

EAGRA -Automated Bugs Identification Through Early Access Game Review Analytics On Game Distribution Platforms

Ali Shahbaz^[0000-1111-2222-3333] and Dr. Khubaib Amjad Alam^[1111-2222-3333-4444]

¹ Fast National University of Computer & Emerging Science, Islamabad, Pakistan

² Fast National University of Computer & Emerging Science, Islamabad, Pakistan
lncs@springer.com

Abstract. User reviews are considered one of the most important source of information about an app and game. The classification and analysis of reviews in order to extract information has proven to be a considerable difficulty. Game applications receive user input in the form of reviews, which can assist developers in selecting games with improved functionality and extracting relevant information such as user feedback, problems, and descriptions of user experiences linked with existing features. Because of the enormous user base and potential benefits of automated feature and bug extraction, game application review analysis has lately arisen as an active topic of research in software engineering. Recently, several research studies have been conducted to mine and categorize user-reviews into actionable software maintenance requests, including feature requests and bug reports for different software systems. However, existing literature have mostly focused on mining functional aspects and analysis of use for software systems. But for gaming industry existing literature is limited to manual analysis of mining functional aspects and analysis of use. To achieve user satisfaction and to survive in the gaming market, addressing these bugs reports to developers is necessary. Therefore, this research formulates this problem as a Multi-label classification problem and propose a bug classification models using RNN (recurrent neural network), LSTM (long term short term memory) and CNN (Convolutional neural network) all vary in accuracy. Representative features are used to train a model for bug classification. Use of feature extraction methods to train a classification model may lead to divergent results, which implies the need for a careful selection of these methods. Several recent studies have emphasized basic state-of-the-art taxonomies for bug classifications into different categories. Therefore, keeping in view these taxonomies this research employs these taxonomies to provide a tools which takes a user reviews dataset and then identify bugs from them and after identification, it will classify the bug into their related bug categories using classification model. This research taking GAME STEAM ENGINE as a case study to scrap gaming reviews and train model on that reviews for bugs classification.

Keywords: Text mining, classification, software repositories, open source software projects, triaging, feature extraction, Gaming reviews.

1 Introduction

Computer games have grown in popularity throughout time, resulting in a multi-billion-dollar industry. The size of the computer gaming industry makes it difficult to create a successful game. In addition, prior studies show that gamers are extremely hard to please, making the quality of games an important issue. Making game quality a crucial consideration. The majority of online gaming stores allow consumers to leave feedback on a game they purchased. Such reviews have the power to make or ruin a game. Other potential consumers frequently rely their purchasing decisions on a game's reviews. There is existing literature in which there is a lot of work on social media reviews, product reviews and app reviews on different dimension [1]. They have limited work on user satisfaction, user requirements and sentimental analysis however existing literature have non-existing automated methods to extract gaming bugs from reviews. Gaming reviews have started gaining attention recently in gaming industry. As a result, reading game evaluations can aid game makers in better comprehending user issues and continue to increase the games' user-perceived quality [2]. It has gained a lot of momentum and is now a major software development industry. Platform games are back in both 2D and 3D (the developed version). A big challenge for them is that the industry lacks an automated system-level approach to gaming test. Currently, most game development organizations. The game is tested using manual or semi-automatic techniques. Such testing methods do not meet industry requirements if you need a more systematic and repeatable approach [3].

Application distribution platforms, such as Google Play or Apple App Store, allow users to rate and review downloaded applications. The popularity of these platforms among both application developers and users has been on the rise in recent years. However, their potential to and impact on requirements engineering processes are not yet well understood [4]. With the growing market and competition in the gaming industry, it is important to produce high-quality games in order to succeed. Developers commonly use bug reports from gamers to locate bugs in their games. Recently, gameplay videos have become popular in the gaming community. Some of these videos showcase a bug, providing developers with new opportunities to collect rich bug information [5].

The app distribution platforms have embraced an open business approach in order to reduce the limitations for small-scale businesses and commercial app developers. Lowering the entry barriers has led to increased competition in the mobile app market. The app distribution platforms have embraced an open business approach in order to reduce the limitations for small-scale businesses and commercial app developers. Lowering the entry barriers has led to increased competition in the mobile app market[4],[5]. The manual analysis is not practical for a large number of reviews, received every day. The research on app review mining has used text mining techniques to extract useful information from the reviews. One way to find valuable information in reviews is to train an ML-based classification model to automatically group the reviews according to their content [6].

If a bug persists into a game's release, it can have a negative impact on both the developers and users. Despite this Effects, it is common to launch games with existing bugs Fixed by subsequent updates in 2014, for example, three-quarters of big-budget games "released on Xbox One, Wii U and PS4" received an update within 24 hours of the game's initial release. The different development processes for games and traditional software may be responsible for the appearance of these bugs. Previous research has shown that it is difficult to test all aspects of a video game comprehensively. Developers have difficulty writing comprehensive tests, because games can have a significantly large number of possible user interactions when compared to other types of software. As a result, many games will be released with it undiscovered bugs that only reveal themselves to the customer start playing the game [7].

This difficulty can often lead to players experiencing failure, which negatively affects the game experience for some [8]. This challenge draws attention to a more comprehensive analysis of reviews to identify issues and requirements, given the potential value that the text of the review means. As per the above discussion there was a need of tool which takes a user review dataset and then identify bugs form them and after identification it will classify the bug into their related bug categories. The goal of this thesis was to automate the extraction of gaming bugs from steam engine game reviews, train classification models to categorize user reviews under different labels, and make the bug categorization process more efficient.

2 Problem Formulation

Mining and analyzing user game reviews has emerged as an important challenge for game developers. There are millions of games in the highly competitive gaming market, and these games have billions of user reviews. This excessive amount of user ratings makes it difficult to analyze and manually interpret the ratings for information extraction.

Software vendors are accumulating huge amounts. Implicit feedback in the form of usage data, error logs, and sensor data. These trends point to a shift to data-driven user-centered identification, prioritization, and management of software requirements. While dealing with large set of data training of classification models are performed using feature extraction [9].

However, the importance of these models for multi-label text classification is not clear. This paper addresses this issue and evaluates the performance of the classification model. The approach to bug classification in game reviews refers to different datasets or evaluation strategies, so the results cannot be compared. To solve this problem, this paper first evaluates the performance of basic feature extraction methods for labeled datasets and the performance of monitored deep learning models. The previous studies have more complex methods to extract features from the reviews [10], [11]. However, because reviews are in text format and contain additional information that isn't important to get as a feature of your app, this strategy risks extracting a lot of irrelevant features. To address this challenge, this task involves training a

convolutional neural network model to perform validation classifications without explicit use of feature extraction methods.

3 Problem Solution

This section describes the proposed system for automated classification of bugs, data used for classification and results obtained in different classification models.

3.1 Input Data

Reviews are scrapped from STEAM ENGINE GAME PLATFORM using customized web scrapper known as SCRAPY. After Pre-processing reviews are labelled into fifteen different bugs categories [1].

3.2 Pre-Processing

The key stage in data mining is data pre-processing. Since the data from gaming reviews is unprocessed, it cannot be used to train the classification system directly. Because it contains unnecessary information, special symbols etc. In order to make the data useable for training purposes, it is first pre-processed. For pre-processing NLP techniques stop word removal, bag of word and lemmatization are applied before passing data for processing to model.

3.3 Feature Selection & Bugs Labelling

After using the "bag of words" technique on the data, a very large dimensionality was obtained in the vocabulary. The majority of these parameters have no bearing on text categorization, which lowers the classifier's effectiveness. The method of feature selection, which selects the top k phrases from the entire lexicon and reduces the dimensionality, is used to increase accuracy and efficiency.

3.4 MLBCGR-LSTM for Prediction

After conducting pilot studies using multiple classification models (RNN, CNN, LSTM). LSTM have shown higher accuracy for multi-label gaming bugs classification. It's working diagram is shown in figure 1.

LSTM receives five arguments and have four dimensions.

It's three layers include

- Output-Dim (Output layer: number of nodes in output layer will be same as input layer)
- Hidden-Dim (Hidden layer: size of hidden layer. It the size of hidden-start of the LSTM)
- Input-Dim (Total number of unique word in sample data)
- Embedding-dim (Size of each embedding vector. Here embedding dimension of GloVe word embedding vectors is 100)

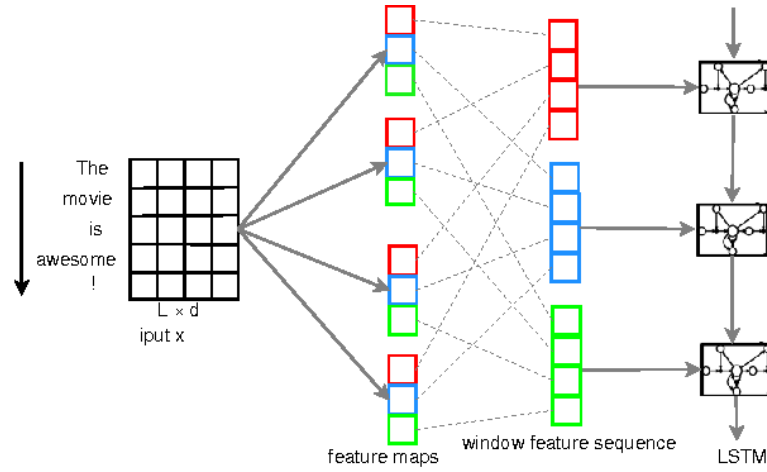


Figure 1 LSTM Model Architecture

There is a multiple feature selection technique such as Pearson Correlation Recursive Feature Elimination, Frequency Inverse Document Frequency (TFIDF). In this research we have used TFIDF for feature selection. Bugs are labelled into fifteen different categories (shown in figure 1) of gaming bugs by gaming experts.

4 Experimental Results and Discussion

After dataset analysis, reviews text is cleaned as described above then Pre-Processing is applied using the best approaches that had applied before in literature. The dataset of user reviews is collected from fifteen games of different categories. Then ten thousand review are labeled manually by gaming experts. The architecture diagram is shown below

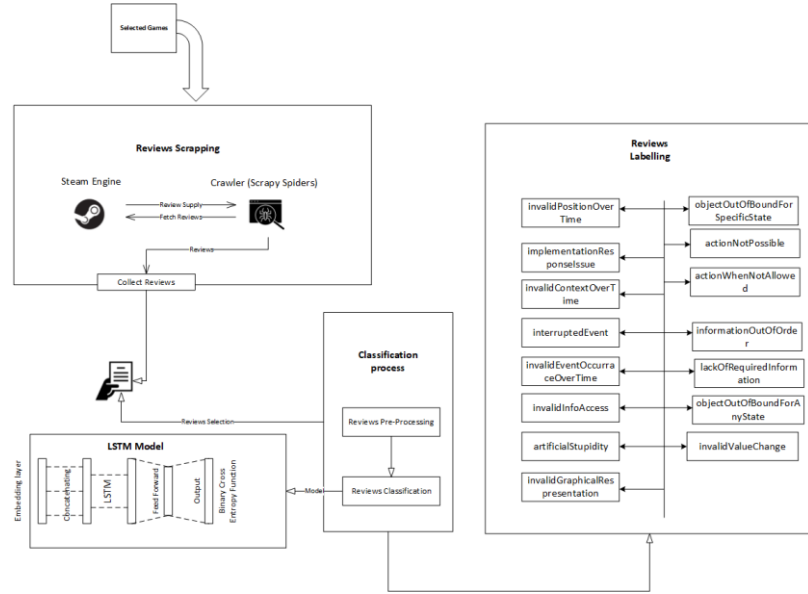


Figure Architecture Diagram of Proposed Solution

To classify the reviews a deep learning LSTM model is designed which is trained over the fifteen thousand reviews dataset. Initially model is trained on 1500 reviews as a pilot study to compare it with other model CNN and RNN. Later model is trained over 10,000 labelled reviews in which 7200 reviews are used for training and 1800 reviews are for validation. Detailed results of each epoch (hyper parameter that defines the number times that the learning algorithm will work through the entire training dataset) are shown in below table.

Epochs	Accuracy	Val-Accuracy
1	0.8234	0.8219
2	0.8237	0.8222
3	0.8344	0.8329
Average	0.8271	0.8256

5 Limitation and Future Work

In the context of ongoing work, the proposed multi-label bug classification model focuses mainly on classifying the bugs into fifteen categories which are defined in existing literature [8]. Also the number of epochs are set to minimum which can also be increased to achieve the better results. This implementation has used the GloVe word embedding technique however other embedding techniques like word2Vec can also be used. The proposed model is evaluated on the basis of accuracy other factors can also be considered to measure its accuracy.

6 Conclusion

This thesis aims at automated classification of bugs in gaming reviews to facilitate developers about what they can improve in games. Existing literature have very serious limitations in terms of multi label classification of bugs in gaming industry. Most of the existing literature is limited to abstract level of bugs classifications. In addition, automated model for the classification of multi-label gaming reviews is rarely seen. But in order to meet the gamers expectations and increase the satisfaction of gamers it is much needed to detect multiple bugs from games. By their solution it can not only increase the quality of the games but also increase the number of users of the games.

State of the art studies have given the taxonomies and studies for classifications of bug which require an immediate need of more comprehensive way to detect it from user reviews. In this research MLBCGR-LSTM algorithm was used to which classify the bugs into specified labels.

For automated multi-label bugs classification of gaming reviews this research implies machine learning based models and deep learning based models such as CNN, RNN and LSTM. These models are implemented using python. On the basis of results accuracies of pilot studies an automated multi-label bug classification model MLBCGR-LSTM is implemented using python with fifteen different games with 10,000 reviews dataset. To classify the bugs into multi label classification different objectives were defined and proved on the course of this research.

References

- [1] G. E. Gorman, "Application of social media analytics: a case of analyzing online hotel reviews," *Online Inf. Rev.*, vol. 38, no. 3, 2014, doi: 10.1108/oir-05-2014-002.
- [2] D. Lin, C. P. Bezemer, Y. Zou, and A. E. Hassan, *An empirical study of game reviews on the Steam platform*, vol. 24, no. 1. Empirical Software Engineering, 2019.
- [3] S. Iftikhar, M. Z. Iqbal, M. U. Khan, and W. Mahmood, "An automated model based testing approach for platform games," in *2015 ACM/IEEE 18th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, 2015, pp. 426–435, doi: 10.1109/MODELS.2015.7338274.
- [4] D. Pagano and W. Maalej, "User feedback in the appstore: An empirical study," in *2013 21st IEEE International Requirements Engineering Conference (RE)*, 2013, pp. 125–134, doi: 10.1109/RE.2013.6636712.
- [5] D. Lin, C. P. Bezemer, and A. E. Hassan, "Identifying gameplay videos that exhibit bugs in computer games," *Empir. Softw. Eng.*, vol. 24, no. 6, pp. 4006–4033, 2019, doi: 10.1007/s10664-019-09733-6.
- [6] C. C. Aggarwal and C. X. Zhai, *Mining text data*, 1st ed., vol. 9781461432. Springer New York, NY, 2013.
- [7] A. Truelove, E. S. de Almeida, and I. Ahmed, "We'll Fix It in Post: What Do Bug Fixes in Video Game Update Notes Tell Us?," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 2021, pp. 736–747, doi: 10.1109/ICSE43902.2021.00073.
- [8] C. Lewis, J. Whitehead, and N. Wardrip-Fruin, "What went wrong: A taxonomy of video game bugs," *FDG 2010 - Proc. 5th Int. Conf. Found. Digit. Games*, pp. 108–115, 2010, doi: 10.1145/1822348.1822363.
- [9] W. Maalej, M. Nayebi, and G. Ruhe, "Data-Driven Requirements Engineering-An Update," *Proc. - 2019 IEEE/ACM 41st Int. Conf. Softw. Eng. Softw. Eng. Pract. ICSE-SEIP 2019*, pp. 289–290, 2019, doi: 10.1109/ICSE-SEIP.2019.00041.
- [10] L. F. S. Britto and L. D. S. Pacifico, "Evaluating Video Game Acceptance in Game Reviews using Sentiment Analysis Techniques," *Proc. SBGames, 2020*, pp. 399–402, 2020, doi: 10.1145/3342220.3344924.
- [11] H.-N. Kang, "A Study of Analyzing on Online Game Reviews Using a Data Mining Approach: STEAM Community Data," *Int. J. Innov. Manag. Technol.*, vol. 8, no. 2, pp. 90–94, 2017, doi: 10.18178/ijimt.2017.8.2.709.
- [12] J. P. Zagal, A. Ladd, and T. Johnson, "Characterizing and understanding game reviews," *FDG 2009 - 4th Int. Conf. Found. Digit. Games, Proc.*, no. McCrear 2007, pp. 215–222, 2009, doi: 10.1145/1536513.1536553.
- [13] D. Lin, C. P. Bezemer, Y. Zou, and A. E. Hassan, *An empirical study of game reviews on the Steam platform*, vol. 24, no. 1. 2019.