What Causes Wrong Sentiment Classifications of Game Reviews?

Markos Viggiato, Dayi Lin, Abram Hindle, and Cor-Paul Bezemer

***Abstract*—Sentiment analysis is a popular technique to identify the sentiment of a piece of text. Several different domains have been targeted by sentiment analysis research, such as Twitter, movie reviews, and mobile app reviews. Although several techniques have been proposed, the performance of current sentiment analysis techniques is still far from acceptable, mainly when applied in domains on which they were not trained. In addition, the causes of wrong classifications are not clear. In this paper, we study how sentiment analysis performs on game reviews. We first report the results of a large scale empirical study on the performance of widely-used sentiment classifiers on game reviews. Then, we investigate the root causes for the wrong classifications and quantify the impact of each cause on the overall performance. We study three existing classifiers: `Stanford CoreNLP`, `NLTK`, and `SentiStrength`. Our results show that most classifiers do not perform well on game reviews, with the best one being `NLTK` (with an AUC of 0.70). We also identified four main causes for wrong classifications, such as reviews that point out advantages and disadvantages of the game, which might confuse the classifier. The identified causes are not trivial to be resolved and we call upon sentiment analysis and game researchers and developers to prioritize a research agenda that investigates how the performance of sentiment analysis of game reviews can be improved, for instance by developing techniques that can automatically deal with specific game-related issues of reviews (e.g., reviews with advantages and disadvantages). Finally, we show that training sentiment classifiers on reviews that are stratified by the game genre is effective.**

***Index Terms*—Natural language processing, Sentiment analysis, Computer games, Steam.**

I. Introduction

Sentiment analysis is a widely adopted Natural Language Processing (NLP) technique to obtain the sentiment (expression of positive or negative feeling) from text data [29, 39]. This technique consists of identifying the sentiment that is present in a piece of text (words, sentences, or entire documents), which corresponds, in its most basic form, to finding whether the text has a positive, neutral, or negative sentiment [29]. Sentiment analysis is a research topic that has gained attention and has presented improvements [15, 49, 56], being developed and applied in several different domains, such as Twitter tweets [3, 5, 8], movie reviews [49], customer reviews of mobile applications [19, 40], video game reviews [50, 52], and various aspects of software development [24, 27, 28, 38, 44]. Sentiment analysis is valuable for game developers because it allows them to capture how players feel about the game and learn about previous games' success or failure factors [50]. This knowledge can help game developers improve their game development processes and guide them in future releases of their game (e.g., by focusing on features that users are more positive about).

Several studies have been published on sentiment analysis with the purpose of developing new techniques, improving current techniques, or applying current techniques and classifiers to existing datasets [11, 22, 24, 27, 28, 29, 50, 52]. However, the performance of such techniques is still far from acceptable, mainly when off-the-shelf sentiment analysis classifiers are applied out of domain, i.e., a classifier is trained in one domain and applied in a different domain without any configuration or adjustment. Normally, sentiment analysis techniques must be adapted to the target domain. For instance, Thompson et al. [52] adapted a sentiment analysis technique that was initially designed for movie reviews to be used in video game chat messages. Despite the low performance of sentiment analysis, no study has investigated the reason(s) for the low performance.

In this study, we investigate how different sentiment classifiers perform on game review data. Game reviews from Steam differ from other types of data to which sentiment analysis is normally applied. Game reviews contain a more complex text structure and generally discuss several aspects of the game, such as the game's storyline, graphics, audio and controls [58]. Texts from micro-blogging and social media (e.g., Twitter) are usually very short [17, 36]. In addition, such texts are broader in scope since they are not necessarily reviewing a game. Prior work [31] also showed that game reviews are different from mobile app reviews in several aspects. For instance, game reviews contain game-specific terminology, which is a challenge for language processing tools.

Although the diversity of game reviews makes them a rich source of data, it also poses challenges to NLP techniques, such as sentiment analysis. For instance, players may mention the graphical aspects and the storyline of the game in the same review [50]. The two pieces of text corresponding to such aspects may have different sentiments, which could confuse the sentiment classifier when making a classification of the overall sentiment of the review.

By applying sentiment classifiers on game reviews, we are able to report the sentiment classification performance and identify cases where sentiment analysis fails. For instance, the following review is an example of a difficult classification

Markos Viggiato and Cor-Paul Bezemer are with the Analytics of Software, GAmes And Repository Data (ASGAARD) Lab, University of Alberta, Edmonton, AB, Canada. Email: viggiato@ualberta.ca, bezemer@ualberta.ca.

Dayi Lin is with the Centre for Software Excellence, Huawei, Canada. Email: dayi.lin@huawei.com. This work is not related to his role at Huawei.

Abram Hindle is with the Department of Computing Science. University of Alberta, Edmonton, AB, Canada. Email: abram.hindle@ualberta.ca

task for current sentiment classifiers: "*Very nice programmed bugs*". The reviewer makes references to a positive word ("nice"), with a stronger intensity due to the use of an adverb ("very"), which might lead the classifiers to classify this instance as positive. However, the overall sentiment of this review should be negative as the reviewer is being sarcastic (the reviewer is pointing out that the game contains bugs). A deeper investigation of wrong classifications (failing cases) allows us to find problematic text patterns for sentiment classifiers and provide insights for game developers about how to improve the performance of sentiment analysis.

In this paper, we first report the results of a large-scale empirical study on the performance of sentiment analysis on 12 million game reviews. Our goals are (1) to investigate how existing sentiment classifiers perform on game reviews, (2) identify which factors impact the performance and (3) quantify the impact of such factors. Note that we do not aim to propose a new sentiment classification technique. Instead, we investigate reasons for wrong classifications of existing classifiers. We studied three widely-used and computationally accessible sentiment classifiers [24, 29]: `Stanford CoreNLP` [49], `NLTK` [7], and `SentiStrength` [51]. The selected classifiers adopt different approaches to classify the text, such as rule and machine learning-based approaches, which gives more confidence to our study and makes the results more generalizable.

We evaluated these classifiers on all the game reviews collected from the Steam platform up to 2016. We then selected the reviews of which all classifiers misclassified the sentiment. We manually analyzed a representative and statistically significant sample of 382 of these reviews to understand which factors might be causing wrong classifications. Finally, we performed a series of experiments to quantify the impact of each identified factor on the performance of sentiment analysis on game reviews. We address the following three research questions:

RQ1: How do sentiment analysis classifiers perform on game reviews?
Investigating the performance of sentiment analysis on game reviews is the first step to understand how current sentiment analysis classifiers work on game reviews and whether they are suitable for this task on such data. We found that sentiment analysis classifiers do not perform well on game review data, with AUC values ranging from 0.53 (`Stanford CoreNLP`), which is slightly better than random guessing, up to 0.70 (`NLTK`).

RQ2: What are the root causes for wrong classifications?
Identifying the causes for wrong classifications contributes to obtain important insights about how to improve existing sentiment analysis for game reviews. We found several causes which mislead the classifiers, such as reviews that make comparisons to games other than the game under review, reviews with negative terminology (e.g., reviews that use the word "kill") which does not necessarily mean the content has a negative sentiment, and reviews with sarcasm.

RQ3: To what extent do the identified root causes impact the performance of sentiment analysis?
Quantifying the impact of each identified root cause to the performance of sentiment analysis is important to support game developers with the prioritization of causes to be resolved and a research agenda to address such issues. We found that reviews which point out advantages and disadvantages of the game have the highest negative impact on the performance of sentiment analysis, followed by reviews with game comparisons. In addition, we deepened our investigation and showed that training sentiment classifiers on reviews stratified by the game genre is effective.

Our study makes three major contributions:

- We evaluate the performance of widely-adopted sentiment analysis classifiers on game reviews from the Steam platform.
- We identify a set of root causes that can explain the wrong classifications of sentiment analysis classifiers on game reviews.
- We quantify the impact of each identified cause for wrong classifications on game reviews and provide a research agenda for addressing these causes.
- We provide access to the data[1] (URLs of game reviews from Steam with the sentiment classification provided by all three classifiers).

The remainder of this paper is organized as follows. Section II provides a background on sentiment analysis classification techniques. Section III discusses related work and Section IV presents the proposed research methodology. Section V discusses the pre-study. In Sections VI, VII, and VIII, we discuss the results, while in Section IX we present our recommendations on how to perform sentiment analysis on game reviews. Finally, Section X concludes our paper.

II. Sentiment Analysis

In this section, we present an overview of the main sentiment analysis techniques along with the most used classifiers that adopt these techniques. In this work, we use 'technique' to refer to the method adopted for the sentiment classification and 'classifier' (which can also be understood as 'tool' or 'framework') to refer to an implementation of a technique (i.e., an actual instance of the technique). Next, we discuss each technique and the representative classifier(s) we chose for our work. For this study, we focus on popular, open source and free-to-use sentiment analysis classifiers.

Sentiment analysis techniques are responsible for identifying the sentiment present in a piece of text, which can be either positive, neutral, or negative [29, 39]. Table I presents an overview of the main sentiment analysis techniques and classifiers which have been proposed in prior studies. This is not an exhaustive list of sentiment classifiers and it comprehends the most reported classifiers in prior studies. The grouping of classifiers under a specific technique category was done based on the method the classifier uses. Classifier names in bold refer to the ones studied in this work. The last column

[1] https://github.com/asgaardlab/sentiment-analysis-Steam_reviews

shows the type of data on which the classifier was originally trained. Next, we detail each technique and the corresponding classifier(s) we chose to use in our study.

1) Machine Learning-based Techniques: Machine learning-based classifiers leverage machine learning algorithms, such as Support Vector Machines, Naïve Bayes, and Neural Networks. Examples of classifiers that adopt this technique are `NLTK` [7], `Stanford CoreNLP` [49], and `Senti4SD` [10]. For our study, we selected `NLTK` and `Stanford CoreNLP`, which are open source, free to use and very popular [24, 28].

`NLTK` is part of a larger NLP package that provides many other functions.[2] Regarding sentiment analysis, `NLTK` uses a bag of words model. In order to apply `NLTK`, we can adopt two different approaches: train a Naïve Bayes classifier on our data and apply the built model (as we did) or use the VADER (Valence Aware Dictionary and sEntiment Reasoner) model, which was trained on social media texts, such as micro-blogs [29]. The latter approach provides four scores for each sentence: *compound* (varies from very negative to very positive as indicated by a score in the range [-1, +1]), *negative* (probability of being negative), *neutral* (probability of being neutral), and *positive* (probability of being positive). In the former approach, we train a Naïve Bayes model to classify each review (it provides the probability of being positive). Figure 1a presents examples of reviews classified by the machine learning version of `NLTK`. As we can see, the positive example is correctly classified. However, `NLTK` is not able to capture the negative sentiment of the sentence "*I am so happy the game keeps freezing*", which contains sarcasm.

`Stanford CoreNLP` was developed by the Stanford Natural Language Processing Group[3] at Stanford University. The authors propose a model called Recursive Neural Tensor Network, of which the implementation is based on a Recurrent Neural Network (RNN). The technique consists of parsing the text to be classified into a set of sentences and performing a grammatical analysis to capture the compositional semantics of each sentence [27, 29, 49]. Then, a score between '0' and '4' is assigned for each sentence, in which '0' means a *very negative* sentiment, '1' means *negative*, '2' refers to a *neutral* sentiment, and '3' and '4' refer to *positive* and *very positive* sentiments, respectively. To classify a game review (composed of more than one sentence), we adopt the following approach [28]: -2*(#0) - 1*(#1) + 1*(#3) + 2*(#4), in which #0 refers to the number of sentences with score 0, and so on. If the resulting score is above zero, the review sentiment is positive; if it is below zero, the review sentiment is negative; otherwise, the review sentiment is neutral.

In Figure 2, we can see an example of how the sentence "*I killed the evil enemy and I won*", which is positive, is wrongly classified using `Stanford CoreNLP` (the root node indicates it is a negative sentence). As we can observe, each node in the parse tree is assigned a score (from *negative* to *neutral* to *positive*) and the final sentiment is obtained via the compositional structure of the tree. We can see that different nodes are assigned different sentiments (relative to the partial sentence composed up to that node) and the sentiment contained in the root is supposed to capture the overall sentiment of the full sentence, which is opposed to only inspecting the sentiment of each word individually and summating the scores. This example was obtained from the `Stanford CoreNLP` sentiment analysis website with the live demo tool.[4]

2) Rule-based Techniques: Rule-based classifiers are based on a predefined list of words along with their sentiment score. The piece of text is split into words, and the scores of each word are composed into a final score for the entire piece. Examples of rule-based classifiers are `SentiStrength` [51], `SentiStrength-SE` [24], and `EmoTxt` [9]. In this work, we apply `SentiStrength`, which is of one of the most used sentiment classifiers across different domains, such as social media (e.g., Twitter) [2], and movie reviews [35].

`SentiStrength` is a rule-based classifier to classify sentences into sentiments based on a word bank in which each word has a sentiment score associated with it (this is also called lexical analysis). This classifier is based on a model trained on the MySpace social media network [29]. The document under analysis must be tokenized into sentences, which are assigned two scores based on the summation of each word's score: a positive strength score (how positive is the text) that ranges from 1 (*not positive*) to 5 (*very positive*), and a negative strength score (how negative is the text) that ranges from -1 (*not negative*) to -5 (*very negative*).

Figure 1b presents some examples of classifications made by `SentiStrength`. We can see that the classifier's approach of getting the sentiment score of each word individually and summating the scores does not work for some cases. The classifier is not able to capture the negative sentiment in the sentence "*I am so happy the game keeps freezing*", which is sarcastic. This possibly happens due to the presence of the word "happy", which is a positive word and misleads the tool to classify the whole sentence as positive. In addition, `SentiStrength` is not able to capture the neutral sentiment in the sentence "*The game was nothing special*", possibly due to the presence of the positive word "special", which is positive.

III. Related Work

In this section, we describe prior work on the application of sentiment analysis on game data and on other types of data. We also discuss empirical studies on game reviews. Note that, in our work, we do not aim at proposing a new sentiment analysis technique. Instead, we investigate the performance of existing sentiment classifier on game reviews and reveal the root causes for wrong classifications. We focus on popular sentiment classifiers, which are not computationally expensive (e.g., deep learning-based classifiers).

Sentiment Analysis on Game Data and Reviews. Thompson et al. [52] studied how to extend a lexicon-based sentiment analysis technique for the purpose of analyzing StarCraft 2 player chat messages. The authors updated the entries to the

[2]https://www.nltk.org/

[3]https://nlp.stanford.edu/

[4]http://nlp.stanford.edu:8080/sentiment/rntnDemo.html [Accessed online: March 11th, 2020]

TABLE I: Sentiment analysis techniques, corresponding classifiers and default training dataset.

Technique	Classifier	Default training dataset	Used by
Machine learning	**NLTK*[7]**	Micro-blog texts	[27], [28], [29], [41], [34]
	Stanford CoreNLP [49]	Movie reviews	[27], [28], [42], [33], [55]
	IBM Alchemy**	—	[27], [28], [48], [6]
	Senti4SD [10]	Stack Overflow posts	[10], [25]
Rule-based	**SentiStrength [51]**	MySpace	[20] [22], [21], [27], [28]
	SentiStrength-SE [24]	JIRA	[24], [25]
	EmoText [9]	Stack Overflow, JIRA	[9], [25], [37]

* Note that we use the machine learning version of NLTK instead of its VADER version (which uses a rule-based approach).
** IBM Alchemy is available as a service within IBM Watson at https://www.ibm.com/watson/services/tone-analyzer/.

True sentiment	NLTK classification	Sentence
Negative	Positive ✗	I am so happy the game keeps freezing
Positive	Positive ✓	Was blown away by some of the developments in the story in this game, not gonna spoil but def a must try

(a) Example of classifications made by NLTK.

True sentiment	SentiStrength classification	Sentence	Positive strength	Negative strength
Negative	Positive ✗	I am so happy the game keeps freezing	**2**	-1
Neutral	Positive ✗	The game was nothing special	**2**	-1
Positive	Negative ✗	Was blown away by some of the developments in the story in this game, not gonna spoil but def a must try	**1**	-2

(b) Example of classifications made by SentiStrength.

Fig. 1: Examples of sentiment classifications.

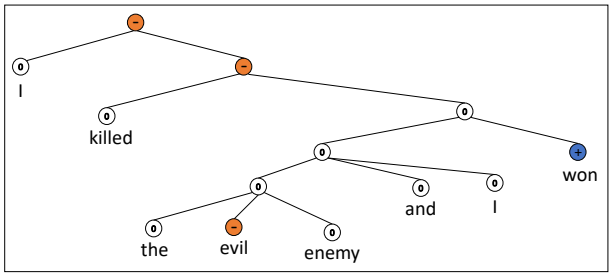


Fig. 2: Example of the Recursive Neural Tensor Network predicting the sentiment in a sentence.

word dictionary and tailored it to the gaming context. The approach was able to classify sentiment and identify toxicity of instant messages across 1,000 games. The best fitting model outperformed the baseline (which predicts that every message has a positive sentiment) for the sentiment classification. The authors also performed a niche analysis, which showed that the model performances remained relatively stable across regions, leagues, and different message lengths. Stråat and Verhagen [50] investigated user attitudes regarding previously released video games. The authors performed a manual aspect-based sentiment analysis on all user reviews from two game franchises: the PC-version of three games from the Dragon Age franchise and the three first games from the Mass Effect franchise. The data was collected from the Metacritic platform. The paper showed that the rating of a user review highly correlates with the sentiment of the aspect in question, in the case of a large enough data set. Zagal et al. [59] studied 397,759 game reviews to identify the sentiment of 723 adjectives used in the context of video games. The authors found that some words which are generally used with a negative (or positive) connotation have a positive (or negative) connotation in the game domain. Finally, Chiu et al. [13], Raison et al. [43], Wijayanto and Khodra [53], and Yauris and Khodra [57] analyzed the sentiment about specific aspects of the game (such as graphics and storyline) in reviews. For example, take the following sentence: "*An okay game overall, good story with very bad graphics*". The player has a positive feeling about the game's storyline, but a negative feeling about the game's graphics. An aspect-based sentiment analysis would compute a different sentiment score for each mentioned aspect.

The aforementioned leveraged different approaches (e.g., lexicon-based and aspect-based) to perform sentiment analysis on different types of data. In contrast, we evaluate existing sentiment analysis techniques on game reviews, identify the causes for misclassifications, and quantify the impact of those causes in the performance of the classifiers.

Sentiment Analysis on Other Types of Data. Agarwal et al. [1] built different types of models (a feature based model and a tree kernel based model) to perform two classification tasks using Twitter data: a binary task to classify tweets into positive and negative classes; and a 3-way task to classify tweets into positive, negative, and neutral classes. The authors showed that both models outperform the state-of-the-art approach by

then, which consisted of a unigram model. The proposed models presented a gain of 4% in performance in comparison to the baseline. Saif et al. [46] also used Twitter-related data to build sentiment analysis models. The authors added semantic features into the three different training datasets: a general Stanford Twitter Sentiment (STS) dataset, a dataset on the Obama-McCain Debate (OMD), and one on Health Care Reform (HCR). The results showed that combining semantic features with word unigrams outperforms the baseline (only unigrams) for all datasets. On average, the authors increased the accuracy by 6.47%.

Lin and He [30] proposed a probabilistic modeling framework based on Latent Dirichlet Allocation (LDA) to detect sentiment and topic at the same time from a piece of text. The authors evaluated the model on a movie review dataset and they only consider two classes: positive and negative. The results showed that the proposed approach obtained an accuracy of 84.60%, outperforming some state-of-the-art approaches. Guzman and Maalej [20] proposed an automated approach to analyze mobile app reviews. The authors used the `NLTK` classifier to identify fine-grained app features in the user reviews. They obtained the sentiment of these features and used topic analysis to group them into higher-level groups. The authors used 7 apps from the Apple App Store and Google Play Store and their approach presented a precision of 59% and a recall of 51%. Rigby and Hassan [44] used a psychometrically-based linguistic analysis tool called Linguistic Inquiry and Word Count (LIWC) to examine the Apache httpd server developer mailing list. The authors assessed the personality of four top developers, including positive and negative emotions present in the mailing list. Among the results, the authors found out that the two developers that were responsible for two major Apache releases had similar personalities, which were different from other developers on the traits of extroversion and openness. Bazelli et al. [4] analyzed StackOverflow posts to identify and compare developers' personality types. They also used the LIWC tool. The results show that, compared to medium and low reputed users, top reputed post's authors are more extroverted, indicating the presence of social and positive LIWC measures as well as the absence of tentative and negative emotional measures. In addition, authors of up voted posts present less negative emotions than authors of down voted posts.

The aforementioned works used data from three different sources: Twitter, movie reviews, and mobile app reviews. On the other hand, we focus on game reviews from a digital distribution platform (Steam).

Studies on Game Reviews. Zagal et al. [58] analyzed and characterized game reviews from different websites. The authors used open coding to come up with the topics present in the reviews. Their findings show that game reviews are rich and varied in terms of themes and topics covered. For instance, players might post descriptions of the game under review, their personal experience, advice to other players who read the review, and suggestions for game improvements. Zagal and Tomuro [60] performed a study on a large body of user-provided game reviews aiming at comparing the characteristics of the reviews across two different cultures. The authors collected reviews from *Famitsu* and *Game World* (Japanese gaming websites) and from *Gamespot* and *Metacritic* (US gaming websites). Among the findings, the authors mention that American players value the replay of a game, while Japanese players are more strict towards bugs. The works mentioned above studied the characteristics of game reviews and what are the differences between reviews from different cultures. Differently, on our work, we use game reviews for the purpose of evaluating existing sentiment classifiers and come up with the causes for wrong classifications.

As we can see, all the aforementioned works proposed new sentiment analysis models and explored the characteristics of game reviews with regard to several different aspects. However, we still lack clarification regarding the performance of existing sentiment classifiers on game reviews, which game review text characteristics impact the performance of sentiment analysis and to what extent they impact it. In our study, we perform a large-scale study with more than 12 million reviews from Steam to evaluate existing sentiment classifiers and reveal text characteristics which are problematic for these classifiers.

IV. Methodology

In this section, we detail the methodology that is used in our study to evaluate existing sentiment classifiers on game reviews from Steam and identify the root causes for wrong classifications. Figure 3 presents a complete overview of our methodology, which is detailed next.

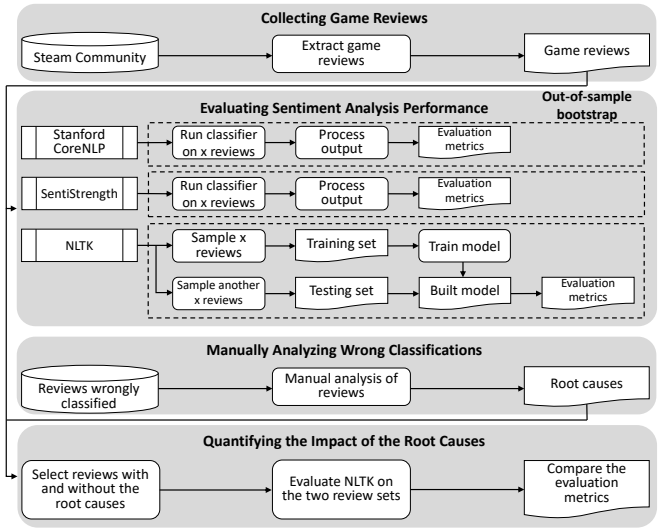


Fig. 3: Study methodology overview.

A. Collecting Game Reviews

We collected the reviews of all 8,025 games that were available in the Steam Store on March 7th, 2016 using a customized crawler. We removed games that had less than 25 reviews from our initial dataset to reduce a possible bias in our results due to a small number of reviews (e.g., because a large portion of those reviews were posted by friends of the developers). In total, we collected reviews of 6,224 games.

We extracted all the reviews for each game from the Steam Community and ended up with a total of 12,338,364 reviews across all supported natural languages. Steam provides a filter for the language of reviews for a game. We crawled the reviews in each language separately using this filter, to identify the language of each review. Most reviews are written in English (6,850,130), but there are also reviews in Russian (1,789,979), German (525,548), Spanish (469,582), Portuguese (441,145), and French (396,057), since the classifiers can handle several different languages. Besides the review itself, we also collected other available data: the recommendation flag (i.e., whether the reviewer recommended the game or not), early access status (i.e., whether a game is in the early access stage or not), the number of playing hours, the author ID, the date when the review was posted, helpful count, not helpful count, funny count, and the URL of the review.

Note that our data consists of Steam game reviews, which is different from Metacritic reviews. The Metacritic website[5] aggregates game reviews from professionals and amateurs. While amateur reviews have similarities with Steam reviews (e.g., Metacritic amateur reviews contain gameplay and experience descriptions), professional reviews are much longer and more complex [47]. Therefore, further investigation is necessary to properly assess whether the sentiment classifiers adopted in our work can be applied to professional Metacritic reviews.

B. Evaluating Sentiment Analysis Performance

We evaluated the performance of three sentiment analysis classifiers on game reviews, namely `Stanford CoreNLP` (version 3.9.2) [49], `NLTK` (version 3.4) [7], and `SentiStrength` (Windows version) [51]. For the purpose of evaluation of the classifiers, we consider the game recommendation flag on Steam as the sentiment truth label in our data, that is, we make the assumption that a review that recommends a game has a positive sentiment, while a review that does not recommend a game has a negative sentiment. Our dataset contains 10,603,348 positive reviews (*recommendation* = 1) and 1,735,016 negative reviews (*recommendation* = 0).

Although our dataset is imbalanced, we do not fix the imbalance since our goal is to evaluate existing techniques and reveal the root causes for wrong classifications rather than proposing a new sentiment analysis technique that outperforms the state-of-the-art. Furthermore, in the real world, data distribution is often imbalanced [12, 56] and existing re-sampling techniques have serious defects for text data [56]. Finally, we adopt the Area Under the Receiver Operating Characteristic Curve (AUC) evaluation metric, which is a robust metric with imbalanced data [23].

Regarding `NLTK`, we have the option to train it on our own data using the Naïve Bayes algorithm. Since it is computationally expensive to train and test it on our entire data, we adopt the out-of-sample bootstrap technique [16] to perform the training and testing, since the use of this technique allows us to avoid possible bias in the training and testing sets as we would have with a simple one-time sampling. In our work, the out-of-sample bootstrap technique consists of randomly sampling 100K reviews (sample) with no replacement from the entire set of reviews (population) to train the classifier. Then, we randomly select another 100K reviews from the pool of remaining reviews to test the classifier. The sample size (100K) was appropriately determined in a pre-study (detailed in Section V). The bootstrap process is repeated 1,000 times, which is enough to represent the entire population and reduce a possible bias in the training and testing sets. Note that for all executions of `NLTK`, before performing the classification itself, we have a preprocessing pipeline, which consists of tokenization, case normalization, and stop word removal. For the `SentiStrength` and `Stanford CoreNLP` classifiers, we also adopted the bootstrap technique and evaluated them on the same 1,000 samples used to test `NLTK`. Note also that we opted for not changing the configurations of the ready-to-use classifiers, such as `SentiStrength` and `Stanford CoreNLP`, as prior work has mostly used them without changes to their configuration [20, 27, 28, 29]. Keeping the configuration of classifiers similar to the configuration previously used allows us to evaluate and compare our results with existing literature more fairly.

For all the classifiers, we computed the Area Under the Receiver Operating Characteristic Curve (AUC). With the bootstrap process, we are able to obtain the AUC distribution for all the classifiers (1,000 AUC values corresponding to the 1,000 bootstrap iterations). The Receiver Operating Characteristic Curve plots the true positive rate against the false positive rate. The AUC measures the classifier's capability of distinguishing between positive and negative sentiments and ranges from 0.5 (random guessing) to 1 (best classification performance). For the cases in which the classification is neutral, we always consider it as a wrong classification since our data has only two labels: positive (the reviewer recommends the game) and negative (the reviewer does not recommend the game).

We compared the AUC distributions using the Wilcoxon rank-sum test. The Wilcoxon rank-sum test is an unpaired, non-parametric statistical test, where the null hypothesis is that two distributions are identical [54]. If the p-value of the applied Wilcoxon test is less than 0.05, then we can refute the null hypothesis, which means that the two distributions are significantly different. In addition to checking whether the two distributions are different, we provide the magnitude of the difference between the two distributions using Cliff's delta d [32] effect size. We adopt the following thresholds for d [45]:

$$\text{Effect size} = \begin{cases} negligible(N), & \text{if } |d| \leq 0.147 \\ small(S), & \text{if } 0.147 < |d| \leq 0.33 \\ medium(M), & \text{if } 0.33 < |d| \leq 0.474 \\ large(L), & \text{if } 0.474 < |d| \leq 1 \end{cases}$$

C. Manually Analyzing Wrong Classifications

To understand why classifiers are making wrong classifications and come up with the root causes which might be leading to the poor classification performance, we performed a manual analysis on the reviews that were wrongly classified by each of the three sentiment analysis classifiers we use. With this approach, we are more likely to identify characteristics from

[5]https://www.metacritic.com/

the review text itself that might confuse the classifier rather than wrong classifications due to bias in a classifier.

We adopt an inductive approach similar to the open-coding technique [14] to manually analyze the reviews. Initially, two authors independently read 100 reviews, being 50 wrongly classified as positive and 50 wrongly classified as negative. They then came up with causes that might have misled the classifiers. After discussing these causes and reaching an agreement on four causes (plus two categories in which the misclassification was unclear), we selected a representative sample with a confidence level of 95% and a confidence interval of 5%, which corresponds to 382 reviews. This sample was then classified into the set of agreed upon causes by one author so we could obtain the percentage of reviews for each cause.

D. Quantifying the Impact of the Root Causes

Based on the previous step, in which we extracted possible causes for wrongly classified reviews, we conducted a series of experiments to evaluate the impact of the identified causes, separately, on the performance of the sentiment analysis classifiers. For each cause, we selected the set of reviews that are affected by that cause (the affected set), the set of the remaining reviews (the unaffected set), computed the AUC distribution for both sets, and compared the AUC distributions using the Wilcoxon rank-sum test and the Cliff's delta effect size.

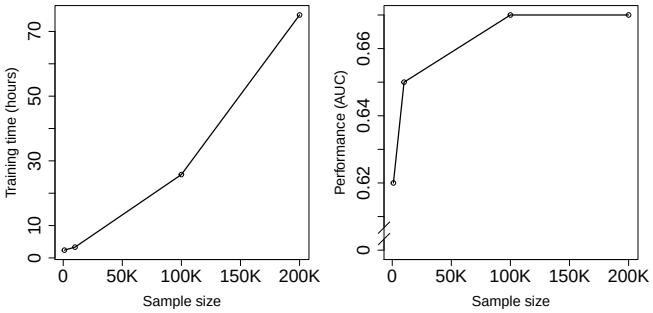
V. Pre-study

We need to find the best sample size to train the `NLTK` classifier. In this section, we present our pre-study to investigate the performance of `NLTK` with different sample sizes to train and test it.

Training `NLTK` on our entire dataset would be computationally expensive. We designed two experiments to determine the proper training and testing set sample sizes so we can apply the out-of-sample bootstrap technique, as we explained in Section IV-B. In Figure 4, we can see the plots regarding our experiments. For both cases, we used the following values for the sample size (number of reviews): 1K, 10K, 100K, and 200K. Figure 4a presents how the training time (in hours) varies with the sample size. As we can see, the time increases quickly with the increase in sample size (jumping from 26 hours, for 100K, to 75 hours of training time, for 200K). Therefore, a sample size larger than 200K would be infeasible.

Figure 4b presents how the performance of the `NLTK` classifier (by means of the median AUC) varies with the increase in the sample size. As we can observe, the plot plateaus when it reaches 100K (presenting an AUC of 0.67), which means using 100K reviews is sufficient for our purpose. Using the result of this experiment together with the result of the previous experiment, we decided to use a sample of 100K game reviews to train and test the `NLTK` classifier. We also used the same sample size to evaluate the `SentiStrength` and the `Stanford CoreNLP` classifiers.

These results provide evidence of the richness of game review data as we do not need the entire dataset to train our model, indicating that, although the sentiment classification is a tricky problem, we have a rich dataset for which the model does not need huge amounts of data to learn from.



(a) Sample size versus training time. (b) Sample size versus AUC.

Fig. 4: Plots of experiments to determine the sample size for `NLTK`.

VI. RQ1: How do sentiment analysis classifiers perform on game reviews?

Motivation: It is important to verify the performance of widely-used sentiment analysis classifiers on game reviews as this is the first step to understand whether current sentiment analysis classifiers are suitable for classifying the sentiment of such data.

Approach: For this research question, we applied the out-of-sample bootstrap with 1,000 iterations to evaluate the `NLTK`, `Stanford CoreNLP` and `SentiStrength` classifiers on the game review. To evaluate the classifiers, we computed five metrics: accuracy, precision, recall, F-measure, and AUC. We also performed an experiment to investigate how the length of the reviews affects the performance of the sentiment classification. The reviews were split into 51 groups according to their length: reviews with less than 20 characters, reviews with length between 20 and 40 characters (exclusive), reviews with length between 40 and 60 characters (exclusive), and so on up to the last group of reviews with more than 1,000 characters. We evaluated each classifier with a sample of 10K reviews from each length range. Finally, we compared the performance of the sentiment classification of game reviews with the sentiment classification of other three corpora (Stack Overflow posts, Jira issues, and mobile app reviews), as indicated by prior work [26, 29].

Findings: Table II presents the metrics for the imbalanced and balanced versions of the dataset. Note that we provide all these metrics for the purpose of comparisons with prior (and future) work, but for our discussions, we will focus on the AUC metric. **`NLTK` achieved the best performance of sentiment analysis (in the studied configuration) on game reviews while `Stanford CoreNLP` presented the worst performance.** Figure 5 presents the distribution of the AUC metric for the classifiers (each value corresponds to an iteration of the bootstrap).

TABLE II: Evaluation metrics (median) for unbalanced and balanced dataset.

Classifier	Acc.	Precision	Recall	F-measure	AUC
NLTK	0.61	0.60	0.70	0.54	0.70
NLTK (balanced)	0.67	0.73	0.67	0.65	0.67
SentiStr.	0.52	0.56	0.63	0.47	0.63
SentiStr. (balanced)	0.63	0.65	0.63	0.62	0.63
Stanf. NLP	0.37	0.52	0.53	0.35	0.53
Stanf. NLP (balanced)	0.53	0.54	0.53	0.51	0.53

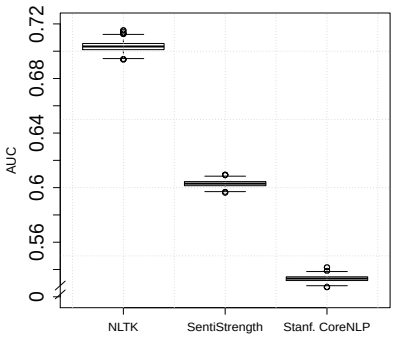


Fig. 5: AUC distribution for NLTK bootstrap.

The AUC for NLTK varies from 0.69 up to 0.72, with a median value around 0.70. For SentiStrength, the AUC ranges from 0.60 to 0.61 with a median of 0.60, while for Stanford CoreNLP, the AUC ranges from 0.53 to 0.54 with a median of 0.53. For all classifier pairs ([NLTK, SentiStrength], [NLTK, Stanford CoreNLP], and [SentiStrength, Stanford CoreNLP]), the Wilcoxon rank-sum test shows that the two distributions are significantly different, with a large Cliff's delta effect size.

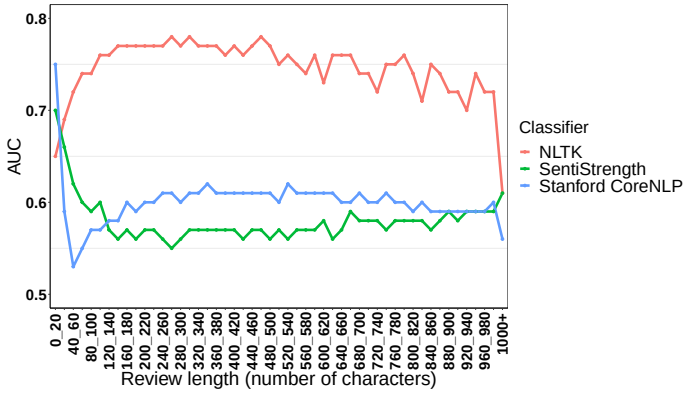


Fig. 6: Performance of classifiers for different length ranges. Note that there is a data point for every range of 20 characters (0-20, 20-40, and so on). However, for the purpose of a better visualization, the figure only displays every other range in the *x* axis (e.g., the label '20_40' is not shown in the plot, but the corresponding data point for that range is present in the plot).

Figure 6 shows that the performance of the classifiers remains mostly stable across different review lengths, with the largest changes occurring for reviews with less than 20 characters (AUC of 0.65 for NLTK) and reviews with more than 1,000 characters (AUC of 0.61 for NLTK). We can also see that NLTK's performance slightly reduces as the review length increases. Finally, we computed the distribution of different review lengths in our dataset. We found that 75% of the reviews are in the range 20-1000 characters (where NLTK performs best), while 20% of the reviews have less than 20 characters, and 5% of the reviews have more than 1,000 characters.

TABLE III: F-measure of sentiment classification across different corpora.

Corpora	NLTK	SentiStrength	Stanford CoreNLP
Game reviews	0.54	0.47	0.35
Stack Overflow posts	0.21	0.34	0.28
Jira issues	**0.55**	**0.62**	**0.52**
Mobile app reviews	0.53	**0.64**	**0.74**

Finally, Table III presents the F-measure metric of the sentiment classification across different corpora. The text in bold represents a classification performance better than for game reviews. As we can see, the classifiers usually perform better when using a corpus other than game reviews. After training NLTK on game reviews, it achieves a performance that is similar to the performance on the Jira issues and mobile app reviews corpora. However, SentiStrength and Stanford CoreNLP work much better on the Jira issues and mobile app reviews corpora compared to game reviews.

> Overall, sentiment analysis classifiers do not achieve a high performance, performing worse on game reviews than on other domains. The median AUC ranged from 0.53 (Stanford CoreNLP) to 0.70 (NLTK).

VII. RQ2: What are the root causes for wrong classifications?

Motivation: Understanding what is causing sentiment analysis classifier to make wrong classifications is essential to extract important insights about how to improve existing sentiment analysis for game reviews. Such knowledge can be used to fix problems in the classification pipeline and achieve a better performance.

Approach: We start by (1) selecting the reviews that were misclassified by all the three classifiers simultaneously (i.e., the intersection of misclassified reviews). We then (2) use the pool of all misclassified reviews to select a statistically representative sample of 382 reviews for the manual analysis. We adopted an open coding-like approach to identify the root causes which could affect sentiment analysis classifiers' performance. Two authors independently analyzed a sample of 100 reviews (50% wrongly classified as positive and 50% wrongly classified as negative) to identify the root causes that may confuse the classifiers. Our manual analysis had an agreement of 83% between the two authors (we consider

TABLE IV: Root causes for misclassifications in sentiment analysis (each review may be assigned to more than one root cause).

Root cause	Definition	Occurrence (%)
Contrast conjunctions	The review points out both the advantages and disadvantages of the game, frequently using contrast conjunctions	30
Game comparison	The review contains a comparison with another game or with a previous version of the game itself	25
Negative terminology	The review contains words such as *kill* and *evil* which are not necessarily bad for specific game genres (e.g., action games)	23
Unclear	It is not clear what might have caused the wrong classification	21
Sarcasm	The review contains sarcastic text	6
Mismatched recommendation	The user might have entered a wrong recommendation: positive (negative) recommendation with a negative (positive) review content	6

an agreement when both authors agreed that the root cause X is related to a review Y). After reaching the agreement, one author analyzed a statistically representative sample of 382 reviews (which yields a confidence level of 95% with a confidence interval of 5) to compute the frequency of occurrence of each cause. Note that each misclassification may be assigned to more than one root cause (if that is the case). The sample of 382 reviews for the manual analysis was obtained from the reviews that were misclassified by all three classifiers. We focused on reviews that were misclassified by all classifiers to better identify characteristics of the review text that affect the sentiment analysis classification, rather than a characteristic of only a single classifier.

Findings: **We revealed four types of possible causes for sentiment misclassifications: use of contrast conjunctions to indicate the advantages and disadvantages of a game in the same review, comparison to other games, reviews with negative terminology, and sarcasm.** Table IV presents all the root causes we identified along with their definitions and percentage of occurrence. As we can see, the most common cause is contrast conjunctions (30%), followed by game comparison (25%), negative terminology (23%), and sarcasm (6%). Cases for which we are not able to clearly identify the cause for the wrong classification (unclear) occurred in 21% of the reviews. Cases in which the review content did not match the recommendation (mismatched recommendation) occurred in 6% of the reviews.

Next, we present each root cause in detail along with corresponding examples of reviews.

Root cause 1: Contrast conjunctions

Description: The review points out advantages and disadvantages of the game.

Symptoms: This type of review frequently makes use of contrast conjunctions (*but*, *although*, *though*, *even though*, and *even if*) when presenting positive and negative points about the game. As we can see in the example below, the review contains a positive view ("*I love this game...*") and a negative view ("*...it keeps flickering please help!*") about the game separated by the conjunction *but*.

Example: "*I love this game but it keeps flickering, please help!*".

Root cause 2: Game comparison

Description: The review compares the game with another game or a previous version of the game itself. Such comparisons might make the sentiment classification more difficult since positive or negative points might refer to the other game or the game itself in a previous version instead of the current game version under review.

Symptoms: The review mentions one or more games [A, B...] in a review for another game [G], or mentions a version 1.x of the game [G] in a review for the version 2.x of the same game [G]. In the example below, the review for the *Terraria* game compares the reviewed version of the game with a previous version.

Example: "*Terraria was one of the best games I've ever played, but after they released 1.2, I stopped enjoying it!*".

Root cause 3: Negative terminology

Description: The review uses (supposedly) negative terminology (i.e., words with a negative connotation), which might mislead the classifier towards a negative sentiment classification even though many times the review text has a positive sentiment (as indicated by the recommendation of the game).

Symptoms: The review contains words that are considered negative in many situations (e.g., *kill*, *evil*), which might not have a negative connotation for games of specific genres, such as first-person shooter games. The review in the example below contains supposedly negative words, such as *kill*, although it is just describing the role of the player in the game. In fact, the reviewer recommended the game and even made it explicitly by assigning a score of 10 out of 10 to the game.

Example: "*I've played like 15 games [...], zombies just go around you, you can't run, just keep trying to kill them.*".

Root cause 4: Unclear

Description: We are not able to clearly identify a pattern or characteristic that might be confusing the classifier.

Symptoms: There is no symptom. We cannot identify a clear possible reason which might mislead the classifier. The review in the example below was classified as positive while it should be negative.

Example: "*Downloaded Game into steam, Played for 40 Hours total. Game disappeared from computer. Redownloaded, Played for a while, Game disappeared again. As*

someone with a download cap and 2 other gamers in the house, was. not. impressed".

Root cause 5: Sarcasm

Description: The review contains sarcastic text. Sarcasm occurs when an apparently positive text is actually used to convey a negative attitude (or vice-versa) [18]. Prior work has shown that sarcasm is difficult to automatically identify [39].
Symptoms: The review contains sarcasm, which is observed when the reviewer writes an (apparently) positive text intending to transmit a negative message (or vice-versa). The review in the example below contains sarcastic text as the reviewer makes use of positive words (e.g., *great*), when the person actually points out a negative aspect about the game.
Example: "*Great for uninstalling 11/10 would uninstall again*".

Root cause 6: Mismatched recommendation

Description: It means the reviewer might have entered a wrong recommendation, which does not match with the review content itself. Note that this root cause is different from sarcasm (as we cannot clearly identify a positive review intending to transmit a negative attitude or vice-versa), however both causes are hard to be automatically identified.
Symptoms: The reviewer is positive about the game, but they did not recommend the game (or vice-versa). The example below presents a review that was classified as positive (as expected since the text clearly expresses a positive sentiment). However, the reviewer did not recommend the game, which we assume was a mistake of the reviewer.
Example: "*I love this GAME!*".

We identified four root causes for wrong classifications of sentiment analysis classifiers: use of contrast conjunctions (30%), game comparisons (25%), negative terminology (23%), and sarcasm (6%).

VIII. RQ3: To what extent do the identified root causes impact the performance of sentiment analysis?

Motivation: It is important to quantify the impact of each identified root cause to the overall performance of sentiment analysis on game reviews. Such knowledge will support the prioritization of the causes to be addressed, the implementation of better sentiment analysis tools to be deployed in gaming contexts, and a research agenda to address such issues.

Approach: For this part of the study, we first identified the root causes which are feasible to be automatically identified in reviews. Then, we implemented detection heuristics to identify reviews affected by each root cause. We focused on the following root causes for which we can automatically identify reviews: **contrast conjunctions**, **game comparison**, and **negative terminology**. After identifying such reviews, we re-ran the `NLTK` classifier on both groups: the set of identified reviews (affected set, which is supposedly harder for the classifier) and the set of remaining reviews (unaffected set, which is supposedly easier for the classifier since they do not contain the cause for the wrong classification).

Note that, in this last part, we focused only on the `NLTK` classifier as it presented the best performance (Section VI) and it can be trained on our data. Furthermore, we also applied the bootstrap technique with 1,000 iterations, as we previously did.

Next, we explain the implemented detection heuristics and the obtained results for each root cause.

A. Contrast Conjunctions

Detection heuristic: We noticed that reviews which point out the advantages and disadvantages of a game usually use contrast conjunctions to transmit the idea of contrast between advantages and disadvantages of the game. We defined a list with the contrast conjunctions we observed in our manual analysis and performed a keyword-based search in our dataset to identify reviews that contain one or more conjunctions of the list. Table V presents the selected conjunctions with examples.

Among the most frequent conjunctions found in the reviews, we have "but" (1,941,535), "although" (104,295), and "even if" (66,802). After the search, we ended up with 10,187,926 reviews in the remaining set (82% of the original dataset) and 2,150,438 reviews in the detected set (identified by the heuristic).

Findings: **Game reviews with contrast conjunctions are indeed more difficult to classify for `NLTK`, with a median AUC that is 11% lower than for reviews without contrast conjunctions.** Figure 7 presents the distributions of the AUC for the sets of reviews without and with contrast conjunctions. The Wilcoxon rank-sum test shows that the two distributions are significantly different, with a large (1.0) Cliff's delta effect size.

As we can observe, the AUC of reviews without contrast is much higher (large Cliff's delta effect size) than the AUC of reviews with the presence of contrast conjunctions. In fact, we found a median AUC of almost 0.75 for the group without contrast, while for the group with contrast the median AUC is around 0.67 (11% lower).

B. Game Comparison

Detection heuristic: We collected the top-500 most played games from Steam and, based on this list, we performed a keyword-based search in our dataset to identify reviews that mention other games.

We collected the most played games from the SteamDB platform.[6] This list was obtained based on the peak number of players who have played the game. For instance, the number one game in the list is *Playerunknown's Battlegrounds* (3,257,248 players), followed by *Dota 2* (1,295,114 players) and *Counter-Strike: Global Offensive* (854,801 players). This data was collected on January 10th, 2020. Note that, although the game reviews were collected in 2016, their age does not impact our analysis.

We performed a keyword-based search on the reviews in our dataset. We ensured that both the game name being

[6]https://steamdb.info/

TABLE V: Contrast conjunctions and corresponding examples.

Contrast conjunction	Example
But	*Nice Matchmaking,* ***but*** *if you are not premium you have no chance...*
Although, though	***Although*** *I really enjoy this game, I do think that PTTM still remains the best in the series...*
Even though, even if	***Even though*** *it gets progressively difficult and you won't get the perfect items each run, you'll find yourself coming back for more...*

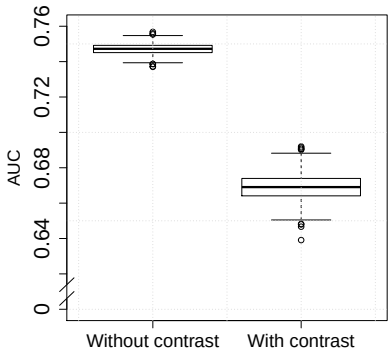


Fig. 7: AUC distribution for reviews without and with contrast.

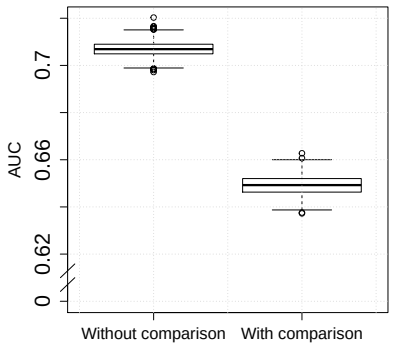


Fig. 8: AUC distribution for reviews without and with comparison.

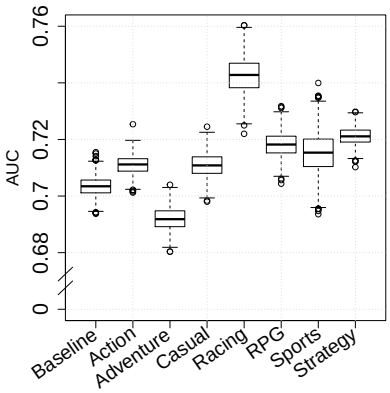


Fig. 9: AUC distribution for reviews of all the game genres and the baseline.

searched and the review text were lower case during the search. Among the most mentioned games in the reviews, we found *Terraria* (31,860), *Dota 2* (30,385), and *Counter-Strike* (21,418). After the search, we ended up with 11,753,211 reviews in the remaining set (95% of the original dataset) and 585,153 reviews in the detected set (identified by the heuristic). *Findings:* **Game reviews with comparisons are actually more difficult to classify for `NLTK`, with a median AUC that is 8% lower than for reviews without comparisons.** After training `NLTK` on both sets, we computed the AUC using the out-of-sample bootstrap with 1,000 iterations, as we did previously. Figure 8 presents the distributions of the AUC for the sets of reviews without and with comparison. The Wilcoxon rank-sum test shows that the two distributions are significantly different, with a large (1.0) Cliff's delta effect size.

As we can see, reviews without comparison present a higher AUC than reviews with comparison. In fact, we found a median AUC of almost 0.71 for the group without comparison, while for the group with comparison the median AUC is around 0.65 (8% lower), which indicates that, similarly to the case of reviews with contrast conjunctions, comparisons can also degrade the performance of sentiment analysis.

C. Negative Terminology

Detection heuristic: We noticed that some game reviews use words with a negative connotation, such as *kill*, *evil*, and *death*. Although such words might refer to negative aspects of something in a usual context, within the context of games they might be used without the negative connotation. For instance, when describing the role of a character in an RPG (Role-Playing Game) game, one might say they need to **defeat** and **kill** the **enemy**. Although the review uses (supposedly) negative words, its final content might be positive towards the game (i.e., the reviewer might recommend the game even when using negative words).

For this root cause, instead of adopting the approach as we did for the previous causes, we propose a stratified training process for the sentiment analysis classifier based on the game genre, which we call per-genre training. We used a customized crawler to collect the game genre from Steam for each review in our dataset and grouped reviews by genre so we could train the classifier separately by genre. We found a list of seven game genres (excluding generic genres reported by Steam, such as Early Access, Free to Play, and Indie): Action, Adventure, Strategy, RPG, Casual, Racing, and Sports.

We established a maximum period of one month to collect the game genres for a randomized version of our data, which resulted in genres for 4 million reviews. It would be infeasible to collect the genre for our entire dataset in a timely manner due to restrictions when using a crawler to collect online data (such as the limited number of requests allowed per a period of time). Furthermore, for some cases, the review or the profile itself was excluded by the user from the Steam platform. In the case of less popular genres for which we are not able to sample 100K reviews for the training and testing sets (casual, racing, and sports genres), we adopted a 80/20 percentage split to train and test with the bootstrap technique. For instance, if we had 10K reviews for a specific genre, we would use 8K for training (80%) and 2K for testing (20%). Table VI presents the number of reviews for each genre.

Findings: **Per-genre training is effective when performing sentiment analysis on game reviews.** Figure 9 presents the distribution of the AUC for all the genres and also for the baseline, which is the evaluation of `NLTK` on the entire dataset

TABLE VI: Game genres and corresponding number of reviews.

Genre	Number of reviews
Action	741,569
Adventure	484,236
Strategy	395,595
RPG	372,033
Casual	128,590
Racing	43,899
Sports	33,890

(Section VI). We can see that, for all the genres except for adventure, the median AUC is higher than the median AUC for the baseline. In fact, we obtained the following median AUC values: 0.70 (baseline), 0.71 (action), 0.69 (adventure), 0.71 (casual), 0.74 (racing), 0.72 (RPG), 0.72 (sports), and 0.72 (strategy). The Wilcoxon rank-sum test shows that the AUC distribution for each genre is significantly different from the AUC distribution for the baseline with a large effect size.

Reviews that use contrast conjunctions to point out advantages and disadvantages of the game have the highest negative impact on the performance (11% lower AUC), followed by reviews with game comparisons (8% lower AUC). Furthermore, we show that per-genre training is effective for sentiment analysis on game reviews as it is mostly able to improve the performance of NLTK.

IX. Recommendations and research directions for sentiment analysis on game reviews

In this section, we provide practical recommendations for performing sentiment analysis on game review data.

No need for huge amounts of data. Through our pre-study we showed that 100K reviews is a sufficient sample size to train and test sentiment analysis classifiers on game reviews. We showed that using more than 100K reviews does not improve the sentiment analysis performance as it plateaus after 100K reviews. Note that this is based on a Naïve Bayes classifier as we aim to provide recommendations for computationally accessible approaches rather than computationally intensive deep learning algorithms. Furthermore, this recommendation is based on the NLTK configurations adopted in the study (i.e., the machine learning version of NLTK with the same preprocessing steps).

Prioritize on studying techniques that can deal with reviews with advantages and disadvantages of the game. Based on the impact that each root cause has on the sentiment analysis performance, we suggest game developers and researchers to develop techniques that can analyze reviews which use contrast conjunctions to point out the advantages and disadvantages of the game under review as this might confuse the classifier. Secondly, we suggest to develop techniques that can deal with reviews which make comparison to games other than the game under review or to previous versions of the game itself. Finally, we suggest the development of techniques to analyze reviews that contain sarcasm.

Stratify reviews by game genre. Different game genres have different characteristics in terms of expressions used by reviewers. Therefore, we recommend to stratify the dataset by genre and train the classifier separately for each genre. This approach helps to avoid mixing different types of data when training the model. For instance, negative words (e.g., *evil*) are used for different purposes in reviews of different genres, such as casual (where the reviewer probably uses it with a negative connotation) and first-person shooter (where the reviewer does not intentionally have a negative connotation).

X. Conclusion and Future Work

In this paper, we perform a large-scale study to understand how sentiment analysis works on game reviews. We collected 12 million reviews from the Steam platform. We investigate the performance of existing sentiment analysis classifiers on game reviews, identify which factors might impact such performance and to what extent.

Our study shows that sentiment analysis classifiers do not perform well on game reviews and we identified root causes for such performance, such as sarcasm and reviews with negative terminology. Reviews that point out advantages and disadvantages of a game (through the use of contrast conjunctions) have a high negative impact on the performance (reducing the median AUC by 11%), followed by reviews that contain comparisons to games other then the game under review (reducing the median AUC by 8%). Furthermore, we show that training classifiers on reviews stratified by the genre is effective and can improve the performance of sentiment analysis. For all genres except adventure, the median AUC was higher than the baseline, with significant different AUC distributions and large effect sizes.

Our study is the first important step towards identifying what are the root causes for wrong classifications in sentiment analysis on game reviews and the impact of each cause. Our study calls upon sentiment analysis and game researchers to further investigate how the performance of sentiment analysis on game reviews can be improved, for instance by developing techniques that can automatically deal with specific game-related issues of reviews (e.g., reviews with contrast conjunctions and reviews with game comparisons). Another future direction is to explore how user characteristics affect the performance of the sentiment classification of game reviews.

References

[1] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. J. Passonneau, "Sentiment analysis of Twitter data," in *Proceedings of the Workshop on Language in Social Media (LSM 2011)*, pp. 30–38.

[2] M. Araújo, P. Gonçalves, M. Cha, and F. Benevenuto, "ifeel: a system that compares and combines sentiment analysis methods," in *Proc. of the 23rd Int'l Conference on World Wide Web*, 2014, pp. 75–78.

[3] D. Bamman and N. A. Smith, "Contextualized sarcasm detection on Twitter," in *Ninth Int'l AAAI Conference on Web and Social Media*, 2015.

[4] B. Bazelli, A. Hindle, and E. Stroulia, "On the personality traits of StackOverflow users," in *2013 IEEE Int'l conference on software maintenance*, pp. 460–463.

[5] S. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based sarcasm sentiment recognition in Twitter data," in *Proc. of the 2015 IEEE/ACM Int'l Conference on Advances in Social Networks Analysis and Mining*, pp. 1373–1380.

[6] G. Biondi, V. Franzoni, and V. Poggioni, "A deep learning semantic approach to emotion recognition using the IBM Watson bluemix alchemy language," in *Int'l Conference on Computational Science and Its Applications*. Springer, 2017, pp. 718–729.

[7] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: Analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[8] M. Bouazizi and T. Ohtsuki, "Opinion mining in Twitter: How to make use of sarcasm to enhance sentiment analysis," in *2015 IEEE/ACM Int'l Conference on Advances in Social Networks Analysis and Mining*, pp. 1594–1597.

[9] F. Calefato, F. Lanubile, and N. Novielli, "Emotxt: A toolkit for emotion recognition from text," in *2017 7th Int'l conference on Affective Computing and Intelligent Interaction Workshops and Demos)*. IEEE, pp. 79–80.

[10] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, "Sentiment polarity detection for software development," *Empirical Software Engineering*, vol. 23, no. 3, pp. 1352–1382, 2018.

[11] L. V. G. Carreño and K. Winbladh, "Analysis of user comments: An approach for software requirements evolution," in *2013 35th Int'l Conference on Software Engineering (ICSE)*. IEEE, pp. 582–591.

[12] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[13] C. Chiu, R.-J. Sung, Y.-R. Chen, and C.-H. Hsiao, "App review analytics of free games listed on Google play," in *Proceedings of the 13th Int'l Conference on Electronic Business, Singapore*, 2013.

[14] J. M. Corbin and A. Strauss, "Grounded theory research: Procedures, canons, and evaluative criteria," *Qualitative sociology*, vol. 13, no. 1, pp. 3–21, 1990.

[15] L. Dong, F. Wei, M. Zhou, and K. Xu, "Adaptive multicompositionality for recursive neural models with applications to sentiment analysis," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[16] B. Efron and R. J. Tibshirani, *An introduction to the bootstrap*. CRC press, 1994.

[17] T. Finin, W. Murnane, A. Karandikar, N. Keller, J. Martineau, and M. Dredze, "Annotating named entities in Twitter data with crowdsourcing," in *Proc. of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pp. 80–88.

[18] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in Twitter: A closer look," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2*. Association for Computational Linguistics, 2011, pp. 581–586.

[19] M. Goul, O. Marjanovic, S. Baxley, and K. Vizecky, "Managing the enterprise business intelligence app store: Sentiment analysis supported requirements engineering," in *2012 45th Hawaii Int'l Conference on System Sciences*. IEEE, pp. 4168–4177.

[20] E. Guzman and W. Maalej, "How do users like this feature? A fine grained sentiment analysis of app reviews," in *2014 IEEE 22nd Int'l requirements engineering conference (RE)*, pp. 153–162.

[21] E. Guzman, O. Aly, and B. Bruegge, "Retrieving diverse opinions from app reviews," in *2015 ACM/IEEE Int'l Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–10.

[22] E. Guzman, D. Azócar, and Y. Li, "Sentiment analysis of commit comments in GitHub: An empirical study," in *Proc. of the 11th Working Conference on Mining Software Repositories*. ACM, 2014, pp. 352–355.

[23] J. Huang and C. X. Ling, "Using AUC and accuracy in evaluating learning algorithms," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 3, pp. 299–310, 2005.

[24] M. R. Islam and M. F. Zibran, "Leveraging automated sentiment analysis in software engineering," in *2017 IEEE/ACM 14th Int'l Conference on Mining Software Repositories (MSR)*, pp. 203–214.

[25] ——, "A comparison of software engineering domain specific sentiment analysis tools," in *2018 IEEE 25th Int'l Conference on Software Analysis, Evolution and Reengineering (SANER)*, pp. 487–491.

[26] ——, "Sentistrength-se: Exploiting domain specificity for improved sentiment analysis in software engineering text," *Journal of Systems and Software*, vol. 145, pp. 125–146, 2018.

[27] R. Jongeling, S. Datta, and A. Serebrenik, "Choosing your weapons: On sentiment analysis tools for software engineering research," in *2015 IEEE Int'l Conference on Software Maintenance and Evolution*, pp. 531–535.

[28] R. Jongeling, P. Sarkar, S. Datta, and A. Serebrenik, "On negative results when using sentiment analysis tools for software engineering research," *Empirical Software Engineering*, vol. 22, no. 5, pp. 2543–2584, 2017.

[29] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, "Sentiment analysis for software engineering: How far can we go?" in *2018 IEEE/ACM 40th Int'l Conference on Software Engineering*, pp. 94–104.

[30] C. Lin and Y. He, "Joint sentiment/topic model for sentiment analysis," in *Proceedings of the 18th ACM conference on Information and knowledge management*, 2009, pp. 375–384.

[31] D. Lin, C.-P. Bezemer, Y. Zou, and A. E. Hassan, "An empirical study of game reviews on the Steam platform," *Empirical Software Engineering*, vol. 24, no. 1, pp. 170–207, 2019.

[32] J. D. Long, D. Feng, and N. Cliff, "Ordinal analysis of behavioral data," *Handbook of psychology*, pp. 635–661, 2003.

[33] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky, "The Stanford CoreNLP

natural language processing toolkit," in *Proc. of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[34] P. Mishra, R. Rajnish, and P. Kumar, "Sentiment analysis of Twitter data: Case study on digital India," in *2016 Int'l Conference on Information Technology*. IEEE, pp. 148–153.

[35] A. Mudinas, D. Zhang, and M. Levene, "Combining lexicon and learning based approaches for concept-level sentiment analysis," in *Proc. of the 1st Int'l workshop on issues of sentiment discovery and opinion mining*, 2012, pp. 1–8.

[36] P. Nand, R. Perera, and R. Lal, "A HMM POS tagger for micro-blogging type texts," in *Pacific Rim Int'l Conference on Artificial Intelligence*, 2014, pp. 157–169.

[37] N. Novielli, F. Calefato, and F. Lanubile, "A gold standard for emotion annotation in Stack Overflow," in *2018 IEEE/ACM 15th Int'l Conference on Mining Software Repositories (MSR)*, pp. 14–17.

[38] M. Ortu, B. Adams, G. Destefanis, P. Tourani, M. Marchesi, and R. Tonelli, "Are bullies more productive?: Empirical study of affectiveness vs. issue fixing time," in *Proc. of the 12th Working Conference on Mining Software Repositories*. IEEE Press, 2015, pp. 303–313.

[39] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1–2, pp. 1–135, 2008.

[40] S. Panichella, A. Di Sorbo, E. Guzman, C. A. Visaggio, G. Canfora, and H. C. Gall, "How can I improve my app? Classifying user reviews for software maintenance and evolution," in *2015 IEEE Int'l Conference on Software Maintenance and Evolution (ICSME)*, pp. 281–290.

[41] D. Pletea, B. Vasilescu, and A. Serebrenik, "Security and emotion: Sentiment analysis of security discussions on GitHub," in *Proc. of the 11th working conference on mining software repositories*, 2014, pp. 348–351.

[42] M. M. Rahman, C. K. Roy, and I. Keivanloo, "Recommending insightful comments for source code using crowdsourced knowledge," in *2015 IEEE 15th Int'l Working Conference on Source Code Analysis and Manipulation (SCAM)*, pp. 81–90.

[43] K. Raison, N. Tomuro, S. Lytinen, and J. P. Zagal, "Extraction of user opinions by adjective-context co-clustering for game review texts," in *Int'l Conference on NLP*. Springer, 2012, pp. 289–299.

[44] P. C. Rigby and A. E. Hassan, "What can OSS mailing lists tell us? A preliminary psychometric text analysis of the Apache developer mailing list," in *Fourth Int'l Workshop on Mining Software Repositories*. IEEE, 2007, pp. 23–23.

[45] J. Romano, J. D. Kromrey, J. Coraggio, J. Skowronek, and L. Devine, "Exploring methods for evaluating group differences on the nsse and other surveys: Are the t-test and cohen's d indices the most appropriate choices," in *annual meeting of the Southern Association for Institutional Research*. Citeseer, 2006, pp. 1–51.

[46] H. Saif, Y. He, and H. Alani, "Semantic sentiment analysis of Twitter," in *Int'l semantic web conference*. Springer, 2012, pp. 508–524.

[47] T. Santos, F. Lemmerich, M. Strohmaier, and D. Helic, "What's in a review: Discrepancies between expert and amateur reviews of video games on Metacritic," *Proc. of the ACM on Human-Computer Interaction*, vol. 3, pp. 1–22, 2019.

[48] V. K. Singh, R. Piryani, A. Uddin, and P. Waila, "Sentiment analysis of movie reviews: A new feature-based heuristic for aspect-level sentiment classification," in *2013 Int'l Mutli-Conference on Automation, Computing, Communication, Control and Compressed Sensing)*. IEEE, pp. 712–717.

[49] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642.

[50] B. Stråat and H. Verhagen, "Using user created game reviews for sentiment analysis: A method for researching user attitudes." in *GHITALY@ CHItaly*, 2017.

[51] M. Thelwall, K. Buckley, G. Paltoglou, D. Cai, and A. Kappas, "Sentiment strength detection in short informal text," *Journal of the American Society for Information Science and Technology*, pp. 2544–2558, 2010.

[52] J. J. Thompson, B. H. Leung, M. R. Blair, and M. Taboada, "Sentiment analysis of player chat messaging in the video game StarCraft 2: Extending a lexicon-based model," *Knowledge-Based Systems*, vol. 137, pp. 149–162, 2017.

[53] S. Wijayanto and M. L. Khodra, "Business intelligence according to aspect-based sentiment analysis using double propagation," in *2018 3rd Int'l Conference on Information Technology, Information System and Electrical Engineering*. IEEE, pp. 105–109.

[54] F. Wilcoxon, "Individual comparisons by ranking methods," in *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[55] Y. Woldemariam, "Sentiment analysis in a cross-media analysis framework," in *2016 IEEE International Conference on Big Data Analysis (ICBDA)*, pp. 1–5.

[56] Z. Xiao, L. Wang, and J. Du, "Improving the performance of sentiment classification on imbalanced datasets with transfer learning," *IEEE Access*, vol. 7, pp. 28 281–28 290, 2019.

[57] K. Yauris and M. L. Khodra, "Aspect-based summarization for game review using double propagation," in *2017 International Conference on Advanced Informatics, Concepts, Theory, and Applications*. IEEE, pp. 1–6.

[58] J. P. Zagal, A. Ladd, and T. Johnson, "Characterizing and understanding game reviews," in *Proc. of the 4th Int'l Conference on Foundations of Digital Games*, 2009, pp. 215–222.

[59] J. P. Zagal, N. Tomuro, and A. Shepitsen, "Natural language processing in game studies research: An overview," *Simulation & Gaming*, pp. 356–373, 2012.

[60] J. P. Zagal and N. Tomuro, "Cultural differences in game appreciation: A study of player game reviews." in *FDG*, 2013, pp. 86–93.