment analysis tool. The polarity_scores method of the SentimentIntensityAnalyzer object in VADER gives a sentiment dictionary. It calculates positive, negative, neutral, and a compound scores for a given text. In this work, the compound value was used to recognize the sentiment value (positive : 1, negative: -1,neutral: 0) as below:

Listing 1. Sentiment Classification

```
for review in reviews:
    vs = analyzer.polarity_scores(review)

    if(vs["compound"] >= 0.05):
        sentiments.append(1)
    elif(vs["compound"] > -0.05):
        sentiments.append(0)
    elif(vs["compound"] <= -0.05):
        sentiments.append(-1)
```

The results were then used to visualize the changes in sen timent value with the new releases. This helps to understan whether the application delivered what the end user expecte over time. For this representation, the most frequent sentimer value for each release should be acquired. Therefore, releas periods have defined. For each release period, the period start on the release data and the end date is the day before the nex release date. An example of generated release periods for th Red Cross application is given in Table I.

TABLE I
RED CROSS APPLICATION RELEASE PERIOD DETAILS

Release Version	Release period start date	Release period end date	Sentiment
2.12.0	2021-11-26	-	-1
2.11.2	2021-08-25	2021-11-26	0
2.11.1	2021-03-08	2021-08-25	0
2.11.0	2020-10-26	2021-03-08	0
2.10.0	2020-07-23	2020-10-26	1
2.9.0	2020-05-26	2020-07-23	-1
2.8.2	2020-04-06	2020-05-26	-1
2.8.1	2020-03-16	2020-04-06	0
2.8.0	2019-11-18	2020-03-16	-1
2.7.3	2019-09-05	2019-11-18	0
2.7.2	2019-07-29	2019-09-05	-1
2.7.1	2019-07-22	2019-07-29	1
2.7.0	2019-07-16	2019-07-22	0
2.6.2	2019-04-30	2019-07-16	1
2.6.1	2019-03-04	2019-04-30	0
2.6.0	2018-11-07	2019-03-04	1
2.5.1	2018-09-04	2018-11-07	1
2.5.0	2018-03-28	2018-09-04	1
2.4.2	2017-03-13	2018-03-28	1
2.4.1	2017-02-08	2017-03-13	1
2.4	2017-01-19	2017-02-08	1
2.3.1	2016-01-29	2017-01-19	1
2.2	2016-01-28	2016-01-29	0
2.1.1	2015-04-14	2016-01-28	1
2.1	2014-11-25	2015-04-14	1

After obtaining the release period and its sentiment value, we plotted diagrams to visualize how sentiment has changed over time for each application.

C. LDA topic modeling

Latent Dirichlet Allocation (LDA) is one of the popular topic modeling mechanisms widely used in NLP. The basis of the LDA is based on the assumption that a document is a combination of topics, whereas these topics are a mixture of words. As the first step in LDA topic modeling, ten topics were generated for each app separately. We have created two sets of ten topics for each app using two corpora generated with term frequency and term document frequency of our data set. Topic modeling was done using LDA modeling. The Python packages Genism, NLTK, Spacy, and Re were required for the implementation.

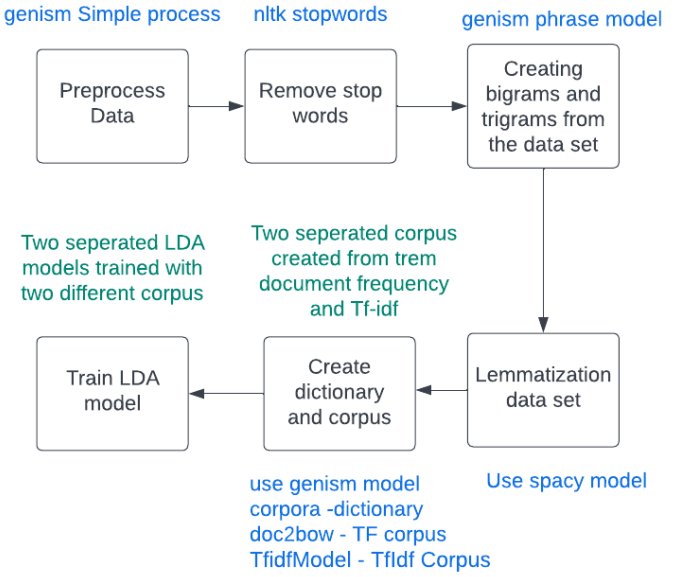

Fig. 1. Process of training LDA topic model

As shown in Fig.1, the generation of the topic was carried out through a stepped process. The first step was to preprocess the data to prepare them in a way that the LDA model can consume. So, the messiness inside the review data was eliminated through this step. During this step, all sentences were tokenized into words, all the characters were converted to lowercase, special characters, extra and trailing spaces were eliminated, and accent characters were replaced with their alternative characters. Although a separate implementation could be done for every task, it was very efficient, accurate, and easy to use the inbuilt mechanism of Simple Preprocess inside the Genism library. The stop words in our data set were removed by using the stop word from NLTK. A data set can contain bigrams and trigrams. Handling the bigrams and the trigrams reasonably within the data set is a must.

Hence, bigram and trigram models were created using the Genism Phrases model, in which the minimum occurrence of the word to be considered as a bigram was set as 5, and the threshold was given as 100. The result from the bigram model was passed to the trigram model to create trigrams if any exist in the data set. At the end of this stage, a set of documents was obtained. Each document consisted of a set of individual words, and some might contain bigrams and trigrams as well. Then the words in this document need to be lemmatized to keep only nouns, adjectives, adverbs, and nouns. Spacy was the Python library used for the lemmatization process. There are two main inputs that the LDA model requires. They are the dictionary and the corpus. Here, the dictionary is nothing but the mapping between words and their integer ids. To create the dictionary, the Genism Corpora model was used. The corpus for the LDA model was created in two ways using the Term Document Frequency and Term Frequency-Inverse Document Frequency. To create a corpus with term-document frequency, the doc2bow method in Genism was used. The created corpus was a mapping of the word id and the word frequency. The corpus with TF-IDF was created using TfidfModel in Genism models, which was a mapping of word id and corresponding TF-IDF. Then the LDA model was trained using a corpus and a dictionary while providing the number of topics required as parameters. Two models were generated separately for the corpus with term document frequency and the corpus with TF-IDF. And the results obtained from each model were evaluated with coherence score performance metrics. For the evaluation cv coherence score, the Umass coherence score, and graphical representations of generated topics using the pyLadVis library were used. As an additional step, the coherence score was plotted against the number of topics to find out the optimal number of topics that can be used to have a better model with a good coherence score.

D. Machine Learning model for Sentiment Analysis

In this project, the main objective of training a machine learning model for sentiment analysis was to identify the most important words or n-gram elements that impacted the classification of positive and negative classes. The random forest classification method was used with the review data set to extract the important features(words). The Python Sklearn learn library was used for the implementation. The review data and the sentiment result related to each review in the data set were used to train the random forest classifier using the ensemble model in Sklearn. As the first step, reviews from our data set were preprocessed to tokenize the sentences, remove stop words and special characters, and convert words into lower and lemmatize. Here we have tried preprocessing with different libraries than what we used in LDA topic modeling to have some experience with other available libraries. In our data set, each review has an associated sentiment that can be positive, negative, or neutral. However, for training the model, only positive and negative sentiments were used as labels. Review data and sentiment data were divided into two sets, called training data sets and testing, to train the model. Four data sets are created as a result, including the training and testing set for each. Before the model training, text transformation and representation of the test data were done using TfidfVectorizer in the Sklearn library. Ngram(2,3) and ngram(3,4) were created using a vectorizer. In ngram (2,3), the training data were vectorized in a way that holds two to three words per element. Similarly, in ngram (3,4), the training data was vectorized in a way that holds three to four words per element. Then the Random Forest Classifier was trained with this vectorized review data set and the sentiment data training set(training labels). In this case, two models were trained separately for the two ngrams created. Afterward, the predefined function 'feature.importances_' within the Random Forest model was used to retrieve the most important words or the ngram elements that impacted the classification to be positive or negative. Retrieving feature importance(most important words, ngram elements) is carried out separately for the two trained models with different ngrams.

E. Technology Acceptance model

In this report the next focus is given using the technology acceptance models (TAM) to analyze how people respond to new releases of the chosen emergency applications. For example, we try to measure the level of satisfaction, perceived ease of use, perceived usefulness, intention of behavior, and attitudes towards apps, and also we try to understand how these indicators change over time.

As the first step, considering the context of emergency mobile apps, keywords for the main indicators of the TAM model generated; perceived_usefulness: 19, perceived_easeofuse: 28, satisfaction: 26, attitude: 27, behavioural_intention: 18.

This small word set is not enough for the classification task because it will not represent all other related key words in the reviews. Therefore, the data set can be augmented by adding synonyms, hyponyms, and hypernyms of the chosen words.

This ended up increasing the data set with a considerable amount of words; perceived_usefulness: 232, perceived_easeofuse: 179, satisfaction: 232, attitude: 286, behavioural_intention: 72.

Next the categorizations of reviews according to the indicators was done. Before categorization, preprocessing of keywords and reviews is needed. For key words, conversion to the lower case, lemmatization, and part-of-speech(POS) tagging were done and, for reviews, only conversion to the lower case. Tokenization, lemmatization, and POS tagging were done.

Subsequently, common words were found between each set of review words with the indicators. The indicator which the review words contain more common words is assigned as the main indicator of that particular review. This process was done to all of the reviews in all four apps. After the assignment of the indicator, the data were split into separate data frames according to the indicator. Main reason behind this is then it will be possible to check the changes indicators over the time. Like in sentiment analysis using VADER, data collected were plotted against the release period.

TABLE II
TAM INDICATORS AND THEIR KEY WORDS

Indicator	List of Keywords
Perceived usefulness	timely, beneficial, quick, fast, efficient, effective, productive, systematic, streamlined, structured, organized, orderly, chaotic, unorganized , confusing, frustrating, problematic, broken, limited
Perceived Easy of use	convenient, manageable, simple, handy, practical, uncomplicated, useful, reliable, foolproof, flawless, disorderly, perfect, messy, infallible, sleek, smooth, poor, complicated, tricky, tangle, unable, crashed, complex, easy, difficult, comprehensive, incomplete, Invalid
Satisfaction	satisfied, useful, fulfill, gratify, meet, beneficial, happy, appeasement, unsatisfied, bad, meaningless, useless, shame, invaluable, missing, worth, Incredible, recommend, fantastic, disappointing, hopeless, best, lifesaver, promising, wonderful, outdated
Attitude	great, amazing, love, cool, helpful, disappointed, useless, dangerous, disorganize, terrible, irresponsible, bad, wrong, awesome, impressed, worse, brilliant, smart, fine, joke, necessary, compulsory, super, terrific, excellent, nice, pitiful
Behavioural Intention	informative, productive, liable, accurate, manageable, inaccurate, user-friendly, disorganize, untrustworthy, uninstructive, precise, responsive, essential, detailed, reliable, effective, unresponsive, powerful

IV. RESULTS AND DISCUSSIONS

A. Data Extraction

Each collected data set for both apps is large and consists of nearly 1000 reviews for the Emergency plus app and 3500+ reviews for the Red Cross First Aid app. From the extracted data, data related to review id, user name, review, rating, and review date were stored in a CSV.

B. Sentiment Analysis using VADER

Emergency Plus apps from both Google and Apple showed sentiment changes as follows in sentiment analysis using VADER.

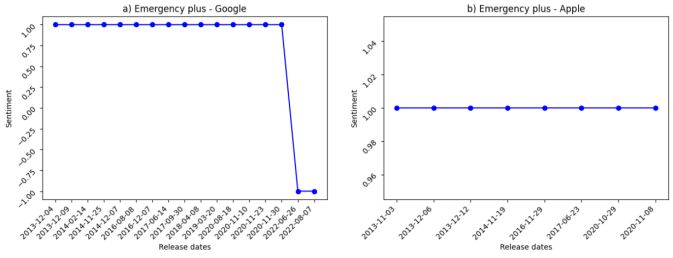

Fig. 2. Sentiment vs release period - Emergency Plus

As we can see in Fig.2 the sentiment value of the Emergency Plus Google has reduced over the time. Last 2 new releases shows negative sentiment value. Emergency Plus Apple application shows continuous positive sentiment over time.

Since the rating is given by the user himself, we can say that it is his true sentiment towards application. Therefore, we can use rating visualizations and the sentiment visualization to analyze the sentiment over time. As previously did for sentiment values, the average rating value for each release period should also be calculated for this purpose.

Fig.3 shows the average rating values for each Emergency Plus application with respect to their release periods.

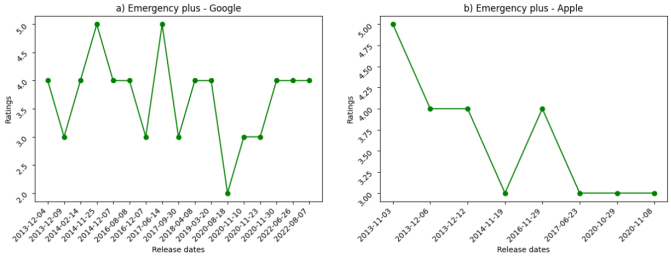

Fig. 3. Rating vs release period - Emergency Plus

Rating values has changed over the time for both Apple and Google applications. Recent rating for Google app is 4 which can be taken as positive. But in sentiment analysis we got a negative sentiment value. There can be many reasons to that. One is that the amount of data in the last release period is small; therefore, the real rating average or sentiment average was not observed.

The rating values for the Apple application have reduced with time and become neutral (by assuming rating 3 is equivalent to neural sentiment). Since sentiment analysis showed positive sentiment for the whole period, we can observe VADER sentiment analyzer has not performed well for these two situations.

Nevertheless, sentiment values of the Red Cross application with respect to the release period is given in the Fig.14.

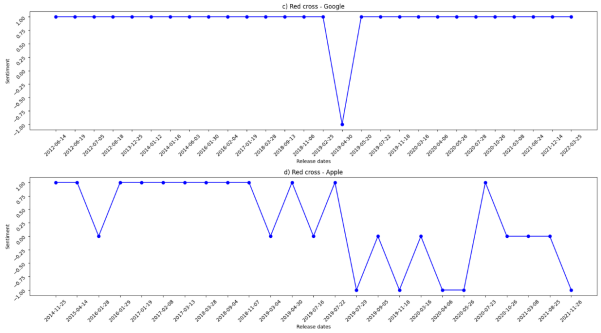

Fig. 4. Sentiment vs release period - Red Cross

The sentiment of the Red Cross Google application was almost positive throughout the time, in addition to one decrease in April 2019. Meantime, sentiment value of Red Cross Apple application has fluctuated over the time. In the most recent release the sentiment value has become negative.

The Google application ratings fluctuated only between 4 and 5. Therefore we can assume that the sentiment value was

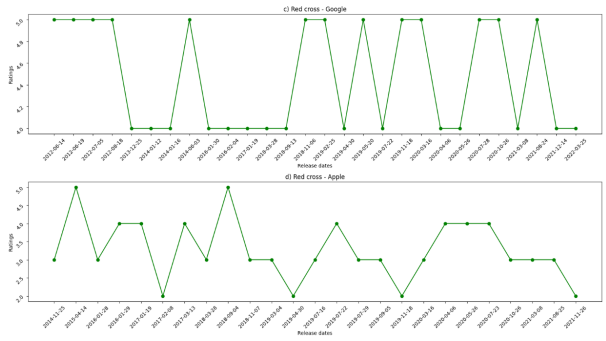

Fig. 5. Rating vs release period - Red Cross

almost positive throughout the time as it has shown in the relevant sentiment graph. The ratings of the Apple application have also fluctuated over time like its sentiment value. Both graphs show somewhat similar movements. Therefore, in this scenario, the sentiment visualization has performed well for the Red Cross applications. Clearly, the considerable amount of data is the reason for the correct representation. If Emergency Plus applications contained more results, we could get more matching and meaningful visualizations such as the latter.

C. LDA topic modeling

LDA modeling was carried out with two different corpora(tf, tf-idf) for each application, as mentioned above in the methodology section. The ten topics shown in Fig.6 were generated by training the models for two corpora of each app as the initial step.

```
 10 Topics obtained from the reviews of Emergency Plus.csv - Tf Corpus
0::['open', 'show', 'crash', 'accurate', 'time', 'handy', 'lot', 'release', 'standard', 'km']
1::['app', 'emergency', 'location', 'phone', 'great', 'need', 'gps', 'address', 'good', 'service']
2::['try', 'iphone', 'much', 'fire', 'actually', 'application', 'dial', 'put', 'today', 'cause']
3::['useful', 'easy', 'feature', 'mobile', 'current', 'helpful', 'check', 'real', 'install', 'developer']
4::['way', 'correct', 'away', 'address', 'live', 'however', 'lose', 'valuable', 'dangerous', 'wrong']
5::['excellent', 'life_save', 'spot', 'mean', 'serious', 'blue', 'available', 'function', 'guy', 'hope']
6::['well', 'recommend', 'still', 'tell', 'able', 'button', 'send', 'never', 'people', 'really']
7::['work', 'update', 'turn', 'long', 'fix', 'old', 'house', 'new', 'case', 'poison']
8::['download', 'place', 'brilliant', 'list', 'rural', 'especially', 'nice', 'big', 'bad', 'know']
9::['idea', 'home', 'also', 'great', 'see', 'screen', 'word', 'know', 'add', 'map']

 10 Topics obtained from the reviews of Emergency Plus.csv - TfIdf Corpus
0::['handy', 'know', 'even', 'rural', 'freeze', 'standard', 'exact', 'concern', 'see', 'km']
1::['love', 'map', 'phone', 'operator', 'second', 'smart', 'still', 'help', 'case', 'system']
2::['address', 'location', 'find', 'life', 'save', 'coordinate', 'accurate', 'wrong', 'street', 'work']
3::['good', 'emergency', 'useful', 'gps', 'old', 'feature', 'people', 'helpful', 'easy', 'locate']
4::['otherwise', 'star', 'install', 'possibly', 'access', 'function', 'quite', 'understand', 'ready', 'google_map']
5::['spot', 'next', 'much', 'kid', 'probably', 'lot', 'local', 'operator', 'tip', 'hope']
6::['number', 'location', 'app', 'well', 'include', 'read', 'need', 'call', 'never', 'great']
7::['problem', 'life_save', 'report', 'cause', 'take', 'apple_watch', 'different', 'hope_never', 'nice', 'dial']
8::['recommend', 'awesome', 'fantastic', 'fire', 'dangerous', 'fact', 'state', 'highly', 'identify', 'thank']
9::['great', 'idea', 'app', 'excellent', 'need', 'open', 'crash', 'time', 'phone', 'long']


 10 Topics obtained from the reviews of Red Cross First Aid.csv - Tf Corpus
0::['useful', 'awesome', 'phone', 'handy', 'teach', 'review', 'wonderful', 'feature', 'miss', 'wish']
1::['life', 'save', 'much', 'basic', 'answer', 'nice', 'cpr', 'bite', 'recommend', 'heat']
2::['love', 'helpful', 'iphone', 'time', 'case', 'update', 'download', 'keep', 'crash', 'step']
3::['see', 'give', 'call', 'thing', 'think', 'medical', 'find', 'way', 'person', 'add']
4::['really', 'video', 'never', 'want', 'job', 'happen', 'sure', 'enough', 'finger_tip', 'relate']
5::['first', 'aid', 'work', 'take', 'knowledge', 'right', 'red_cross', 'class', 'training', 'course']
6::['app', 'great', 'emergency', 'know', 'information', 'easy', 'thank', 'well', 'learn', 'informative']
7::['info', 'situation', 'section', 'reference', 'fix', 'star', 'application', 'practical', 'access', 'content']
8::['amazing', 'instruction', 'understand', 'issue', 'problem', 'today', 'arc', 'point', 'several', 'accurate']
9::['good', 'need', 'help', 'test', 'get', 'even', 'come', 'prepare', 'tell', 'ever']

 10 Topics obtained from the reviews of Red Cross First Aid.csv - TfIdf Corpus
0::['awesome', 'handy', 'case', 'really', 'help', 'teach', 'step', 'glad', 'always', 'app']
1::['basic', 'need', 'video', 'perfect', 'instruction', 'never', 'know', 'application', 'follow', 'important']
2::['helpful', 'download', 'work', 'crash', 'hand', 'time', 'prepared', 'update', 'search', 'keep']
3::['useful', 'get', 'iphone', 'app', 'go', 'wonderful', 'thing', 'answer', 'knowledge', 'great']
4::['life', 'save', 'much', 'medical', 'safety', 'phone', 'kid', 'question', 'hope', 'help']
5::['good', 'info', 'aid', 'learn', 'first', 'amazing', 'app', 'lot', 'great', 'information']
6::['great', 'app', 'easy', 'information', 'understand', 'useful', 'review', 'person', 'finger_tip', 'simple']
7::['nice', 'situation', 'ever', 'prepare', 'job', 'app', 'well', 'design', 'ipad', 'little']
8::['educational', 'come', 'emergency', 'make', 'way', 'life', 'fun', 'help', 'know', 'right']
9::['love', 'informative', 'test', 'excellent', 'thank', 'tool', 'app', 'great', 'red_cross', 'useful']
```

Fig. 6. 10 Topics generated for each app by different corpora

As an additional step, the performance of the topic model was observed using performance metrics such as the coherence score, perplexity, and using some graphical representations. The coherence score measures how interpretable the topics are to humans. It can be calculated either by the Cv score or the Umass. Score. Perplexity is the measure of how well a model predicts a sample. Generally, the generated model is said to be better at higher Cv coherence and lower the Umass score and perplexity.

TABLE III
PERFORMANCE RESULTS OF THE LDA TOPIC MODEL (WHEN 10 TOPICS ARE USED)

App Name	Corpus Type	Perplexity	Cv score	U mass score
Emergency plus	TF corpus	-6.97	0.47	-12.65
Emergency plus	TF-IDF corpus	-9.78	0.49	-12.79
Red cross First Aid	TF corpus	-7.51	0.41	-9.43
Red cross First Aid	TF-IDF corpus	-9.78	0.35	-7.85

In Tabel V, performance results (rounded to two decimals) obtained for each app are recorded. There is no exact way of deciding whether the coherence score obtained is good or not good. Because the score value calculated is dependent on the data set that is used to train the model. Therefore, there is a chance that the coherence score of 0.5 may be good in one case and not in another. But, increasing the score makes the model better. Usually, the coherence score increases with the number of topics that are used to train the model. Therefore, as an additional step in our analysis, an investigation was carried out to find the optimal number of topics that can be used for training to achieve a better topic model. To achieve it, the cv coherence score was plotted as a function of the number of topics to identify the optimal number of topics at which point the highest cv coherence was achieved. The plotting was done for the two apps separately with their two different corpora.

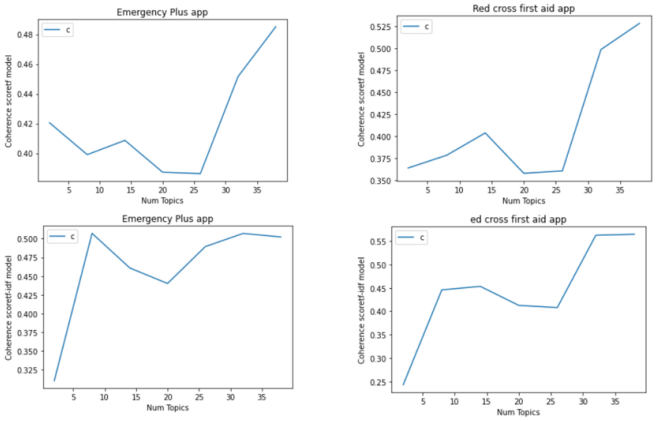

Fig. 7. Variation of cv coherence with a number of topics

The graphs in Fig.7 depict how the coherence score increases as the number of topics increases. A higher cv score

is recorded near the point 35 topics. But, if the model is trained for a much higher number of topics, it will cause repeating the same word in several topics. Therefore, the optimal number of topics to use for model training have to be decided depending on the research requirement. Accordingly, in our project, the rough optimal number of topics for each case identified by observing the drawn graph is presented in Tabel V.

TABLE IV
OPTIMAL NUMBER OF TOPICS FOR BETTER PERFORMANCE

App Name	Corpus Type	Optimal number of topics
Emergency plus	TF corpus	13
Emergency plus	TF-IDF corpus	8
Red cross First Aid	TF corpus	14
Red cross First Aid	TF-IDF corpus	8

After training the models again for the optimal number of topics, the graphical representation was used to decide the best model for each app.

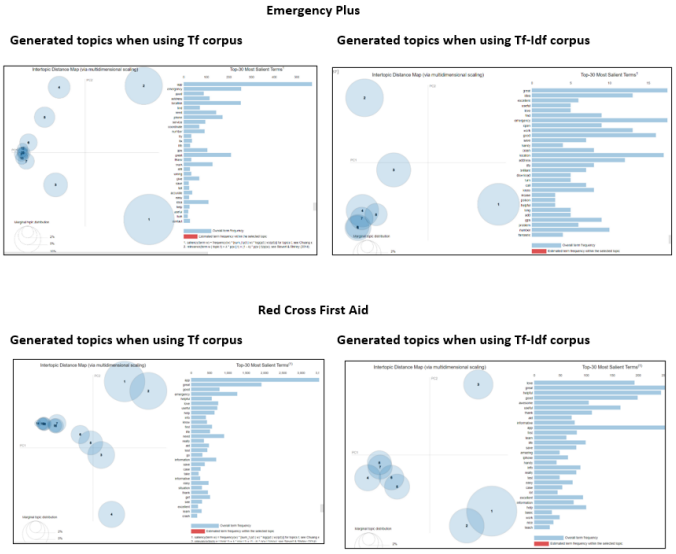

Fig. 8. Generated Topics for Emergency plus App

Fig.8 shows the graphical representations obtained using pyLadVis.The topics generated are represented by bubbles. The model is good if the bubbles are large and scattered over the quadrant with fewer overlappings. In our case topics generated from LDA models trained with the tf- idf corpus of each app is having bubbles that are larger, less overlapped, and scattered over the quadrant compared to the topics generated with the model in which the tf corpus is used. Therefore, the topics from the LDA models trained with the tf-idf corpus are chosen as the topics for each app. As a result, eight topics have been generated for each app as depicted in Fig.9.

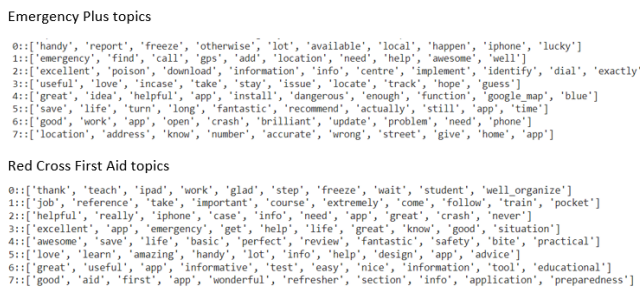

Fig. 9. Generated Topics with better model

D. Machine Learning model for Sentiment Analysis

The Random Forest Classification model was trained with 2963 review data and a set of labels (sentiment data). After training the classifiers for both ngram (2,3) and ngram (3,4), 50 of the most important words or ngram elements of the ngram are extracted for each ngram. The most important 50 ngram elements are graphically represented with their feature importance score in Fig.10. Each selected element in the ngram was classified into positive and negative classes. The results obtained are listed in the Tabel V.

According to the results obtained, we could find out that majority of the most important ngram elements were specified into the positive class. For the ngram(3,4) no single ngram elements classified into the negative class were found. But for the ngram(2,3) fewer ngram elements were classified into the negative class. This could be because our original data set contained fewer negative ratings compared to positive ratings. But, from the most important features that we extracted, some of the feature requests, pros, and cons of the app from the user's point of view can be identified. For example, there is an important ngram element 'gps coordinate' which was classified as negative, suggesting that there is an issue with 'gps cordinates' in the app that should be addressed as a feature request. By looking at the important ngrams classified as positive, we can conclude that apps are good, user-friendly, instructive, and helpful and users are satisfied with them by referring to the ngram elements such as 'app easy use', 'best app ever', 'lot useful information', 'step step instruction', 'thank red cross', 'thanks red cross'. But still, there are some conflicting most important ngrams such as 'hope never use' which sounds like negative but classified as positive. Therefore, we expanded our analysis to check how our trained models performed by creating confusion matrices and obtaining a classification report in terms of precision, FI score, and recall.

As demonstrated in Fig.11 both the classification models have correctly classified almost all actual true values. But it is performed worst in predicting actual false values. Therefore, almost all the false values are classified as true. From the observations on the training data set, we can say that this is because our training data set contains fewer negative reviews compared to positive reviews. Therefore our trained model is

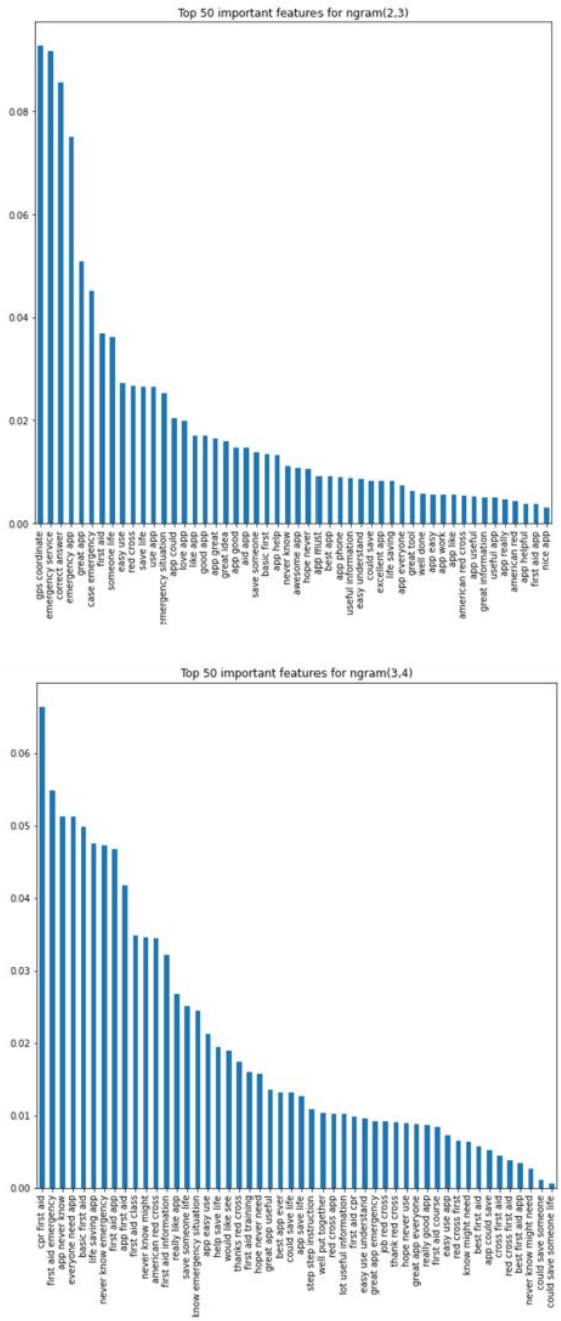

Fig. 10. Top 50 important features for ngrams

biased. However, looking at the confusion matrices and original values, we can conclude that the classification model for ngram(2,3) is slightly performing better than the classification model for ngram(3,4).

E. Technology Acceptance model

To analyze changes in the TAM indicators over time, we separated each application's data frame in to five data frames according to the type of the indicator. Then we visualized

TABLE V
SPECIFY IMPORTANT FEATURES TO CLASSES

Ngram	Positive elements	Negative elements
ngram(2,3)	american red, american red cross, app easy, app everyone, app good, app great, app help, app helpful, app like, app must, app phone, app really, app useful, app work, awesome app, basic first, best app, case emergency, could save, easy understand, easy use, emergency situation, excellent app, first aid, first aid app, good app, great app, great idea, great information, great tool, hope never, life saving, like app, love app, never know, nice app, red cross, save life, use app, useful app, useful information, well done	aid app, app could, correct answer, emergency app, emergency service, gps coordinate, save someone, someone life
ngram(3,4)	american red cross, app could save, app easy use, app first aid, app never know, app save life, basic first aid, best app ever, best first aid, best first aid app, could save life, could save someone, could save someone life, cpr first aid, cross first aid, easy use app, easy use understand, everyone need app, first aid app, first aid class, first aid course, first aid cpr, first aid emergency, first aid information, first aid training, great app emergency, great app everyone, great app useful, help save life, hope never need, hope never use, job red cross, know emergency situation, know might need, life saving app, lot useful information, never know emergency, never know might, never know might need, really good app, really like app, red cross app, red cross first, red cross first aid, save someone life, step step instruction, thank red cross, thanks red cross, well put together, would like see	No negative ngram elements found

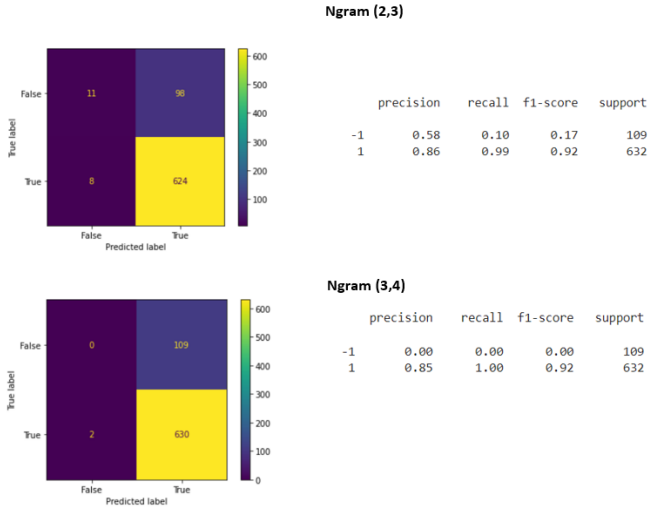

Fig. 11. Confusion matrices and classification report for trained models

the rating and sentiment values over time for each indicator. Interpretations of the results are given below.

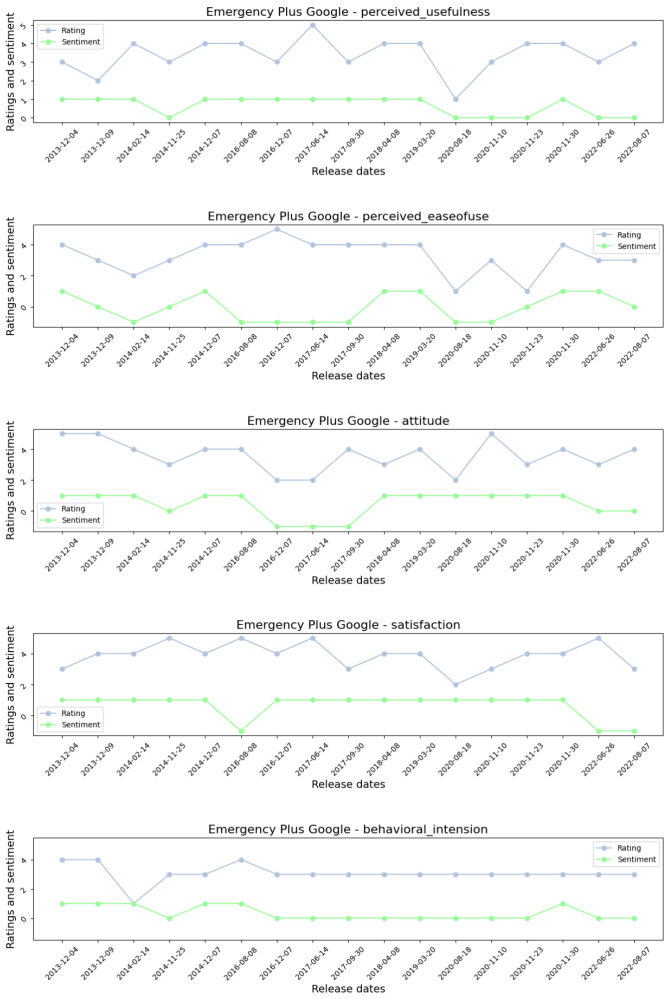

Fig. 12. Emergency Plus Google - Indicators

After looking into the perceived usefulness graph of the Emergency Plus Google application in Fig.12 it is clear that the sentiment value of the reviews that were fallen under perceived usefulness has fluctuated first and now it has become neural in the recent releases, although the rating has become 4. Therefore, we can assume that the end users have experienced a decrease in the usefulness of this particular application according to the sentiment. But it contradicts what rating interprets.

Users also think that they need more effort to use the new version of the application than the previous version according to both sentiments and ratings, but it was not a worse scenario. Users are neutral about the effort and their actual intention in using the application. Attitude and satisfaction with the application have decreased with time, depending on the sentiment value.

However, the smaller number of reviews in the Emergency Plus reviews data sets has affected the results, as the rating and sentiment graphs have slight mismatches.

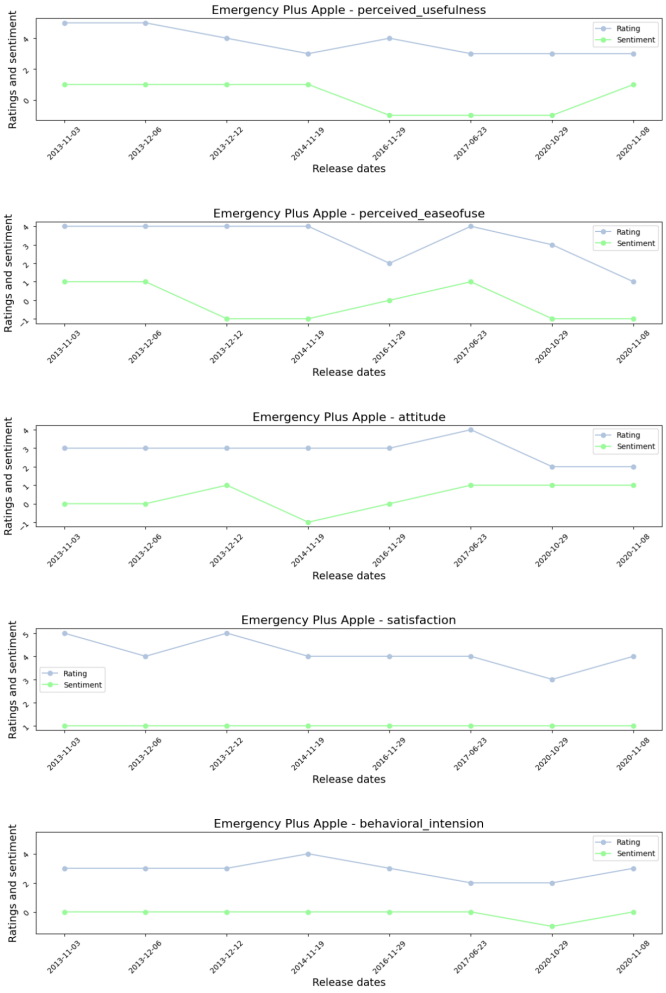

Fig. 13. Emergency Plus Apple - Indicators

In contrast to the Google version, apple version's (Fig.13) indicators showed positive changes over time apart from the data on ease of use. Red Cross Google application(Fig.14) has also performed better over time in all five indicators. Therefore, we can assume that the users think this application is useful and easy to use, and they have a positive attitude toward it. Since they are satisfied about the functionalities, their actual intention in using this application has become positive.

But the Red Cross Apple application (Fig.15) shows decrements in the sentiment and rating values in all the indicators. In easy-use and behavioral intention graphs although sentiment values have decreased, but they remain neutral in the most recent releases. For end users, the application is not useful anymore and they have a negative attitude about it since they are not satisfied with the outcome.

Therefore using TAM model analysis, we can observe that the recent Emergency Plus Apple and Red Cross Google releases have shown a good acceptance rate while the other two applications need more improvements in a way that they cater users' needs.

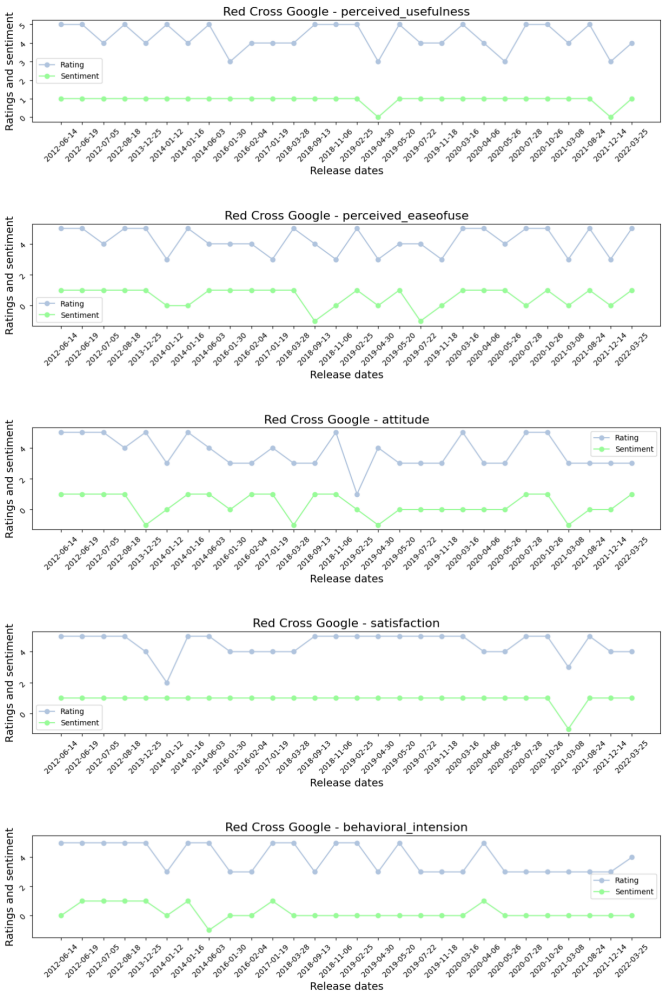

Fig. 14. Red Cross Google - Indicators

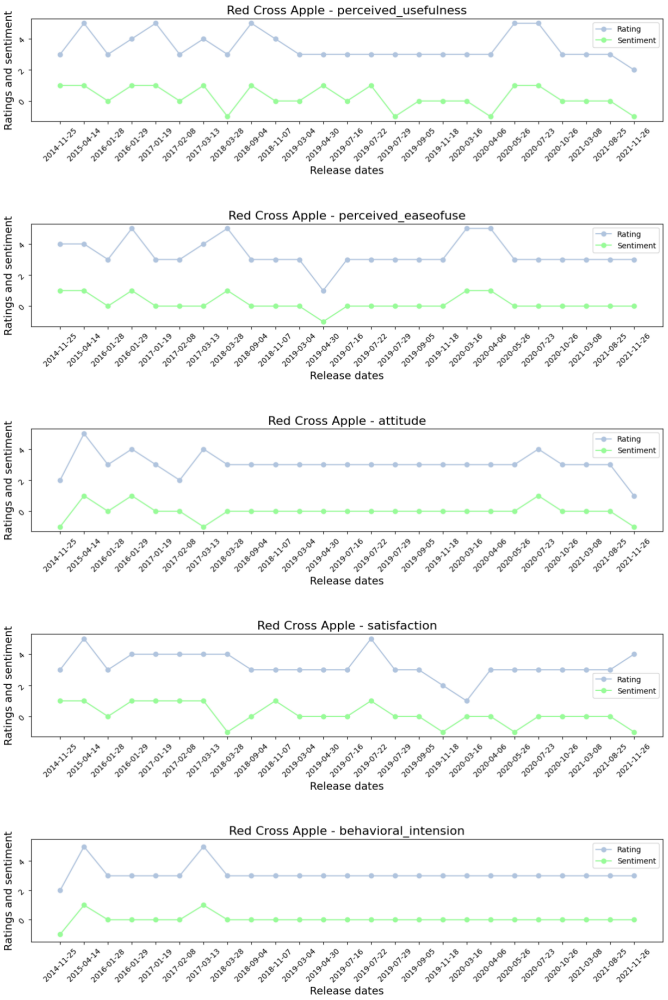

Fig. 15. Red Cross Apple - Indicators

Next, using the indicator key words shown in Table II, we have calculated the **Pearson correlation values** with its associated p-value values to determine which are the most strongly correlated indicators.

The Pearson correlation method is the most common method of measuring a linear correlation for numerical variables, although it can also be used with string variables. Correlation value is always between -1 and 1, where 0 is no correlation, 1 is total positive correlation, and 1 is total negative correlation. In positive correlations as x increases, y increases and in negative correlations as x increases, y decreases.

To find the correlation between the indicator key word lists, first the similarity values of all possible key-word pairs should be identified. For this FuzzyWuzzy python library has used and the library uses Levenshtein Distance to calculate the differences between couple of words. After calculating the similarities, the values were stored in separate text files as three-value tuples; word1, word2 and similarity. The word vector method **evaluate_word_pairs** is used to calculate the Pearson correlation and the p-value. Values are given up to 4 decimal points.

TABLE VI
PEARSON CORRELATION BETWEEN INDICATORS

Indicator1	Indicator2	Correlation Coefficient	P-value
PU	PE	-0.3593	0.4841
PU	A	-0.3253	0.3588
PU	S	0.1951	0.7110
PU	BI	1.0	1.0
PE	A	-0.1869	0.5046
PE	S	-0.1893	0.6256
PE	BI	-0.1200	0.9234
A	S	-0.2067	0.4597
A	BI	-0.0820	0.8955
S	BI	0.0558	0.9644

From the above results, we can only observe one perfect and strong correlation where correlation value and the pvalue is 1. It is between perceived usefulness and the intention of behaviour. Therefore we can conclude that when the usefulness of an application increases the users' actual intention to use the application also increases.

V. Conclusion

In this study, we analyzed reviews of two emergency applications, First Aid Red Cross and Emergency Plus from the Apple app store and Google play store. As the study's goal was to identify users' behavior toward the applications, several NLP approaches were employed to evaluate and extract useful information.

The sentiment analysis of the reviews was performed using VADER, and reviews were classified as positive(1), negative(-1) and neutral(0) using compound score. Afterwards, we compared our results with the ratings given by the users assuming that it is the true sentiment of user. Although the Emergency Plus apps results showed minor mismatches between the ratings and sentiments, due to the considerably larger amount of data, rating and sentiment results of the Red Cross First aid application visualized similar fluctuation over time. That helps us to correctly interpret users' sentiment over time.

During this study, we focused on retrieving most discussed topics by the user using LDA topic modeling. As an initial step, we generated 10 topics per app, and then the study was extended to find a better model with an optimal number of topics where the coherence score is high. From multiple results obtained, we concluded the best model by looking at graphs generated by pyLadvis.

Further, a Random Forest Classifier with Positive and Negative Sentiments was trained to find out the most important words or n-gram elements that impacted the classification for positive and negative classes. From the result obtained, we were able to identify some of the feature requests, pros, and cons of the app from the user's perspective. When observing how far this trained model is good, we recognized that it was good in predicting positive sentiments but not in predicting negative, ones since the training data set containing fewer negative sentiments compared to positive ones made the training model biased.

Using the basic knowledge on TAM model, we identified set of keywords that represent each TAM indicator. After separating the reviews according to the indicators, changes of the sentiment and rating values over time were observed separately. This visualization helps us to come up with conclusions regarding the user acceptance levels of the all four applications including both Apple and Google versions. Finally, Pearson correlation was used to identify the indicators which are correlated with each other. Correlation value between perceived usefulness and the behaviour intention showed as 1 which is a perfect correlation where intention in using the application increases when the usefulness of the application increases.

As a whole, our results show user behavior, sentiment toward emergency apps and highlight the most important topics regarding users' requests, demands and preferences in terms of emergency solutions or technology features.

References

[1] S. Panichella, et al. (2015) "How can I improve my app? classifying user reviews for software maintenance and Evolution," 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME) [Preprint]. Available at: https://doi.org/10.1109/icsm.2015.7332474. 1955.

[2] R. Cross, "Red Cross on Play Store,"[Online]. Available: https://play.google.com/store/apps/details?id=com.cube.arc.fahl=en

[3] "Emergency Plus on Google Play," National Triple Zero Awareness Work Group, [online].

[4] X. Fang, J. Zhan, "Sentiment analysis using product review data," Journal of Big Data 2,5, 2015. [Online]. Available: https://doi.org/10.1186/s40537-015-0015-2

[5] D. M. Blei, "Probabilistic topic models," Communications of the ACM, vol. 55(4), pp. 77-84, 2012.

[6] P. Kherwa, P. Bansal, "Topic Modeling: A Comprehensive Review ," EAI Endorsed Transactions on Scalable Information Systems, vol. 7, 2019. Available: https://eudl.eu/pdf/10.4108/eai.13-7-2018.159623

[7] V. Venkatesh, F.D. Davis, "A theoretical extension of the technology acceptance model: Four longitudinal field studies," Management science, vol. 46(2) pp. 186–204.

[8] V. Venkatesh, M. G. Morris, G. B. Davis, F. D. Davis, "User acceptance of information technology: Toward a unified view," MIS quarterly, pp. 425-478, 2003.

[9] Y. H. Yoon (2016), "User acceptance of mobile library applications in academic libraries: an application of the technology acceptance model," The Journal of Academic Librarianship, vol. 42(6), pp. 687-693.

[10] S. Koul, A. Eydgahi (2017), "A systematic review of technology adoption frameworks and their applications," Journal of technology management innovation 12, pp. 106-113,

[11] F. D. Davis(1989), "Perceived usefulness, perceived ease of use, and user acceptance of information technology," MIS quarterly, pp.319-340.

[12] Valence Aware Dictionary and Sentiment Reasoner (VADER). [Online]. Available: https://github.com/cjhutto/vaderSentiment