# Customer Sentiment Analysis for Service Issue Detection Using Negative Sentiment

MSc Research Project
Data Analytics


Anil Prajapati
Student ID: x23167297

School of Computing
National College of Ireland


Supervisor: Musfira Jilani

**National College of Ireland**
**Project Submission Sheet**
**School of Computing**

| | |
|---|---|
| **Student Name:** | Anil Prajapati |
| **Student ID:** | X23167297 |
| **Programme:** | Data Analytics |
| **Year:** | 2023 |
| **Module:** | MSc Research Project |
| **Supervisor:** | Musfira Jilani |
| **Submission Due Date:** | 12/12/2024 |
| **Project Title:** | Customer Sentiment Analysis for Service Issue Detection Using Negative Sentiment |
| **Word Count:** | 5262 |
| **Page Count:** | 17 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

**ALL** internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

| | |
|---|---|
| **Signature:** | |
| **Date:** | 12th December 2024 |

## PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies). | ☐ |
| **Attach a Moodle submission receipt of the online project submission**, to each project (including multiple copies). | ☐ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | ☐ |

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Customer Sentiment Analysis for Service Issue Detection Using Negative Feedback

Anil Prajapati
23167297

**Abstract**

Nowadays, accessing and consuming digital media is lot easier and cheaper than ever before. With this growing demand it is becoming crucial for the product developer to be more accurate while delivering their products. However, it is challenging to deliver a seamless experience and often end up releasing product with issue. Detecting and addressing this becomes crucial for the developers. Develops mostly rely testers and developer forums to address this issue which can be slow and is not able to address all the issue. This work aims to help the product developers to understand and react to the launch issues by understanding the negative sentiment among the user and using that to detect possible issues and address the issue quickly. This study demonstrate the use of Machine Learning (ML) and LLM to detect the negative sentiment and issue with product and services. The user of BERT for the sentiment analysis proved to be contextually accurate compared to VADER and LLM model such as Llama is able to identify issue in those negetive sentiment.

## 1 Introduction

Digital products such as game and software are usually plagued with bugs and issue. Common issue such as incompatibility with the devices, crashing, performance issue and general usability are very common. Users find it often difficult to understand if the issue is with the product or it's a user error. Regardless of the scenario user find it really frustrating and difficult to get the proper support. They are usually not familiar with ways to report this bug and are often relied on review portal to raise their concern. Although reviews are meant to be describing the goods or the bad of the product and help others recognize if the product is really suitable of them, frustrated customer having no choice head to the review section to raise their concern.
During product roll out period, their is a higher chance for such situation where an oversight can result in failure. Having a system which can help in understanding potential issue as early as possible will be a real helpful tool to improve the quality of the product and build a positive reputation for the brand. This system also helps the customers to get their issue resolved as quickly as possible without the need to following any support procedure
This work aim to understand, 'Can LLM be used for identifying faults in product using review data'. We aim to develop a system which uses Machine learning model such as BERT to understand the rising negative sentiment for the product and use LLM model

such as llama to detect the potential issue by asking the model to detect the issue with the product. This system can prove to be helpful to the developer as they will be able to early detect issue which is causing a great amount of negative sentiment to build up for the product.
The result is a end to end system which uses online reviews to detect the issue. The system can be will be very flexible and customization as we are using LLM model and we can easily modify our issue detection as per the developers need. In the next few section, we will discuss about the related work on this topic then we will have a look at the methodology of our proposed system. We will then formulate the design specification and the Implementation of our or system and finally we will have detailed look at the evaluation of our system. We will discuss the concluding in the final part and also suggest future works.

# 2 Related Work

The work involves few key concepts such as sentiment analysis which helps in understanding the sentiment associated with the review and feature extraction/ topic extraction for the comment which will help in detecting the issue. We aim to combine both the process in order to create a useful tool to achieve our goal.

## 2.1 "Using Online Reviews for Customer Sentiment Analysis"

(1) This study aimed at understanding customer sentiment as an essential for innovation and product success as positive words significantly influence consumer adoption. The growing reliance of online reviews and feedback serves as a metrics for sentiments. Traditional Metrics such as online review, volume face has limitations such as being prone to extremity bias as either highly satisfied or dissatisfied customers are more likely to leave reviews. Also, ratings can be skewed and fail to represent an average consumer sentiment.
Also, there can be sudden rise in the volume of reviews in case of controversies or marketing effort which indicates product publicity rather than consumer opinion.
The study analyses amazon reviews for smartphone revealed hat there is discrepancies between sentiment scores and review rating. Emerging brands like ZTE showed higher enthusiasm through longer reviews and positive sentiment scores, compared to big brands like Apple and Samsung. Dictionaries like GI and QDAP produced similar result while others like LM showed variance, underscoring the importance of context-appropriate tools.
The study highlights the advantages of Sentiment analysis such as it helps normalize biases present in the traditional metrics. Provides deeper understanding through additional dimension lie enthusiasm and comparative sentiment scores across brands. Employing text mining tool in conjunction with the traditional methods provides a holistic view of customer feedback.
The potential of text mining as a more nuanced and less biased approach to understand customer sentiment from online review. By incorporating alternative metrics like sentiment scores and review lengths, marketers can gain actionable insights into consumer opinions and enthusiasm, improving decision-making in product innovation and brand strategy.

## 2.2 Sentiment Analysis for Assessing Customer Satisfaction in Chatbot Service Encounters

(2) This literature review synthesizes research on the use of sentiment analysis to evaluate customer satisfaction in chatbot interactions, focusing on four key areas: emotionally intelligent chatbots, chatbot applications in customer service, customer service encounter satisfaction, and sentiment analysis methodologies.

Emotionally intelligent chatbots (EICBs) detect and understand emotions expressed in user interactions through text, voice, or language structure. These chatbots enhance user engagement by responding empathetically, significantly benefiting sectors like mental health and customer service.

Chatbots function as initial support agents, guiding users before escalating issues to human agents when necessary. Their adoption has been driven by cost-efficiency and the ability to mimic human conversational patterns.

The study uses predefined sentiment lexicons to classify words into positive, negative, or neutral categories. Challenges include accounting for contextual polarity and extracting linguistic patterns.

Research emphasizes the need for chatbots to simulate empathy and understand user sentiments better. There is a lack of solutions for capturing and responding to customer sentiments in real-time, a gap this paper seeks to address.

## 2.3 "Polarity Sentiment-Based Intelligent Chat Bot for Judicious Customer Service Escalation"

(3) This study synthesizes existing research to provide context for the development of intelligent chatbots capable of leveraging sentiment analysis to enhance customer service interactions. It highlights advances in natural language processing (NLP), challenges in chatbot systems, and sentiment analysis methodologies.

Chatbots simulate human conversations and can be applied in domains like ticket booking, e-commerce, and customer service. Core technologies include NLP, which involves lexical, syntactic, semantic, and pragmatic analyses for understanding and generating human-like responses.

## 2.4 "Trending Topic Analysis Using a Novel Sub-Topic Detection Model"

(4) This study highlights existing research in sentiment classification and sub-topic detection for Twitter summarization. The study introduces the concept of a Foreground Dynamic Topic Modelling (VF-DTM) framework, addressing issues such as noisy data and conflicting summaries in trending topic analysis. Sentiment classification often relies on supervised learning, requiring large labeled datasets. Twitter datasets, due to their informal and diverse nature, pose significant challenges for manual annotation and domain generalization.

This study highlights challenges in sentiment classification and sub-topic detection, particularly in handling diverse and noisy Twitter datasets. The proposed VF-DTM model aims to address these gaps by providing an efficient, noise-resistant framework for trending topic analysis and summarization.

### 2.5 "LLM-based Vulnerability Detection"

(5) This paper explore the use of LLM models such as chatGPT for detecting and mitigating cyber security vulnerabilities in an IOT environment. The IoT ecosystem faces increasing cyber threats, including DDoS, phishing, and ransomware attacks. Existing methods to mitigate these threats include network segmentation, device management, and behavioral analysis.
The study (6) highlights the role of LLM model such as GPT model which is capable of understanding natural language and is able to analyze vast quantity of text such as logs and security reports and code. This study propose a novel approach VFFinder that localizes vulnerable functions based on Common Vulnerabilities and Exposures (CVE) descriptions and the corresponding source code utilizing Large Language Models (LLMs). Th study (7) mentions about similar model for open source software to detect vulnerabilities. They present a context-based learning approach to enhance the capability of code vulnerability detection named VUL-GPT. Their method combines code retrieval and code analysis, leveraging in-context learning to improve the performance of the GPT model in vulnerability detection. The study (8) also uses a LLM model such as chat GPT-4 for to perfrom natural language processing and detect unwanted malicious code in the Java programming language using the Mixtral model.

### 2.6 "Conversational Product Recommendation Using LLM"

The study(9) investigates the application of Large Language Models (LLMs) as sales assistants in e-commerce to provide conversational product recommendations. It discuses the limitation while using traditional system and focuses on creating dynamic conversation between user and chat bot. The process uses Gemini model and is a two stage process, collating a group of candidate products and answering user question and recommending using product features.
The cahtbot functions as a sales assistant recommending products. The result states a 69% success rate in the first dialogue round. This system is able ot resolve the cold-start problem for a recommendation system.
The study demonstrates the feasibility of using LLMs for conversational product recommendations, highlighting their potential for improving user experience and system scalability. However, further refinements in dialogue generation and safety measures are necessary for real-world deployment.

## 3 Methodology

In our proposed solution and innovative approach to identify product issue. The Figure 1 show the entire process flow on how we will use the review data and how it will go through each model and detect issue. This process will utilize user reviews where we perform sentiment analysis in order to understand the overall positive or negative sentiment of the user. The negative comment will most probably indicate users' dissatisfaction and potential issues they are facing as they are more likely to leave a review in cases of dissatisfaction.
In most of the cases the negative review will be an actual review about the feature and experience of the product of services, however in some cases this comment will highlight problems with product. To perform the sentiment analysis, we will use several sentiment

analysis model and compare.

Our next step will be to analyses the negative comment for extracting the issues. This issue can range from performance, price quality or the overall user experience of the consumer. Using a ML model, we can detect this issue and highlight the category of the issue which we detect. These detected issues will help the developers of the product to address the issues as soon as possible.

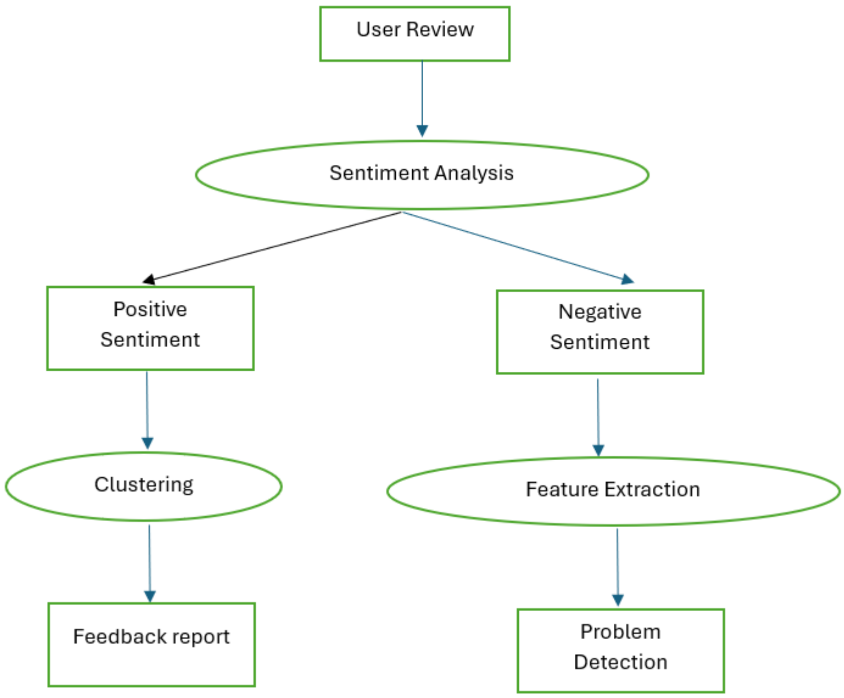


Figure 1: Flow Diagram

## 3.1 Data Gathering

For this study we will be focusing on game review and game related issues. Game development require rigorous testing before deploying, these testing can range from developers to end users performing beta testing on the game. However, despite all the efforts at testing, due the game world and mechanics being too large an complex and also a larger amount of available hardware configuration in the market makes it nearly impossible to test all the possible scenarios. At the end, some bugs or glitches do make it to the finished product and end user will end up facing these issues.
Nowadays games relay being more reliant on online, users are quick to leave a negative review for the product if the face any type of performance issues with their game. We will try to analyses this negative sentiment of the review in order to understand the nature of the issues that the gamers experience.
For our data source, we will use Steam™ (10) which a PC game distribution platform owned by Valve™ Corporation a US based company. Steam is the most popular game distribution platform among the PC community for purchasing games. They provide a

polished platform for gamers to buy the game and has a clean platform of share their experience with other games using their review. After reviewing the Steams' user and subscriber agreement (11) and terms of services(12), we can use the user review as long as we are making a reasonable amount of request to their servers. As long as we do not load the servers severely enough to render the server unusable for the other users, we can scrape these user data. In order to protect the identity of user, we will not capture any userid of the user. Since the platform does not allow multiple review and also restricts user who have not purchases the game, we can be comfortable about the fact that there will be very few duplicates in the data. However, similar comment can be present in the data as the users might just give a short positive comment.
The other issue which we face is the comment are across all the regions where steam is available. Therefor we will find comments which are non-English in our data set. We can translate the non-English comments however, translation can eliminate certain sentimental aspect of the comment and also may alter the context of the comment. Therefore, for simplicity purpose, we will only use the English review for our study.

## 3.2 Data extraction process

For collecting the data, we will be using a python library called the (13). Which is a helpful tool to extract all the review for a particular game. The library requires us to put the app id which is a unique identifier for each game set by the platform to be set as the input and get the Json response as our output. The library also ensures that we are not making too many requests to their server and has a internal cool-down which ensure that we are not violating their user policy.

## 3.3 Data Processing

The platform allows the use om emojis and stickers in the comments. We will remove any such characters from our data and remove comments which are purely emojis. Also, the comments can be very long as well. We will limit the comment length to be not more than 500 words. We believe this should be long enough to understand the sentiment part an also the main issue plaguing the user experience.

## 3.4 Sentiment Analysis

Once we have cleaned our data, we will now perform our sentiment analysis on our data. For this we are using two different libraires, first will be Vader (14) and the other will be using Transformer from Hugging faces(15). Both provides a high level API to perform sentiment analysis. In case of transformer, we will use an additional model DistilBERT model fine-tuned on the SST-2 to check for any differences between them.

## 3.5 Issue Detection

Once we have detected the negative sentiment using the above method, we can move forward with detecting any potential issue with the game. Most of the time we can expect the review to be a general review describing the users experience with the game.

However, on occasion where there is a patch or a content roll out, we can expect a shift in sentiment and potentially higher chances for user to experience issues. This study from (7) also uses LLM model for detecting code vulnerabilities which leverages incontext learinig.

# 4 Design Specification

For sourcing the data we are using the Steams API for making request to their server. The API allow a wide variety of features however we will be utilizing the feature to extract the review from the platform. To do that we will be using a tool called steamreview which is a python tool which perform the exact task which we require. This save us the time to manual implement each call to the server and aslo ensure the calls are limited to not overload their servers. For manipulating the data and performing useful operation we will use pandas data from which is a handy tool for all type of data related operation. For performing the sentiment analysis we will be using VADER and Transformer liabilities which provide high level tool for sentiment analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is particularly effective for analyzing social media, customer reviews, and other text sources where the language is informal. VADER is part of the nltk library in Python and is widely used because of its simplicity, speed, and focus on analyzing sentiment in short text. VADER relies on a pre-built lexicon of words with associated sentiment scores. Each word in the lexicon has a value indicating its positivity, negativity, or neutrality. It is also context aware and considers punctuation. It is also light weight and fast. VADER produces four scores for a given text. Positive, the proportion of text that is positive. Negative, the proportion of text that is negative. Neutral, the proportion of text that is neutral. Compound, a normalized, aggregated sentiment score ranging from -1 (most negative) to +1 (most positive). Transformers library, a popular open-source framework for natural language processing (NLP) tasks. It provides easy access to pretrained transformer models, such as BERT, GPT, and others, allowing developers and researchers to build applications like text classification, sentiment analysis, question answering, and more with minimal setup. It provides several pre-trained model such as DistilBERT, BERT, RoBERTa, etc. They are light weight and fast. For detecting Issue with the review we will using a LLM model such as llama3.2. Llama is a LLM models are capable of generating human-quality text, translating languages, writing different kinds of creative content, and answering your questions in an informative 1 way. For using llama we will be using a powerful, versatile and open source platform called ollama (16). It provides a model which can run locally on our device. Since it is running locally, we can apply this multiple times on our entire data set and there is no need to query the data to a remote server. This makes it faster and usable for our application.

# 5 Implementation

To analyze customer reviews and detect performance issues, the implementation involved several sequential steps, starting from data collection to sentiment analysis and issue detection. The process was carefully designed to ensure accuracy and relevance in identifying negative feedback and performance-related concerns.

## 5.1 Data processing

The data was collected using the steamreviews library, which allowed the extraction of customer reviews for a specific product or service from the Steam platform. The dataset included review text, timestamps, and associated metadata. The focus was on obtaining a comprehensive set of customer feedback to capture a broad range of sentiments and opinions. Once the data was collected, a cleaning process was conducted to prepare it for analysis. The first step in cleaning was to ensure all the review are not more than 500 words. To focus the analysis on a single language, non-English reviews were removed using the langdetect library. This step was essential as the sentiment analysis tools employed are optimized for English text. Any reviews that were either incomplete or irrelevant were also excluded during this stage.

## 5.2 Sentiment Analysis

The cleaned dataset was then subjected to sentiment analysis using two approaches: VADER and a Transformer-based model. VADER, a lexicon-based sentiment analysis tool, was used to classify each review as Positive, Negative, or Neutral based on the compound sentiment score. This approach is particularly effective for analyzing short and informal text like customer reviews. Additionally, a pre-trained Transformer model, specifically a fine-tuned DistilBERT model, was used to classify the sentiment in a more nuanced and context-aware manner. This dual analysis ensured robust sentiment classification, leveraging the strengths of both lexicon-based and deep learning-based techniques.

## 5.3 Detecting Issues

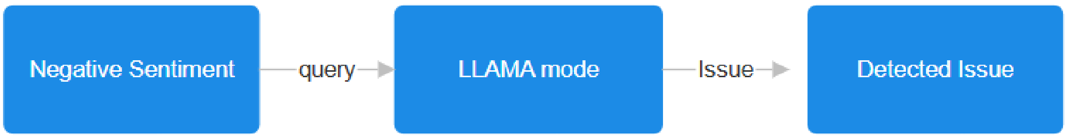


Figure 2: Issue Detection

With the sentiment analysis results, the negative reviews were further analyzed to detect potential performance issues. The Figure 2 show the issue detection process. This step involved scanning the negative reviews for keywords commonly associated with performance problems, such as "lag," "crash," "slow," and "unplayable." By focusing on these specific terms, the system was able to identify and isolate reviews that highlighted technical or functional issues related to the product or service. This will be done using topic modeling and using LLM to detect the issue in the review.

# 6 Evaluation

In this section we will discuss the observation of application of each of our sentiment analysis model and also discuss about the output of the LLM model.

## 6.1 Sentiment Analysis using Vader

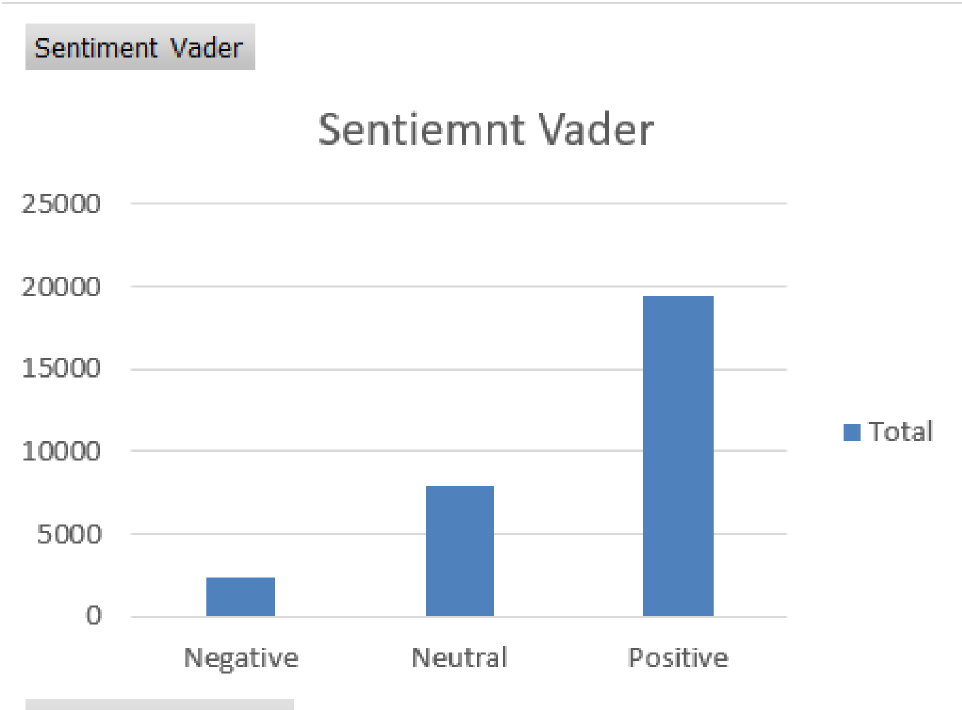


Figure 3: Sentiment count Vader

The above Figure 3 shows the count for sentiment using Vader. Majority of the data is positive and for simplistic purpose we will assume the neutral data to also part of positive response as they are likely to be game reviews.
The first model we will look at is Vader which uses a pre-built lexicon of words, each associated with a sentiment score (positive, negative, or neutral).
The lexicon is tailored for social media and includes common expressions, emojis, and slang. In our case, it will be common for the data to contain slangs and sarcastic data, therefore this should help in those cases.
Since our data is not labeled, we will be evaluating a random sample of our data set for it performance.
We used a random sample of n=84 reviews to manually evaluate the results. We are able to correctly identify 84% of the sentiment correctly. However, evaluating it like this is can be misleading and we will look into this in a more depth.
If we look at the outpost of the sentiment against the word count in the Figure 4, we observe that in case where it is hard to determine anything like in cases where review is less that 3 words, Vader determines them as neutral which is expected. As the word count increases we see a sudden spike in the positive comments when the words are less than 5. In these case we can conclude that VADER is much likely to determine a text as

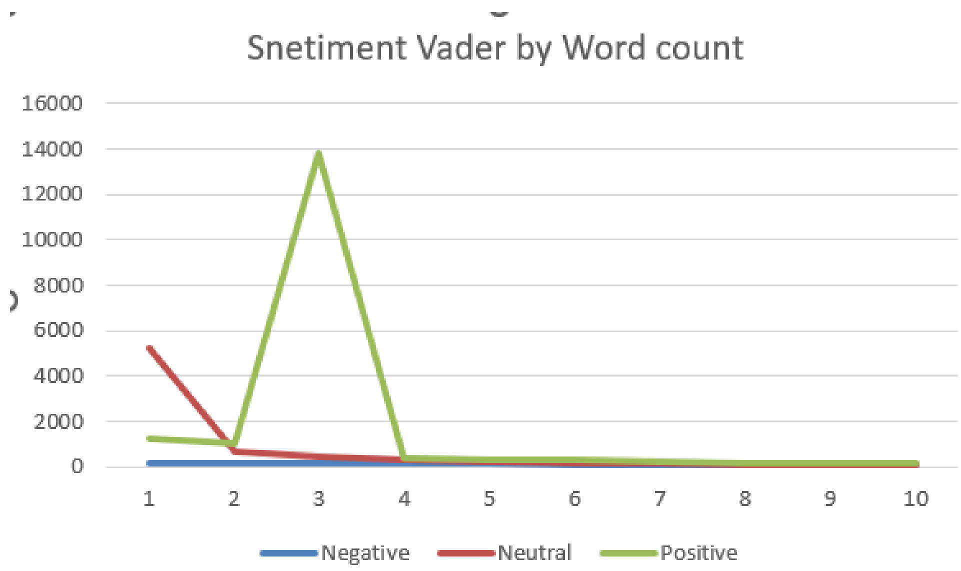


Figure 4: Vader Sentiment by word count (less than 10)

positive in case of less words.

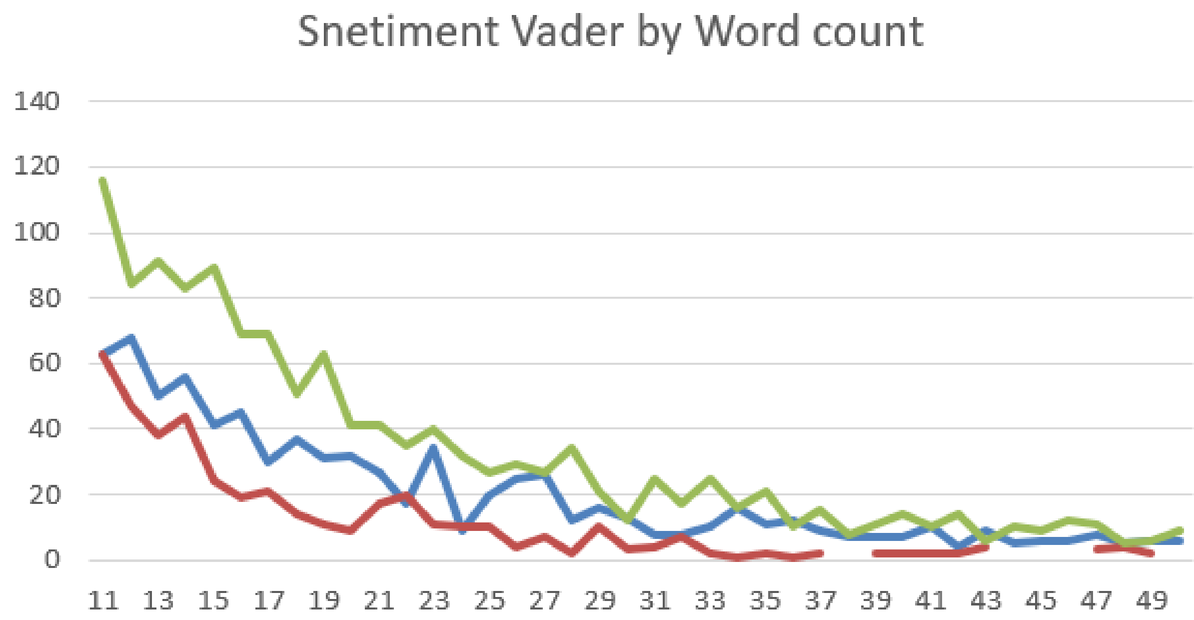


Figure 5: Vader Sentiment by word count (more than 10)

When we have words more than 10 is where we can observe in the figure 5 the negative sentiment well. However, the it more likely to detect the sentiment for longer statements. We will look at a few example where we observe the limitation of VADER.

| review |
| --- |
| I never played Dota 2 before because when I asked my friend if the game mechanics were easy to use, he said he hated that there is no camera lock system like in League of Legends, where the camera follows the champion/hero. But the thing is, it's more fun, even though the camera setting isn't friendly to me. I managed to adjust some settings, which is okay for me. To be honest, it's more fun than League of Legends. I also heard there's no smurfing, which I hate in League, and fewer toxic players. I hope thi |
| Ever felt like commanding a bunch of over-eager heroes on a warpath? Welcome to Dota 2, the game that makes chess look like tic-tac-toe. It's like a fantasy sitcom with extra bonus rage-quits! |
| IF U HAVE A GOOD DAY U NEED TO PLAY THIS GAME CUZ THIS GAME IS SO FKING DOG MAY DOG PLAYER HERE |
| This is a game you play after League's soft handed estrogen gamers have taken a toll on your mental health. Enter DotA 2: Introducing Peruvians on the US East and West server (More foreigners in your home regions than American players just like in real life!) These players though atleast 2X the Testosterone do not know a lick of english nor do they apparently know how to play their hero with 5k games... Point being no matter what MOBA you go to you will always have these terrible NPC players on your team. D |
| all of a sudden keeps on crashing on main menu -_- disabled vulkan tried to run on vulkan. Tried to put -safe, default etc. simultaneously and still crashing! uninstall.. |
| I played League of Legends for over 12 years so I had an easier time picking this up than you might if you've never played a MOBA before, but I highly recommend trying this out. It's more in-depth than League so it feels more worth putting time into, it has striking graphics, and you don't have to worry about grinding for months to unlock the roster because they're all free! |

Figure 6: Vader Error

In the figure 6 are the few review where we see that Vader is considering it to be positive while the sentiment is mostly negative. This patter is a repeat occurrence especially if the review are sarcastic, funny or if is dark humor. Comments like "best way to get hypertension" is picked up as a positive, however words like best in this case if not positive. On the other hand, negative sentiments are being handled much better by VADER. When we specifically looked at a sample of negative comments (initial random sample were mostly positive), they were all flagged as negative. This will be mostly due to lack of positive sounding words in the comment helps in reliably flagging them as negative.

## 6.2 Sentiment Analysis using Transformer (BERT)

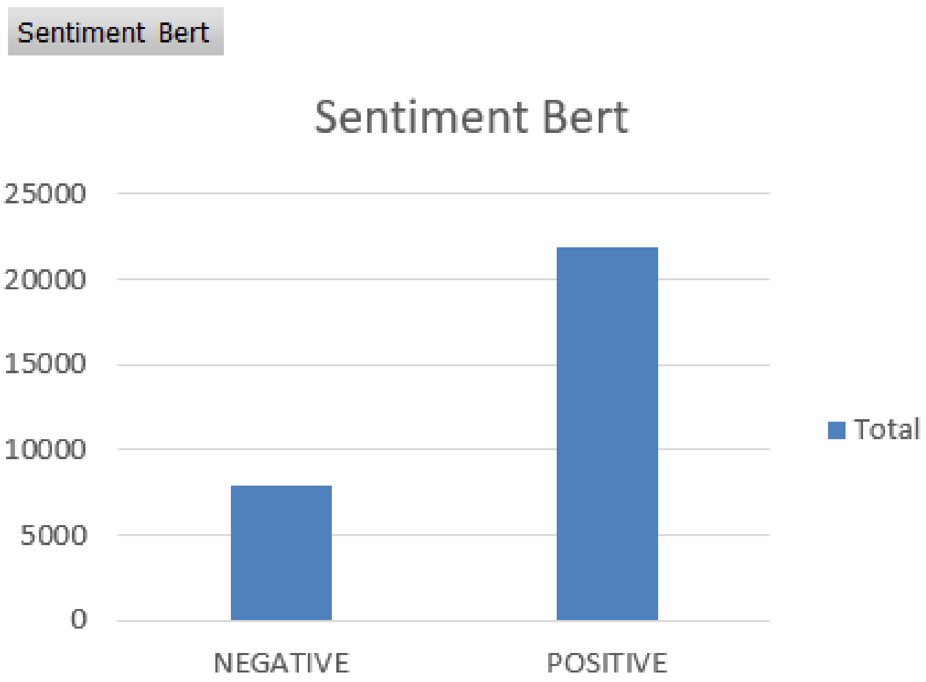


Figure 7: Sentiment count BERT

In the figure 7 we can have a look at the sentiment count using BERT. When compared to Vader, we can observe that the count for the negative classes is much higher for

BERT.
The second sentiment analysis model which we will be looking at is using transformer library by Hugging faces. It uses BERT and DistilBERT for its sentiment analysis. BERT is a bidirectional model trained on large text corpora for natural language understanding task. DistilBERT is a light weight version of BERT optimized for faster inference. In the

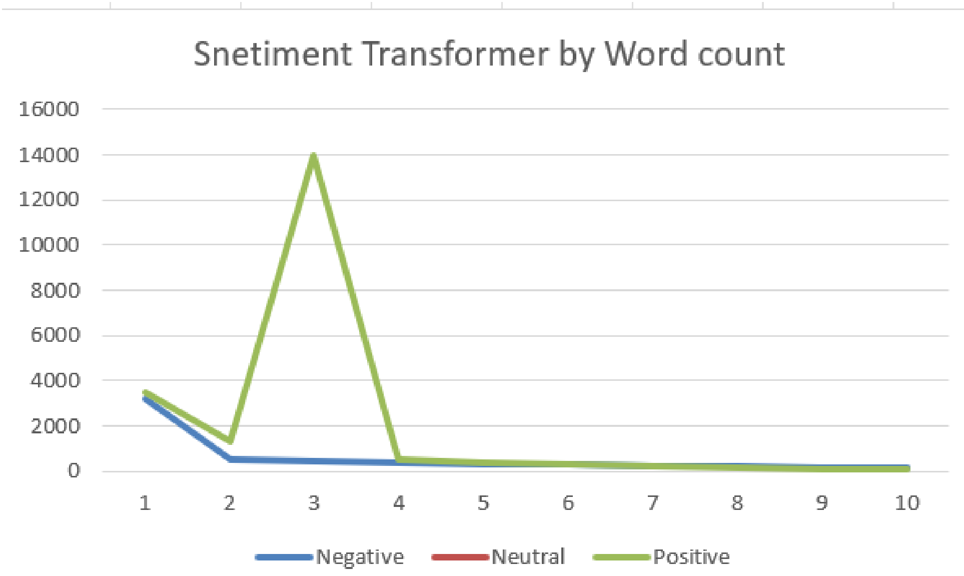


Figure 8: BERT by word count (less than 10)

figure 8 we observe a similar pattern as we observe in VADER. However on close inspection BERT is able to identify negative sentiment in shorter comment better than VADER.

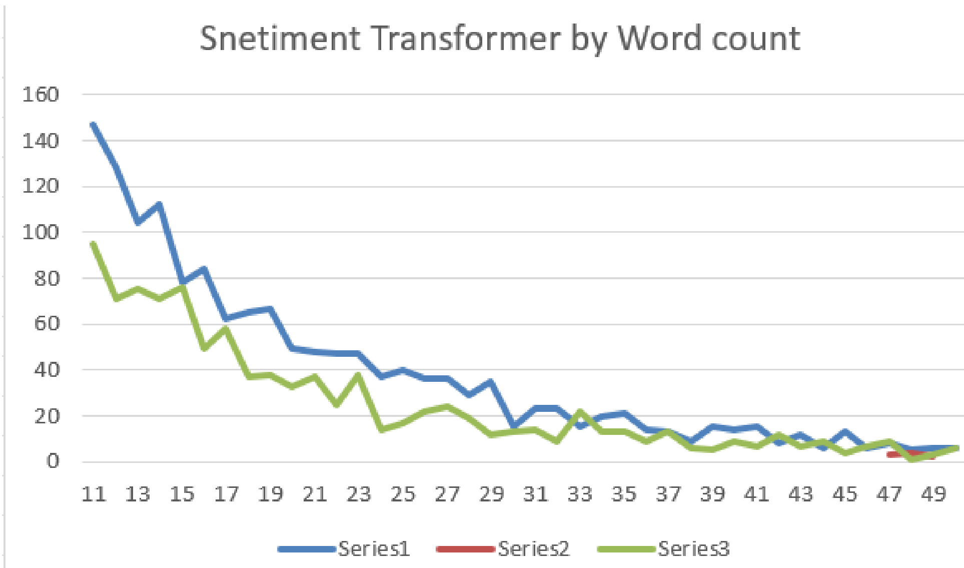


Figure 9: BERT by word count (more than 10)

For higher word count we observer in the figure 9 that the negative and positive sentiment counts are much closer compared to VADER. When we take a look at the above sample and manually evaluate the results we can observe that BERT was able to detect the negative sentiment well in longer sentences. Also, some of the sarcastic comment were also getting picked up well which are negative but they might appear positive due to the use of positive words. The bidirectional approach used by BERT might be helping in resolving the sarcasm batter than VADER.

## 6.3 Issue Detection using LLM (Llama3.2)

For detecting the issue with the game we are going to utilize LLM model such as Llama3.2 for detection. Llama is an open source AI model developed by Meta. It is capable of lightweight text only models and vision capabilities. The text only models has 1 billion and 3 billion parameters and in this project we are using model with 1 billion parameter. Due to its light weight design, we can easily use this to query and identify if any issue is highlighted by the user.
For detecting the issue we are working on the negative sentiments only as we are more likely to associate issue with a negative sentiment. We will pass each of the review to the LLM model and ask if it can detect issue in the review. Below mentioned is the sample query for a comment "crashed on my 3rd game lol"

[crashes, match making, fps drop, performance, no-issue] out of these common issues in games, which issue can you detect in the following comment "crashed on my 3rd game lol". Reply only the issues. If multiple issues are there, reply in a single sentence separating with commas.

The LLM responds with a catagory of issue and a confidence score for each of the comments. We will now look at the count analysis of the output which get from the model.

| Issue Type | Count of sentiment |
|---|---|
| Crash | 22336 |
| FPS Drop | 605 |
| No Issue | 3204 |
| Performance Issue | 3593 |
| **Grand Total** | **29738** |

Figure 10: Count of Issue

In the Figure 10the counts which we get is a bit surprising. The count for Crash as the issue is much high compared to other type of issue and no issue. Since we are working with review data, we were expecting to find No Issue to be the most dominant class.
In order to investigate this, we look at the confidence score as well which indicate the confidence level of the issue which is detected.
If we look at the averages in the Figure 11 we can see that the average confidence for crashes and fps drops are much lower than No issue and performance issue. This indicate that the crash as an issue is much less reliable compared to other issues.
We will now try to address this by increasing the threshold for detecting the issue. We

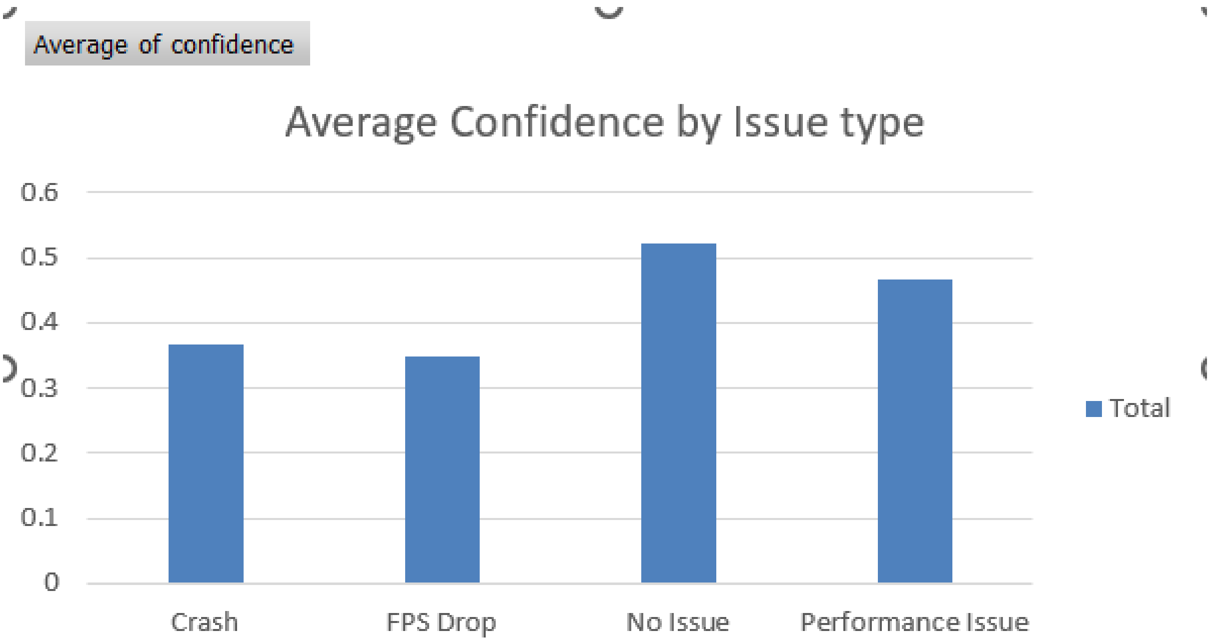


Figure 11: Average confidence by Issue Type

will perform this by increasing the threshold of confidence level. We will set the Issue as no issue if the confidence score is less than 0.7 and we will check for the change in count.

| Issue type | Count of category |
|---|---|
| Crash | 84 |
| FPS Drop | 3 |
| No Issue | 7608 |
| Performance Issue | 208 |
| **Grand Total** | **7903** |

Figure 12: Count of Issue Confidence 70%

With the confidence score raised to 70%, we can see in the figure 12 that the issue category has now dropped significantly. This is now more inline with what we expected from the data. On further manual inspection, we an observe that nearly 39% of the issue were identified correctly. The number is pretty low however we have observed a few patterns using which we can further improve our result.
When we observe the correctly predicted issue, we can see that nearly 75% of the performance issue were detected correctly. Crashes being only 50% accurate. We also notice that the most of the miss occur on instances where review are complaining about other people behavior. The model detect such comments as performance issue while detecting it. This may be due to the wording used while describing the player behavior which sound similar to lack of performance of the game. For example, the player commenting about his teammates being noncompetitive or lack of skill, LLM picks this up as lack of

performance of the game incited of player. We observe a large instance of misrepresentation of the issue class. However, nearly all instances of issues were detected correctly from our manual sample.
By optimizing the query to our LLM, we can improve the result by using issue specific query to our LLM. Instead of categorizing all the issue at once, we can evaluate one issue at a time. This can help in scenario where there are no issue present in the review. Other way we can improve is to use a larger model. Since the system which we are using has very limited resource, this option can be explored when better hardware is available.

# 7 Conclusion and Future Work

By implementing Sentiment Analysis model like BERT and using LLM model such as llama, we were able to successfully identity issue with the game. The both the libraries are able to identify the sentiments however, due to the behavior of gamers these libraries fell short in some cases. In review where the comment is sarcastic were not picked up well by VADER while BERT did better for such instances. VADER was much accurate for longer review. Both the model performed well in case of negative sentient which was the primary focus of our research. BERT had a slight advantage at detecting the negative sentiment because it was able to pick up sarcasm well. For future work, we can try to explore a sentiment analysis model which is better suited for sarcastic environment. Most of the model utilize a more general data sources such as social medial platform such as twitter, facbook and reddit where people are more likely to present themselves well and in polite manner. For gaming environment where toxic behavior is much common occurrence a model tailed for such environment will help us better.
This model can have other application in the gaming space for content and communication monitoring where game specific nuance can be preserved while extremely negative behavior such as racism and threats can be punished.
In the second part, explored the use of LLM model such as llama3.2 for detecting the game issue in the negative sentiment. We pass the comment to the llama model and ask if the comment highlight a predefined set of issue. Once we have the issue resolved by the model, we noticed the data need to be adjusted since a lot of the comment without the issue were highlighted as an issue. By adjusting the threshold we were able to nearly identify 40% of the comments with issue correctly.
For future work, we can develop a tailored sentient model for sarcastic behavior detection for better application in gaming environment. The issue detection mechanism can be modified and use in different platform such as social media, review websites and forums for tracking the negative sentiment across a wider range. This will further help in picking up issue if the users are much more comfortable to raise their voice on social media rather than the review platform of the game distribution platform. The fault detection mechanism can be more specific to the issue and additional consideration can be given to player behavior on the platform. This will help in mitigating the impact of player behavior getting flagged as performance issue and help focus more on the actual issue.
A active monitoring system can be developed for continuous monitor during the roll out period of a new release or update. Also, the LLM can also be used for content moderation to keep the platform free from hate speach.
In conclusion the research demonstrated a successful application of ML model for senti-

ment detection and LLM model for identifying potential issue with the services associated wit the negative sentiment of the users.

# References


[1] R. Y. Kim, "Using online reviews for customer sentiment analysis," *IEEE Engineering Management Review*, vol. 49, no. 4, pp. 162–168, 2021.

[2] M. Rawat, S. E. Hosseini, and S. Pervez, "Sentiment analysis for assessing customer satisfaction in chatbot service encounters," in *2023 16th International Conference on Developments in eSystems Engineering (DeSE)*, 2023, pp. 105–109.

[3] B. S. Prasad and C. M. V. S. Akana, "Polarity sentiment-based intelligent chat bot for judicious customer service escalation," in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 2021, pp. 1–6.

[4] S. Halima Banu and S. Chitrakala, "Trending topic analysis using novel sub topic detection model," in *2016 2nd International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, 2016, pp. 157–161.

[5] H. Li and L. Shan, "Llm-based vulnerability detection," in *2023 International Conference on Human-Centered Cognitive Systems (HCCS)*, 2023, pp. 1–4.

[6] Y. Wu, M. Wen, Z. Yu, X. Guo, and H. Jin, "Effective vulnerable function identification based on cve description empowered by large language models," in *2024 39th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2024, pp. 393–405.

[7] Z. Liu, Q. Liao, W. Gu, and C. Gao, "Software vulnerability detection with gpt and in-context learning," in *2023 8th International Conference on Data Science in Cyberspace (DSC)*, 2023, pp. 229–236.

[8] A. A. Hossain, M. K. PK, J. Zhang, and F. Amsaad, "Malicious code detection using llm," in *NAECON 2024 - IEEE National Aerospace and Electronics Conference*, 2024, pp. 414–416.

[9] T.-J. Chang, L. H.-M. Lin, and R. T.-H. Tsai, "Conversational product recommendation using llm," in *2024 IEEE 4th International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*, 2024, pp. 340–343.

[10] Valve, "Steam store." [Online]. Available: https://store.steampowered.com/

[11] ——, "Steam subscriber agreement." [Online]. Available: https://store.steampowered.com/subscriber_agreement/

[12] ——, "Terms of service." [Online]. Available: https://store.steampowered.com/eula/471710_eula_0

[13] steamreviews, "steamreviews." [Online]. Available: https://github.com/woctezuma/download-steam-reviews

[14] VADER, “Vader.” [Online]. Available: https://github.com/cjhutto/vaderSentiment

[15] cjhutto, “Bert.” [Online]. Available: https://huggingface.co/docs/transformers/en/model_doc/bert

[16] ollama, “ollama.” [Online]. Available: https://ollama.com/