Journal of Optimization of Soft Computing (JOSC)

Vol. 2, Issue 1, pp: (37-43), June-2024

Journal homepage: https://sanad.iau.ir/journal/josc

Research paper

Comparison of Pre-Trained Models in Extractive Text Summarization of Mobile App User Reviews

Mehrdad Razavi Dehkordi [1], Hamid Rastegari [1,2*], Akbar Nabiollahi Najafabadi [1,2], Taghi Javdani Gandomani [3]

[1]. *Faculty of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran*
[2]. *Big Data Research Center, Najafabad Branch, Islamic Azad University, Najafabad, Iran.*
[3]. *Department of Computer Science, Shahrekord University, Shahrekord, Iran.*

Article Info

Article History:
Received: 2024/03/11
Revised: -
Accepted: 2024/05/26

DOI:

Keywords:
Mobile applications, Summarization of User Reviews, Google Play Store Analysis, Pre-trained Model

* *Corresponding Author's Email Address: rastegari@iaun.ac.ir*

Abstract

Since the inception of mobile apps, user feedback has been extremely valuable to app developers as it contains users' feelings, bugs, and new requirements. Due to the large volume of reviews, summarizing them is very difficult and error-prone. So far, many works have been done in the field of extractive summarization of users' reviews; However, in most researches, old methods of machine learning or natural language processing have been used, or if a model has been trained for summarizing using transformers, it has not been determined whether this model is useful for summarizing the reviews of mobile users. No? In other words, the model for summarizing texts has been presented in a general-purpose form, and no investigation has been carried out for its use in special purpose summarization. In this article, first, 1000 reviews were randomly selected from the Kaggle database of user reviews, and then given to 4 pre-trained models bart_large_cnn, bart_large_xsum, mT5_multilingual_XLSum, and Falcon'sAI_ Text_Summrization for summarization, and the criteria Rouge1, Rouge2 and RoungL were calculated separately for each of the models and finally it was found that the pre-trained Falcon's AI model with a score of 0.6464 in the rouge1 criterion, a score of 0.6140 in the rouge2 criterion and a score of 0.6346 in rougeL The best model for summarizing users' reviews is the Play Store.

1. Introduction

According to the official statistics of IDC website[1], about 300.3 million smartphones were produced by manufacturers by the fourth quarter of 2022, 75.8% of which were smartphones with the Android operating system. The Android operating system has its own store, called Google Play Store, which includes all its produced apps by developers[2] [1][2][3].

The apps available in the store are downloaded by many users and Google Play Store users could comment on the desired application. Studies have demonstrated the reviews made by users on apps

[1] IDC - Smartphone Market Share - Market Share. https://www.idc.com/promo/smartphone-marketshare

[2] Android Apps on Google Play. https://play.google.com/store/apps

contains important information, including bug reports, feature requests and user experience of working with the app[4][5][6]. Previous studies have revealed the reviews recorded by users could contribute to app development process and improve future app versions[7][8]. Moreover, reviews contain important information for app analysts, designers and developers[6][9][5]. Due to the high volume of reviews with important information, it was difficult to summarize them for handling by the development team manually, and as a result, having a tool to summarize and export the summarized reviews to the development team is very useful; Because having a tool or a model for summarizing reviews makes a summary list of requirements or bugs feed backed by users available to the development team and the development team does not waste much time reading each review and maintaining the software and providing More successful timely updates[10][11][12]. So far, many works have been done in summarizing the reviews of users of mobile applications, but in most of them, either natural language processing parameters have been used or machine learning methods have been used, which are old. Today, many pre-trained models are used. To summarize the reviews of users, using transformers are provided and all the models provided are general purpose and are not provided for a specific task. The purpose of this research is to compare and select the best pre-trained model in the extractive text summarization of user reviews of mobile applications in the Play Store. In this research, at first, 1000 reviews from the dataset including user reviews provided by Kaggle were randomly selected and given to 4 pre-trained models bart_large_cnn, bart_large_xsum, mT5_multilingual_XLSum and Falconsai. Finally, Rouge criteria have been measured for each model. The continuation of this article is organized as follows: in the second part, the concepts and works done in summarizing the reviews and their challenges are discussed, in the third part, the pre-trained models and their parameters are stated, in The fourth section discusses the dataset and evaluation criteria, the fifth section compares the 4 models presented in the summary, and finally, the sixth section provides conclusions and suggestions for future work.

2. Concepts, Literature Review and their Challenges

In this section, the concepts of text summarization are discussed first, and then the work done in the field of summarizing user reviews will be discussed, and finally, their challenges and problems will be discussed.

2.1 Concepts of text summarization

Text summarization was first introduced by Luhn in 1950 in the first IBM computers using the bag of words method[13]. In this method, the number of frequency of words that were used repeatedly in the text was counted, and based on that, a score was given to each sentence, and summarization was done based on this score. In the following, summarization methods were advanced by using linguistic parameters available in natural language processing. Then, new methods for converting sentences into vectors such as word2vec [14] and deep learning methods using LSTM architectures [15], RNN networks [16] and convolutional neural networks [17] were presented.
In general, there are 2 methods for summarizing texts:

A- Extractive summarization of reviews is done with the aim of identifying words and sentences and using them to create a summary of the text. In this method, the selection of words and sentences is based on their importance. This process includes three parts: separating the sentences and words, calculating the score and selecting the sentences and words with the highest score[18][19][20].

B- Abstractive summarization that has been developed and automated traditional methods. In this method, the key parts of the sentences and the main ideas of the sentence are processed using quoting. This method of summarizing includes the stages of analyzing sentences and quotations, which is done with two methods based on structure and based on meaning[21].

2.2 Work done in summarizing user Reviews

In this part, some of the works done in the field of summarizing reviews will be discussed according to the method used by them. Also, at the end, their challenges and problems will be discussed.

Table 1- Important works presented in the field of text summarization

Year/Reference	Main goal of Research	Challenges & Problems
2009/[22]	Investigating the problems in summarizing texts and providing a classification for summarizing methods	Lack of attention to methods based on extractive and abstractive text summarization, natural language processing, machine learning and deep learning
2014/[23]	Presenting a hybrid method based on extractive and abstractive summarization	In the described method, features based on natural language processing are not used
2014/[24]	Reviewing the work done from 2000 to 2013 and presenting a consolidated method based on statistics.	In this method, the cognitive features of language such as visualization have not been addressed, and its effect on summarization has not been measured
2016/[25]	Presenting two definitive methods for extractive and abstractive summarization of reviews	No testing has been done for the presented method
2017/[26]	A study based on automatic extraction of key words of texts and summarizing them	The method presented by them is not fully and clearly stated and the feature extraction part model is not stated
2017/[20]	Explain the advantages and disadvantages of topic-based, iteration count, and graph-based methods	The stated methods are not well explained.
2017/[27]	Processing related to extractive summarization methods is described in different languages	The exact idea about how to score features and how to extract them is not explained
2020/[28]	The method, processes, main structure, dataset and how to measure the efficiency of automatic summarization models are mentioned.	How to classify and extract features is not described in detail
2020/[29]	Summarizing a set of documents based on previous work	There is no explanation about the different methods

In Table 1, the important works done in the field of summarizing reviews by both extractive and abstractive methods are stated. The presented works have challenges and problems as follows:

- Summarization methods based on deep learning have not been addressed at all
- In some articles, a method for summarization is presented, but the presented method has not been tested with any dataset
- In some summarization methods, the proposed method is not described in full detail
- None of the presented methods are specific and all of them are general and introduced to summarize all the texts.
- The methods of summarizing texts using transformers have not been discussed.
- To summarize the reviews of users of mobile applications, no specific method has been stated

Considering the challenges and problems mentioned above, providing a method or searching for a high-performance method for summarizing the reviews of mobile application users is required.

RQ. Which of the pre-trained models based on extractive text summarization in terms of Rouge criteria is suitable for summarizing the reviews of mobile application users?

3. Pre-Trained Models in User Reviews Text Summarization

3.1 bart_large_cnn

It is a pre-trained model in English and fine-tuned with CNN newspaper news using 400 million parameters[30]. This model is available on the hunggingface.com website, which includes many pre-trained models for various tasks such as summarizing, categorizing, masking, sentiment analysis, searching for text keywords, etc. To use this model, it is sufficient to give the parameters max_length (maximum number of words of the input text) and min_length (minimum number of words of the input text) as input to the model along with the desired text[31].

3.2 bart_large_XSUM

The model was trained using 226 million BBC articles from 2010 to 2017 in the categories of politics, news, weather, sports, business, science, health, education and family, entertainment and arts.

3.3 mT5_multilingual_XLSum

This model is based on unsupervised learning method using different parameters for Amharic, Arabic, Azerbaijani, Bengali, Burmese, Chinese, English, French, Gujarati, Hausa, Hindi, Igbo, Indonesian, Japanese, Kirundi, Korean, Kyrgyz, Marathi, Nepali, Oromo, Pashto, Pidgin, Portuguese, Punjabi, Russian, Scottish, Serbian, Spanish, Thai, Turkish, Ukrainian, Uzbek, languages and etc.... It Performs tasks such as summarizing, translating, correcting words, language acceptance, etc[32].

3.4 Falcon's AI

This model is trained based on the original T5 model for the summarization task only, so that it can produce accurate and good results in extractive summarization. This model is trained to generate text summaries with higher efficiency than the base T5 model; In addition, this model is trained using a dataset based on summaries made by humans[33].

4. Experimental Design

In this section, the dataset used, the testing environment, the comparison criteria, and how the tests are performed are explained.

4.1 Test Environment

Python programming language version 3.10.11 and Visual Studio Code version 1.78.2 programming environment have been used to test the model. The reason for using this environment is the ease of Debug and compatibility with Microsoft Visual Studio.

To compare the model with other models, the computer of the Big Data Research Center located in the Islamic Azad University of Najafabad branch with an Intel Xeon E5-2650 v4 processor, 16 GB of DDR4 RAM, without a graphics card and Windows 10 was used.

4.2 Used Dataset

The database provided by Kaggle has been used to train the model. Tables 2 and 3 provide complete information about the dataset and its features.

Table 2 - Dataset used along with details

Provider	Number of Reviews	Number of Apps	Number of Categories
Kaggle	51000	10842	32

Table 3 - Features available for each application in the used dataset

Number	Feature
1	App Name
2	Category
3	Average app rating (0 to 5)
4	Number of Reviews
5	App Size in mb
6	Number of Installations
7	Free or paid
8	Price of app in case of not free
9	Age limit for using app
10	Date of last app update
11	Last Version of app

12	Minimum android version required for installing app
13	Reviews for app in text format
14	Reviews Label (Feature Request, Bugfix and Information Giving)

As can be seen in Table 2 and 3, the above dataset contains the play store reviews submitted by users for the application. From the mentioned database, 1000 reviews are randomly selected and given to 4 models for summarization.

4.3 Data Preprocessing

Before sending each review to the summarization models, we have performed text-related preprocessing operations such as removing special characters (e.g. #, * and ...), whitespace, and punctuation on the data; In addition, all the letters related to reviews have been converted to lower case. The reason for doing this is the ease of work for the review summarization system, which works on the basis of transformers [33] [34].

4.4 Rouge Evaluation criterion for pre-trained model evaluation

To answer the research question, our main goal is to compare pre-trained models to find the best model for extractive summarization of user reviews. For this purpose, the models are checked in terms of the F-Mesaure criterion with the Rouge evaluation criterion, which is specific for the evaluation of the summarization methods.
Rouge (Recall-Oriented Understudy for Gisting Evaluation) is a set of benchmarks and a software package specifically designed to evaluate machine summarization, but can also be used for machine translation. These criteria compare a summary or machine translation with reference summaries or translations (of high quality and produced by humans). The rouge criterion itself has subsets that are defined in different articles based on the number n of common tuples between sentences. The main and the summarized sentence are calculated. This means that the rouge1 criterion calculates the number of common 1s between two sentences, the rouge2 criterion calculates the number of 2s in common, and the rougeL criterion calculates the L number of common tuples between two sentences. Then, based on the degree of similarity, precision, recall and F-Measure are calculated[34] and finally, based on the F-Measure parameter, it will be decided whether the presented method is suitable for summarization or not.

4.5 How to Perform the Tests

RQ. To find the best model in summarizing user reviews, first, 1000 reviews are randomly selected from the mentioned dataset and then given to each of the models for summarization separately. Each test is repeated 10 times and summarized reviews are kept at each stage. Then, through the libraries available in the Python software, for each model, 1000 original texts along with 1000 summarized texts are given to Python, and then each review is compared with its summarized review separately, and the rouge measure is compared with F-Measure. It is calculated for that. In the following, the amount of this parameter is recorded, the next review along with the summary is prepared for processing. Finally, the average parameters of rouge1, rouge2 and rougeL are calculated for each review

5.Experiment Results

RQ. In this part, to answer the research question, the results of the tests related to the pre-trained models in summarizing reviews are calculated according to the rouge criterion.

Table 4 - Comparison of pre-trained models in summarizing 1000 reviews

Model	rouge1 Average	rouge2 Average	rougeL Average
bart_large_cnn	0.3801	0.3517	0.3521
bart_large_XSUM	0.1736	0.0518	0.1434
mT5_multilingual_XLSum	0.1976	0.0602	0.1695
Falcon'sAI	0.6464	0.6140	0.6346

As can be seen in Table 4, Falcon's AI model has a better result than other models in summarizing reviews.

- It should be noted that in none of the articles', pre-trained models have not been compared for the task of summarizing the reviews of mobile application users.
- On the other hand, because the Falcon's AI model has been trained with different datasets of reviews, texts, news, etc. for summarizing, it has been able to get better results in summarizing users' reviews.

6.Conclusion & Future Work

Due to the fact that the number of reviews submitted for applications is very large, summarizing them by the development team is a difficult and time-consuming task. If there is a method or a tool to summarize the reviews, it can save the time of the development team and help to implement new features in the application, fix their bugs and make the application successful. There are many pre-trained models for summarizing texts, but none of them have been specifically adjusted for summarizing reviews. In this article, 4 pre-trained models were compared in the extractive summarization of reviews according to rouge parameter in summarizing 1000 reviews from Kaggle dataset. Finally, it was found that the Falcon's AI method is a suitable method for the extractive summarization of reviews. By using the pre-trained model in summarizing reviews, the development team will easily have a summary list of reviews after categorizing the reviews, and will not waste time reading long reviews from the development team. Falcon's AI model was able to obtain a score of 0.6346 in the rougeL parameter due to the use of many parameters and precise adjustment using texts in different categories. This means that the degree of similarity of the summarized text with the original text is appropriate.

In the future, more pre-trained models can be examined and compared in summarizing reviews, if there is a dataset, a model can be presented for abstract summarization of users' reviews also in the field of Persian language, pre-trained models.

References

[1] M. R. Dehkordi, H. Seifzadeh, G. Beydoun, and M. H. Nadimi-Shahraki, “Success prediction of android applications in a novel repository using neural networks,” *Complex Intell. Syst.*, vol. 6, no. 3, pp. 573–590, 2020, doi: 10.1007/s40747-020-00154-3.

[2] W. Martin, F. Sarro, Y. Jia, Y. Zhang, and M. Harman, “A survey of app store analysis for software engineering,” *IEEE Trans. Softw. Eng.*, vol. 43, no. 9, pp. 817–847, 2017, doi: 10.1109/TSE.2016.2630689.

[3] M. Razavi, H. Rastegari, and A. Nabiollahi-najafabadi, “User Reviews Classification in Play Store Applications Using Deep Learning : An Empirical Study,” pp. 43–57.

[4] 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)*, IEEE, Aug. 2014, pp. 153–162. doi: 10.1109/RE.2014.6912257.

[5] D. Pagano and W. Maalej, “User feedback in the appstore: An empirical study,” in *2013 21st IEEE International Requirements Engineering Conference (RE)*, IEEE, Jul. 2013, pp. 125–134. doi: 10.1109/RE.2013.6636712.

[6] L. V. G. Carreno and K. Winbladh, “Analysis of user comments: An approach for software requirements evolution,” *Proc. - Int. Conf. Softw. Eng.*, pp. 582–591, 2013, doi: 10.1109/ICSE.2013.6606604.

[7] W. Maalej and D. Pagano, “On the socialness of software,” *Proc. - IEEE 9th Int. Conf. Dependable, Auton. Secur. Comput. DASC 2011*, pp. 864–871, 2011, doi: 10.1109/DASC.2011.146.

[8] N. Seyff, F. Graf, and N. Maiden, “Using mobile RE tools to give end-users their own voice,” *Proc. 2010 18th IEEE Int. Requir. Eng. Conf. RE2010*, pp. 37–46, 2010, doi: 10.1109/RE.2010.15.

[9] A. Al-Subaihin *et al.*, “App store mining and analysis,” in *Proceedings of the 3rd International Workshop on Software Development Lifecycle for Mobile*, New York, NY, USA: ACM, Aug. 2015, pp. 1–2. doi: 10.1145/2804345.2804346.

[10] 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 - International Conference on Software Engineering*, IEEE Computer Society, May 2014, pp. 767–778. doi: 10.1145/2568225.2568263.

[11] E. C. Groen, J. Doerr, and S. Adam, “Towards Crowd-Based Requirements Engineering A Research Preview,” in *Requirements Engineering: Foundation for Software Quality*, S. A. Fricker and K. Schneider, Eds., Cham: Springer International Publishing, 2015, pp. 247–253.

[12] S. A. Licorish, B. T. R. Savarimuthu, and S. Keertipati, “Attributes that predict which features to fix: Lessons for app store mining,” *ACM Int. Conf. Proceeding Ser.*, vol. Part F1286, pp. 108–117, 2017, doi: 10.1145/3084226.3084246.

[13] H. P. Luhn, “The Automatic Creation of Literature Abstracts,” *IBM J. Res. Dev.*, vol. 2, no. 2, pp. 159–165, 2010, doi: 10.1147/rd.22.0159.

[14] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, “AMMUS : A Survey of Transformer-based Pretrained Models in Natural Language Processing,” pp. 1–42, 2021, [Online]. Available: http://arxiv.org/abs/2108.05542

[15] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[16] A. M. and G. H. Alex Graves, “Speech Recognition with Deep Recurrent Neural Networks , Department of Computer Science, University of Toronto,” *Dep. Comput. Sci. Univ. Toronto*, vol. 3, no. 3, pp. 45–49, 2013, [Online]. Available: https://ieeexplore.ieee.org/stampPDF/getPDF.jsp?tp=&arnumber=6638947&ref=aHR0cHM6Ly9pZWVleHBsb3JlLmllZWUub3JnL2Fic3RyYWN0L2RvY3VtZW50LzY2Mzg5NDc/Y2FzYV90b2tlbj1OQUo1VFJxWk5JRUFBQUFBOmtPZmdDbS00NGhqaGI2N3dMd2JrU3lSaEdJREhBWnpMSkxoT201Um5YMXR0S0poUDAtM2hkbT

[17] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.

[18] V. Gupta and G. S. Lehal, “A Survey of Text Summarization Extractive techniques,” *J. Emerg. Technol. Web Intell.*, vol. 2, no. 3, pp. 258–268, 2010, doi: 10.4304/jetwi.2.3.258-268.

[19] N. Moratanch and S. Chitrakala, “A survey on extractive text summarization,” *Int. Conf. Comput. Commun. Signal Process. Spec. Focus IoT, ICCCSP 2017*, no. November, 2017, doi: 10.1109/ICCCSP.2017.7944061.

[20] M. Allahyari *et al.*, “Text Summarization Techniques: A Brief Survey,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 10, 2017, doi: 10.14569/ijacsa.2017.081052.

[21] N. Moratanch and S. Chitrakala, “A survey on abstractive text summarization,” *Proc. IEEE Int. Conf. Circuit, Power Comput. Technol. ICCPCT 2016*, no. November, 2016, doi: 10.1109/ICCPCT.2016.7530193.

[22] S. Gholamrezazadeh, M. A. Salehi, and B. Gholamzadeh, “A Comprehensive Survey on Text Summarization Systems.” doi: 10.1109/CSA.2009.5404226.

[23] G. L. A. Babu and S. Badugu, “A Survey on Automatic Text Summarisation,” *Lect. Notes Networks Syst.*, vol. 612, pp. 679–689, 2014, doi: 10.1007/978-981-19-9228-5_58.

[24] R. Mishra *et al.*, “Text summarization in the biomedical domain: a systematic review of recent research.,” *J. Biomed. Inform.*, vol. 52, pp. 457–467, Dec. 2014, doi: 10.1016/j.jbi.2014.06.009.

[25] N. Andhale and L. A. Bewoor, “An overview of text summarization techniques,” *Proc. - 2nd Int. Conf. Comput. Commun. Control Autom. ICCUBEA 2016*, no. May, 2017, doi: 10.1109/ICCUBEA.2016.7860024.

[26] J. R. Thomas, S. K. Bharti, and K. S. Babu, “Automatic keyword extraction for text summarization in e-newspapers,” *ACM Int. Conf. Proceeding Ser.*, vol. 25-26-August-2016, 2016, doi: 10.1145/2980258.2980442.

[27] M. Gambhir and V. Gupta, “Recent automatic text summarization techniques: a survey,” *Artif. Intell. Rev.*, vol. 47, 2017, doi: 10.1007/s10462-016-9475-9.

[28] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, “Automatic text summarization: A comprehensive survey,” *Expert Syst. Appl.*, vol. 165, no. November 2021, 2021, doi: 10.1016/j.eswa.2020.113679.

[29] L. Abualigah, M. Q. Bashabsheh, H. Alabool, and M. Shehab, “Text Summarization: A Brief Review,” *Stud. Comput. Intell.*, vol. 874, no. January, pp. 1–15, 2020, doi: 10.1007/978-3-030-34614-0_1.

[30] M. Lewis *et al.*, “BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” *CoRR*, vol. abs/1910.1, Oct. 2019, doi: https://doi.org/10.48550/arXiv.1910.13461.

[31] S. Narayan, S. B. Cohen, and M. Lapata, “Don’t give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization,” *Proc. 2018 Conf. Empir. Methods Nat. Lang. Process. EMNLP 2018*, pp. 1797–1807, 2018, doi: 10.18653/v1/d18-1206.

[32] T. Hasan *et al.*, “{XL}-Sum: Large-Scale Multilingual Abstractive Summarization for 44 Languages,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Online: Association for Computational Linguistics, Aug. 2021, pp. 4693–4703.

[33] L. Basyal and M. Sanghvi, “Text Summarization Using Large Language Models: A Comparative Study of MPT-7b-instruct, Falcon-7b-instruct, and OpenAI Chat-GPT Models,” 2023.

[34] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries,” in *Text summarization branches out*, 2004, pp. 74–81.