Mobile Application User Review Based Feature Request and Bug Discovery

By A.D.S.R.Gunathilaka

Overview

- Introduction
- Previous Studies
- Proposed Approach
- Revised Approach

Introduction

Introduction : What is Requirement Elicitation

- ❖ Requirement Elicitation is the practice of understanding and capturing the business domain knowledge, stakeholder goals, and user needs.
- ❖ It is a critical activity in the Requirement Engineering (RE) process, and it plays a significant role in the overall quality of the RE outcome [1].

[1] D. Zowghi and C. Coulin, “Requirements elicitation: A survey of techniques, approaches, and tools,” in Engineering and managing software requirements. Springer, 2005, pp. 19–46

Introduction : Importance of user feedback

- User involvement is a major contributor to success of software projects [2].
- User comments can be used to improve user satisfaction of software products [5].
- Feedback typically contains multiple topics related to the application, such as user experience issues, bug reports, and feature requests [3] [4].
- Feedback content has an impact on download numbers of the application [4].
- Majority of low star rating feedback usually contains shortcomings and bug reports of the application, where as four to five star ratings mainly consist of praise and feature requests [4].

[2] M. Bano and D. Zowghi, "A systematic review on the relationship between user involvement and system success," Information and Software Technology, vol. 58, 06 2014.
[3] D. Pagano and B. Bruegge, "User involvement in software evolution practice: A case study," 05 2013.
[4] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," 07 2013.
[5] H. Li, L. Zhang, L. Zhang, and J. Shen, "A user satisfaction analysis approach for software evolution," 2010 IEEE International Conference on Progress in Informatics and Computing, vol. 2, pp. 1093–1097, 2010.

Previous Studies:Summary of existing approaches

Source	Preprocessing	Processing
Carreno and Winbladh, 2013 [13]	Tokenizing, Lower case Conversion, removing non-words and non-numerical,stopwords	Topic modelling and Sentiment Analysis
Fu et al., 2013 [14] (WISCOM)	Removing non-english comments, Splitting strings into word using predefined delimiters(. , : ( ) / [ ] ! * ; " ' +), lower case conversion, removal of uncommon words	Analysis of Inconsistent reviews: sentiment analysis and linear regression model, Topic analysis: LDA
Guzman and Maalej, 2014 [11]	Noun, verb, and adjective extraction, Stopword removal, Lemmatization.	Sentiment Analysis and Topic modelling with LDA
Chen et al., 2014 [15] (AR-Miner)	Converting the raw user reviews into sentence level reviews, tokenizing, removal of all non alphanumeric symbols, lowercase conversion, removal of extra whitespace, stop words and rare words and stemming.	Review filtering :EMNB (Expectation Maximization for Naive Bayes) and Topic modelling: LDA and ASUM (Aspect and Sentiment Unification Model)

Previous Studies:Summary of existing approaches

Vu et al., 2015 [8] (MARK)	Misspelled words, acronyms, and abbreviations and Non English reviews removal,Word stemming and PoS tagging	Ranking: Sentiment analysis,Clustering:K-means, Search and Trend Analysis: VSM(Vector Space Model)
Gu and Kim, 2015 [7] (SUR-Miner)	Separating sentences,fixing common typos and contractions	Classification: Max Entropy, Text feature Extraction: TrunkWords, Character N-Gram, POS tag and Parsing tree
Guzman et al., 2015 [16]	Noun, verb, and adjective extraction, Stopword removal, Lemmatization.	Naive Bayes, Support Vector Machines (SVMs), Logistic Regression Neural Networks and Ensembles of them.
Maalej et al., 2016 [10]	stop-word removal, stemming, lemmatization, tense detection, and bi-grams.	Classification: Naive Bayes, Decision Tree and Maximum Entropy

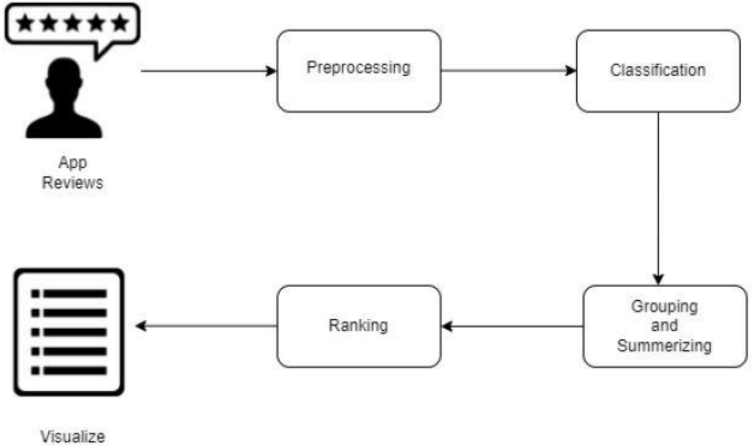
Previous Studies:Summary of existing approaches

Anchiêta and Moura, 2017 [17]	Removal of emoji and emoticons, reviews with five stars, and reviews with less than three words,Tokenizing, removing of stop words, and stemming	Clustering with K-means with BoW (Bag-of-words) model and TF-IDF then Topic modelling with LDA and NMF (Non-negative Matrix Factorization)
Guzman et al., 2017 [18] (ALERTme)	Tokenizing, Lower case Conversion, extracting n-grams, removing stop words and stemming	MNB (Multinomial Naive Bayes), BTM (Biterm Topic Model)
Dhinakaran et al., 2018 [19]	Removal of stop words and lemmatizing	Naive Bayes, Logistic Regression, and Active Learning.

Previous Studies:Summary of existing approaches

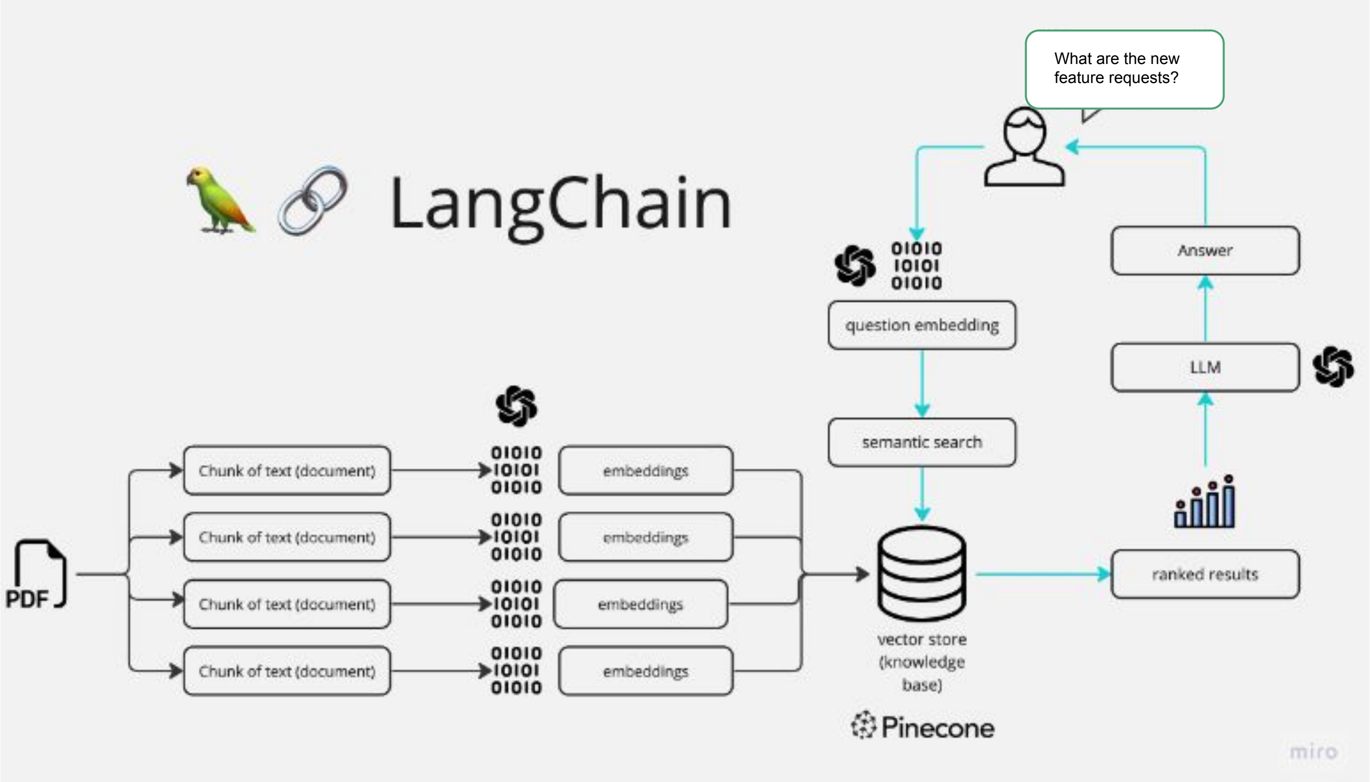
Stanik et al., 2019 [20]	Traditional machine learning :lowercase conversion, masking account names, links,hashtags and lemmatization. Deep Learning: -	Traditional machine learning feature extraction: POS tagging, TF-IDF, sentiment, fast-Text,Traditional machine classification: Decision Tree, Random Forest, Naive Bayes, and Support Vector Machine, Deep Learning: CNN, Transfer learning, Hyper tuning
Aslam et al., 2020 [21]	Spell checking, removal of special characters , stop words,lowercase conversion tokenizing and lemmatization.	Feature extraction:Sentiment analysis, Classification :CNN
Hadi and Fard, 2021[22]	-	PTMs (BERT, XLNet, RoBERTa and ALBERT)
Restrepo et al., 2021 [23]	Tokenizing using BERT tokenizer and adding paddings to tokens	Classification: Transfer Learning and PTMs(BERT and MBERT)

Proposed Approach : High Level Solution[15,18]



[15] N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, “Ar-miner: mining informative reviews for developers from mobile app marketplace,” in Proceedings of the 36th international conference on software engineering, 2014, pp. 767– 778.
[18] E. Guzman, M. Ibrahim, and M. Glinz, “A little bird told me: Mining tweets for requirements and software evolution,” 09 2017.

Revised Proposed Approach : High Level Solution



[24] https://github.com/alejandro-ao/ask-multiple-pdfs

References

1. D. Zowghi and C. Coulin, “Requirements elicitation: A survey of techniques, approaches, and tools,” in Engineering and managing software requirements. Springer, 2005, pp. 19–46
2. M. Bano and D. Zowghi, “A systematic review on the relationship between user involvement and system success,” Information and Software Technology, vol. 58, 06 2014.
3. D. Pagano and B. Bruegge, “User involvement in software evolution practice: A case study,” 05 2013.
4. D. Pagano and W. Maalej, “User feedback in the appstore: An empirical study,” 07 2013.
5. H. Li, L. Zhang, L. Zhang, and J. Shen, “A user satisfaction analysis approach for software evolution,” 2010 IEEE International Conference on Progress in Informatics and Computing, vol. 2, pp. 1093–1097, 2010.
6. W. Maalej, M. Nayebi, T. Johann, and G. Ruhe, “Toward data-driven requirements engineering,” IEEE Software, vol. 33, pp. 48–56, 01 2015.
7. N. Aslam, A. RAMAY, X. KEWEN, and N. Sarwar, “Convolutional neural network-based classification of app reviews,” IEEE Access, vol. 8, pp. 1–11, 10 2020.
8. X. Gu and S. Kim, “"what parts of your apps are loved by users?" (t),” 11 2015, pp. 760–770.
9. P. M. Vu, T. T. Nguyen, H. V. Pham, and T. T. Nguyen, “Mining user opinions in mobile app reviews: A keyword-based approach,” arXiv preprint arXiv:1505.04657, 2015.
10. B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, “Why people hate your app: Making sense of user feedback in a mobile app store,” Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 08 2013.
11. W. Maalej, Z. Kurtanović, H. Nabil, and C. Stanik, “On the automatic classification of app reviews,” Requirements Engineering, vol. 21, 09 2016.
12. 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, 2014, pp. 153–162.
13. J. Verma and A. Patel, “Evaluation of unsupervised learning based extractive text summarization technique for large scale review and feedback data,” Indian Journal of Science and Technology, vol. 10, pp. 1–6, 05 2017.
14. L. V. G. Carreno and K. Winbladh, “Analysis of user comments: an approach for software requirements evolution,” in 2013 35th international conference on software engineering (ICSE). IEEE, 2013, pp. 582–591.
15. B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, “Why people hate your app: Making sense of user feedback in a mobile app store,” Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 08 2013.
16. N. Chen, J. Lin, S. C. Hoi, X. Xiao, and B. Zhang, “Ar-miner: mining informative reviews for developers from mobile app marketplace,” in Proceedings of the 36th international conference on software engineering, 2014, pp. 767– 778.
17. R. T. Anchiêta and R. S. Moura, “Exploring unsupervised learning towards extractive summarization of user reviews,” in Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web, 2017, pp. 217–220.
18. E. Guzman, M. Ibrahim, and M. Glinz, “A little bird told me: Mining tweets for requirements and software evolution,” 09 2017.
19. V. Dhinakaran, R. Pulle, N. Ajmeri, and P. Murukannaiah, “App review analysis via active learning: Reducing supervision effort without compromising classification accuracy,” 08 2018, pp. 170–181.
20. C. Stanik, M. Haering, and W. Maalej, “Classifying multilingual user feedback using traditional machine learning and deep learning,” 09 2019, pp. 220–226.
21. N. Aslam, A. RAMAY, X. KEWEN, and N. Sarwar, “Convolutional neural network-based classification of app reviews,” IEEE Access, vol. 8, pp. 1–11, 10 2020.
22. M. A. Hadi and F. H. Fard, “Evaluating pre-trained models for user feedback analysis in software engineering: A study on classification of app-reviews,” 2021.
23. P. Restrepo, J. Fischbach, D. Spies, J. Frattini, and A. Vogelsang, “Transfer learning for mining feature requests and bug reports from tweets and app store reviews,” 08 2021.
24. https://github.com/alejandro-ao/ask-multiple-pdfs

Thank You

Q & A