CritiQ

Game Testing and Evaluation Platform with Machine Learning for Game Developers

Detailed Project Plan

Supervisor: Dr. Chim T. W.

Group Members:

Lee Chi Ho 3035785520

Cheng Pak Yim 3035784942

Siu Yuk Shing 3035779430

Submission Date: 1st October 2023

Department of Computer Science, The University of Hong Kong

Table of Contents

1. Project Background and Motivation

Game review platforms, such as Steam and Opencritic, primarily cater to end-users and offer a collection of critic and player reviews for games that are released or in "Early Access". However, these platforms have limited functionality for sorting and filtering reviews. As a result, players and game developers who want to get a comprehensive overview of game feedback must manually sift through numerous reviews and perform analysis themselves, which is inconvenient or inefficient.

This project's objective is to develop an automated review analysis system and web application for both developers and players. Considering indie game developers who usually work independently or in small teams, it is difficult for them to allocate resources to read valuable comments from players (Lin et al., 2019). Two main challenges are faced by them. First, they have limited financial resources, which prevents them from being able to access tools that can do in-depth analysis of player and tester feedback. Second, although indie game developers receive a small number of reviews from game review platforms daily (Lin et al., 2019), their tight development schedule and limited time per week to spend on development provide significant difficulties in reading all the reviews. As a result, many valuable reviews are ignored. Our platform can help them to reduce the time and financial costs associated with manual data analysis.

The proposed project aims to bridge the gap between game developers and players by creating an advanced and intuitive game review platform. Leveraging natural language processing (NLP) techniques and machine learning, the platform will automate the analysis of reviews. Research has shown that early reviews and feedback are crucial for obtaining valuable insights into player sentiments and feedback (Lin et al., 2019). Our platform will use sentiment analysis and topic modeling techniques to give an overview of the overall sentiment and key topics in reviews.

This can help users to extract meaningful information from reviews without manual analysis, saving them time and effort.

Furthermore, our platform aims to support games in development by providing early feedback that can shape future game releases and builds. Studies on Steam game reviews have found that “Early access" and Indie game reviews are longer and more informative than non-early access reviews (Lin et al., 2019), making it more difficult to parse and analyze the content. They often contain critical information about bugs and suggestions that can help developers improve their games (Lin et al., 2019). Our platform will enable developers to gather this information from players during the development phase and help them to make iterative improvements and enhance the overall quality of the game. This feedback loop has been shown to help developers address potential issues, refine features, and make informed decisions that align with players' preferences (Tong, 2021). The platform will establish a collaborative environment between developers and players, which can foster the growth of innovative and engaging games, ultimately benefiting both parties involved.

2. Project Objectives

- Research into Natural Language Processing, including tokenization, stemming, and stopwords removal.
- Sentiment Analysis, Topic Modelling, and Keywords Extraction to provide feedback on user comments.
- Perform web-scraping for data extraction, processing, and model training.
- Provide intuitive data visualization to users in the web application.
- Develop a full-stack scalable modern web application.

3. Methodology

Our project aims to provide a fully featured game review website, including a Frontend Web Application, Backend System, and multiple Machine Learning Models. This section explains the methodology used to develop this platform and its technical details.

This section consists of three subsections. The first subsection will present the Natural Language Processing (NLP) techniques that were applied in this project, the second subsection will describe the Web Application that interacts with the users, and the third subsection will explain the backend infrastructure that supports both the web application and the Machine Learning Processing.

3.1 Machine Learning and Natural Language Processing (NLP)

Our system aims to perform three NLP tasks, namely Sentiment Analysis, Topic Modeling, and Keyword Extraction. The following paragraphs will describe each of these techniques in more detail.

3.1.1 Sentiment Analysis

Sentiment analysis is one of the numerous tasks of natural language processing. It aims at extracting sentiments and opinions from texts (Birjali et al., 2021). The expression of opinions and sentiments on various platforms, such as social media sites, forums, and blogs, become more prevalent among people (Birjali et al., 2021), in which such phenomenon can also be observed in online game platforms and review sites. Since review from other customers plays a significant role in our decision-making process when we buy something, analyzing the general impression of a product is significantly crucial in maintaining the business. Not only do game reviews reflect players' expectations on various aspects, for example, gameplay, sound effects, visuals, and difficulty, but they also demonstrate the engagement of the game's community, which the player can resonate with. Hence, sentiment analysis is an indispensable element in game review analysis, which should be included in our platform for the benefit of both players and developers.

3.1.1.1 Goal

Sentiment analysis can help potential players get a better sense of how a player feels about the game through a review. This way, they can see the overall mood and attitude of the reviewer towards the game. Although game platforms, for instance, Steam, provide labels about the satisfaction of a player with a game in the reviews, this type of binary classification lacks granularity in understanding latent sentiments prevailing in these game reviews. Game reviews often contain hidden mockery and sarcasm that are not common in other types of reviews, such as those for mobile apps, restaurants, or airlines. The presence of humorous and satirical content (Memes) from the community in the review may also obscure the potential players' perception of the game's intended experience. By analyzing the sentiment of various game reviews, the system can filter relevant and contributing comments from a sea of them across different review platforms, such as Steam, Metacritic, and OpenCritic, enhancing their purchasing experience.

For game developers, sentiment analysis contributes to acknowledging the reaction of the gaming community towards their products. This facilitates better decision-making in the development process. It is worth mentioning that reviews with negative sentiment from loyal players can arise after an update of a game, indicating their discontent with the game's community (Lin et al., 2019). Thus, it is crucial to address these issues promptly to maintain a stable number of active players, which exhibited a higher correlation with the number of reviews than the number of installs (Lin et al., 2019), before the drop in impression toward the game after making a deviated decision in gameplay improvement.

3.1.1.2 Model Selection

A variety of machine learning models, such as Support Vector Machine (SVM), multinominal Naïve Bayes (NB), and deep neural networks (DNN), combined with different preprocessing techniques, for instance, bag-of-words and word2vec embedding, were applied to a three-class sentiment classification problem in a previous study on sentiment analysis on game reviews (Ruseti et al., 2020). Despite the decent performance achieved, we recognize room for improvement from sentiment analysis on different domains. For example, harnessing the power of large-language-models in sentiment labeling for a better sentiment classification (Steinert & Altmann, 2023), applying state-of-the-art architectures in the classification (Huang et al., 2023), and integrating metadata of the reviewer along with the review itself as the input of the model (Lin et al., 2019). Hence, model selection should focus on testing these hypotheses and selecting the best performance model.

Currently, a classifier using TF-IDF and Random Forest to perform binary sentiment classification is implemented as a baseline for exploring different combinations of preprocessing techniques and machine learning models. This project will explore and evaluate various preprocessing techniques, such as word2vec embedding and transformer encoding, and apply them to different machine learning models, such as CNN, LSTM, and BERT.

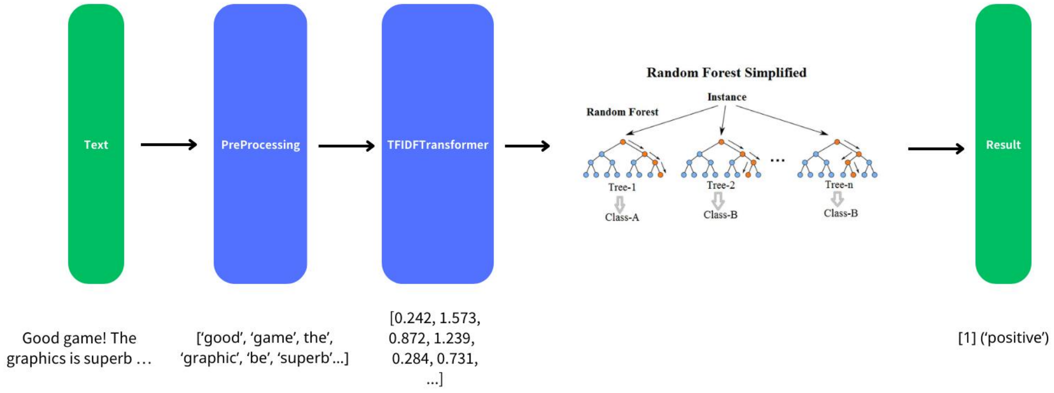

Figure 1: Visualization of the currently implemented baseline. The graph "Random Forest Simplified" was created by Mr. Chung (Yi, 2019).

3.1.1.3 Training

Before training, review data with the labeled sentiment are scraped from existing game review platforms like Steam and Opencritic. The review data then undergoes various preprocessing techniques, such as stop word removal (words that are common and should be ignored by the vectorizer), punctuation and special character elimination, as well as lemmatization and stemming. These techniques help reduce the vector space produced by word embedding through bag-of-words or pre-trained word2vec embedding, which can improve the model accuracy and training performance. Next, review data will be parsed into our machine learning model to learn a binary sentiment classification. Further research shall be performed on the feasibility of providing a "Neutral" Sentiment for reviews and online training with new reviews from platform users.

3.1.2 Topic Modelling

Topic Modelling is an unsupervised machine learning technique that aims to discover the hidden themes in a textual data set (Churchill & Singh, 2022), and thus categorize them. This technique has been widely applied in various fields, from identifying themes in literature (Jockers & Mimno, 2013), to demystifying social media posts (Egger & Yu, 2022), and enhancing our understanding of various types of texts (Churchill & Singh, 2022). Topic modeling helps game developers distinguish constructive reviews containing valuable suggestions from pure semantic opinions, making it possible for them to prioritize areas to improve. Hence, topic modeling has its merits to be integrated into the system.

3.1.2.1 Goal

This section will discuss the goals of applying topic modeling for both player and developer users.

For players, topic modeling enables prospective players to understand certain aspects of a game. Classifying and grouping reviews allows potential players to quickly glance at a specific aspect of a game, such as graphics, compatibility, and gameplay, without the need to read hundreds of fewer related comments. This shortens the purchasing decision-making progress. Overall, topic modeling can provide a better reading and purchasing experience for potential players to find games that best suit their interests and help them make informed decisions.

For developers, topic modeling allows them to better prioritize tasks to be done to cater to their players' needs. Analysis has found that comments that report bugs are not always negative in sentiment (Lin et al., 2019). It shows that sentiment analysis alone cannot distinguish contributing comments from other comments, as players may report bugs constructively or neutrally, rather than expressing frustration or anger. Different from sentiment analysis, topic modeling can distinguish reviews into bug reports and suggestions from unhelpful, pure emotional comments. This technique empowers developers to quickly locate issues that players complain about the most, for instance, game-blocking bugs and crashes (Lin et al., 2019). This enables developers to reprioritize less urgent tasks and reallocate manpower and time to improve the game to provide a more satisfactory gaming experience to the current players, which in turn attracts new players with the positive image projected by the developers.

3.1.2.2 Model Selection

A previous study has applied Latent Dirichlet Allocation (LDA) on Steam reviews to conduct topic modeling on reviews for five games (Stepien, 2021). However, follow-up improvements are required, for example, to improve the readability of categorized topics and provide top-matching reviews within a topic (Stepien, 2021). Besides, an abundance of techniques, such as Bayesian filtering, Word2Vec, lda2vec, and BERTopic have been developed targeting various text lengths and contexts. Hence, we aim to evaluate the performance, efficiency, and training time of the five models and select the best model to apply in the context of game reviews.

3.1.2.3 Training

First, the reviews will be pre-processed with stop words and special character removal, stemming, and lemmatization, to reduce the state space of words. Then, word embedding is applied to convert the words into vectors. Next, LDA and other unsupervised machine learning models will be applied to the reviews to examine their performance in extracting topics. Finally, Visualization tools, for instance, LDAvis will be applied to provide an intuitive overview of the distributions of discovered topics. Hyperparameter tuning on the number of extracted topics is required to fine-tune the performance of the model.

3.1.3 Keyword Extraction

Keyword extraction is a technique to extract a set of words from a document without human effort in reading the texts (Khan et al., 2022). With the exponential growth in content produced daily on the Internet, manually indexing documents, such as scientific journals with keywords, has become impossible. Similar circumstances can be found in-game reviews. Therefore, the ability to generate keywords of keyword extraction proves significant in describing long texts with few keywords. While the game reviews on Steam have a median length of 30 words (Lin et al., 2019), other game review platforms, such as Metacritic and OpenCritic, offer more comprehensive reviews that elaborate on the virtues, problems, and suggestions of the game in details. Thus, it is necessary to index these lengthier reviews to provide a comprehensive overview of the expected game experience.

3.1.3.1 Goal

It is more difficult to find relevant or important keywords from longer reviews than shorter or context-lacking reviews. Using keyword extraction, our application can provide keywords that are more important in a longer review. This significantly reduces the time needed for both players and developers to go over reviews.

For players, keyword extraction can assist prospective players in quickly identifying important aspects of a game that align with their interests the most. Extracted keywords serve as succinct summaries that highlight the features of a game, such as gameplay, storytelling, and sound effects. These keywords can help players scan and compare reviews efficiently, thereby reducing their time and cost in making purchase decisions.

For developers, in addition to the summarization benefits, keyword extraction enables automatic labeling of the model-generated topics to provide a readable tag for developers. Although topic modeling can extract latent themes to some degree, no meaningful description of generated topics is produced after topic modeling. A previous study found that it is necessary to manually assign interpretations to topics generated from game reviews to examine the result of its topic modeling pipeline (Stepien, 2021). Considering the stringent schedules of developers, keyword extraction saves valuable time from manually summarizing categorized reviews in each topic, while providing interpretable results for further follow-up at the same time.

3.1.3.2 Model Selection

During this phase of development, we aim to try out different models and evaluate their effectiveness in keyword extraction. Models, including KeyBERT, Text Rank, Position Rank (Page Rank Algorithm), TF-IDF, and existing libraries such as YAKE (Yet Another Keyword Extractor) will be tested and compared. Because of the specific domain knowledge involved in various types of games, such as strategy, action, and sports, the effectiveness and accuracy of these models on reviews are worth examination.

3.2 Web Application Implementation

3.2.1 Technology and Framework

The front end will be developed using the modern React library, TypeScript, and the Next.JS framework. React is a popular commonly used web framework with many community-made packages. Next.JS, a React meta-framework, will be used to provide a maintainable, responsive, performant, and highly scalable web application. TypeScript is a syntactic superset of the JavaScript language and will be used to provide type safety.

3.2.2 Features and Visualization

The web application can be categorized into Developer Mode and User Mode.

In Developer Mode, users can create, update, and remove their game from our database. This model also provides the aggregated Information processed by our Machine Learning Model and Algorithms, which is a detailed overview of their game reviews and critiques using an intuitive dashboard User Interface, to the game developers. We plan to make use of the graphing library Chart.JS to provide high-performance and responsible graphs to visualize complex data.

As for the User Mode, users will be able to search and find the most popular games on our platforms and review games that are public or have access shared. This mode also allows them to leave comments for reviews made by other users; this is a way to encourage interaction between users and to share their opinions with other reviewers.

3.2.3 Design Approach

The web application will adapt to different screen sizes and orientations by implementing Responsive Web Design. This technique will make the webpage elements adjust their size, position, and visibility according to the viewport so that the website will have a natural and intuitive appearance on various devices, such as phones, desktops, and tablets.

3.2.4 Page Structure

The web application will consist of 7 pages that serve different functions. The detailed explanations are as follows:

1. Login page, where users can enter their credentials to access the website.
2. Register page, where users can create an account to use the website.
3. Add a Review page, where users can write and submit their reviews for games.
4. Edit and add a Game page, where users can modify or upload their games to the database.
5. Developer dashboard page, where users can see the aggregated information and feedback for their games.
6. User landing page, where users can see and find out the most popular games.
7. The profile page allows users to edit their personal information and settings and see their reviews.

3.2.5 Hosting

We plan to host the website on Vercel, which is a web hosting service that provides easy CI/CD and has a free tier for the current time. If needed, the possibility of upgrading to a paid plan will be considered.

3.3 Backend Infrastructure Implementation

3.3.1 Technology and Framework

The backend system will be built using the Java Spring Framework, a common framework used to build high-performance and scalable Enterprise solutions. Spring Boot provides most of the functionality we need for a scalable backend system, including security, MVC, Batch Processing, High Performance, and non-blocking event loops.

3.3.2 Features

The backend system will provide Authentication such as Registration and Login features using secured JWT (JSON Web Token) to provide secured Access. All data access requests will be made to the backend system through RESTful HTTP requests. Most business logic and validation will only be performed in the backend system to provide security and high performance.

3.3.4 Database

MySQL will be used as the database of choice for the relational nature of our data and high performance. The database will be hosted on Digital Ocean, a cloud service provider.
Java provides the JPA (Java Persistence API) and the use of the Spring JPA library will allow for high-performance CRUD operation and reduce boilerplate code for database access. JPA also provides features to support the ACID consistency model. The Database Tables and Entities will solely be created and managed by Spring JPA to ensure data integrity and consistency (See Figure 2).

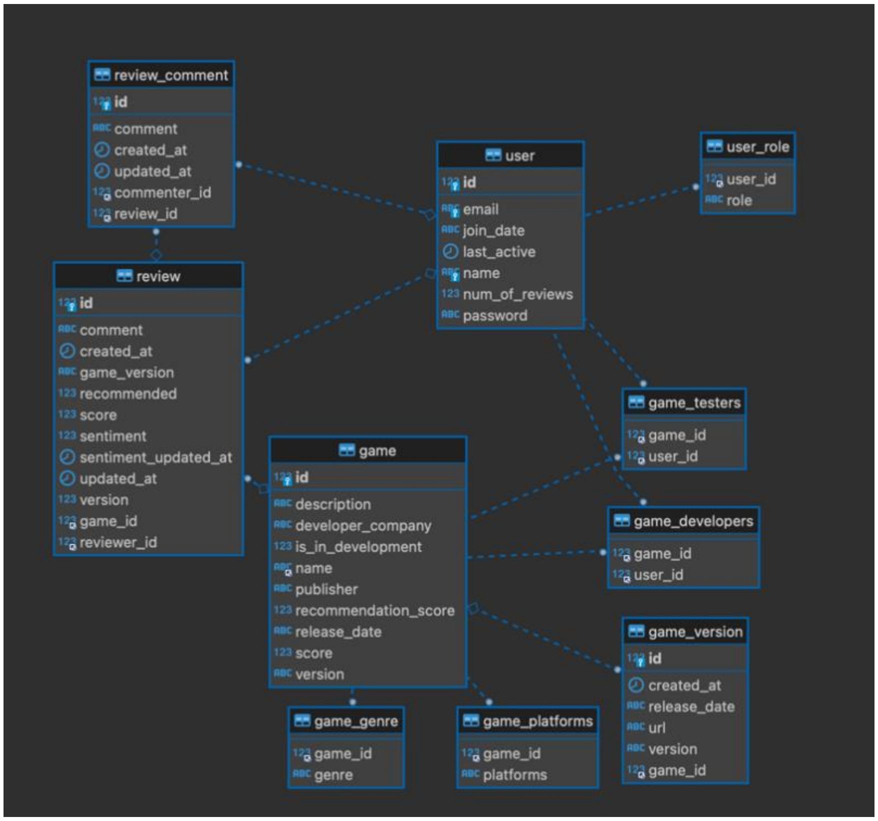

Figure 2: Entity Relations Diagram

3.3.4 Object Storage

This project will utilize a **Simple Storage Service (S3)** compatible storage to store all files provided by Digital Ocean, including text, images, videos, and audio files. Using an S3-compatible storage will allow the application to make use of the Amazon Web Service (AWS) Software Development Kit (SDK) to securely access our files.

The use of S3-compatible storage service will provide high availability, high scalability, and high-performance file access and upload while maintaining simple implementations.

3.3.5 Message Queue

RabbitMQ will be used for inter-process communication that maintains a durable message queue to provide high fault tolerance and asynchronous communication. Message Queue will be used to support Machine Learning features.

In case of system failure or downtime, messages sent to the message queue will remain intact and after resuming functionality, the system can continue to read and parse the messages.

4. Schedule and Milestones

Period	**Work to be done**
Sep - Oct	- Acquire detailed requirements for the web application and Machine Learning Model. - Analyze the feasibility and effectiveness of different machine learning models. - Build up a basic Frontend and Backend to support data entries. - Literature Review and Testing for Machine Learning Model. - Submit Detailed Project Plan and Deliver First Iteration of FYP website.
Oct – Nov	- Basic Web Application Design - Database Structure Design - Scraping data for further model training - Design software architecture and system - Create use case diagram for the application
Nov – Dec	- Model tuning and retraining. - Integration of Model with Backend System for data processing. - Training Sentiment Analysis model - Implement sign up/sign in pages with authentication using secured JWT. - Implement add new games/reviews pages.
Dec - Jan	- Fine-tuning Sentiment Analysis model - Training Topic Modelling Model - Training Keyword Extraction model - Implement landing page/game page for general users and developers to display games and aggregated reviews
Jan – Feb	- Integrate Sentiment Analysis model into our application. - Fine-tuning Topic Modelling Model - Fine-tuning Keyword Extraction model
Feb – Mar	- Final tuning of all models - Integrating all models into our application.
Mar – Apr	- Final tuning of all models - Debug and Refactor Code - Prepare for Final Presentation - Prepare Demo and Demo Video

Table 1: Project Schedule and Milestones

5. Bibliography

Birjali, M., Kasri, M., & Beni-Hssane, A. (2021). A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, *226*, 107134. https://doi.org/10.1016/j.knosys.2021.107134

Churchill, R., & Singh, L. (2022). The Evolution of Topic Modeling. *ACM Comput. Surv.*, *54*(10s), Article 215. https://doi.org/10.1145/3507900

Egger, R., & Yu, J. (2022). A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts [Methods]. *Frontiers in Sociology*, *7*. https://doi.org/10.3389/fsoc.2022.886498

Huang, A. H., Wang, H., & Yang, Y. (2023). FinBERT: A Large Language Model for Extracting Information from Financial Text*. *Contemporary Accounting Research*, *40*(2), 806-841. https://doi.org/10.1111/1911-3846.12832

Jockers, M. L., & Mimno, D. (2013). Significant themes in 19th-century literature. *Poetics*, *41*(6), 750-769. https://doi.org/https://doi.org/10.1016/j.poetic.2013.08.005

Khan, M. Q., Shahid, A., Uddin, M. I., Roman, M., Alharbi, A., Alosaimi, W., Almalki, J., & Alshahrani, S. M. (2022). Impact analysis of keyword extraction using contextual word embedding. *PeerJ Computer Science*, *8*, e967. https://doi.org/10.7717/peerj-cs.967

Lin, D., Bezemer, C.-P., Zou, Y., & Hassan, A. E. (2019). An empirical study of game reviews on the Steam platform. *Empirical software engineering : an international journal*, *24*(1), 170-207. https://doi.org/10.1007/s10664-018-9627-4

Ruseti, S., Sirbu, M.-D., Calin, M. A., Dascalu, M., Trausan-Matu, S., & Militaru, G. (2020, 2020). Comprehensive Exploration of Game Reviews Extraction and Opinion Mining Using NLP Techniques.*Advances in Intelligent Systems and Computing*

Steinert, R., & Altmann, S. (2023). Linking microblogging sentiments to stock price movement: An application of GPT-4. *arXiv e-prints*, arXiv:2308.16771. https://doi.org/10.48550/arXiv.2308.16771

Stepien, K. (2021). Topic Modelling and Data Analysis: Impact of using topic modelling on game reviews. In.

Tong, X. (2021). Positioning game review as a crucial element of game user feedback in the ongoing development of independent video games. *Computers in human behavior reports*, *3*, 100077. https://doi.org/10.1016/j.chbr.2021.100077

Yi, C. (2019). Random Forest Simplified. In.