Detecting Spam Game Reviews on Steam with a Semi-Supervised Approach

PENGZE BIAN, LEI LIU, and PENNY SWEETSER, Australian National University, Australia

The potential value of online reviews has led to more and more spam reviews appearing on the web. These spam reviews are widely distributed, harmful, and difficult to identify manually. In this paper, we explore and implement generalised approaches for identifying online deceptive spam game reviews from Steam. We analyse spam game reviews and present and validate some techniques to detect them. In addition, we aim to identify the unique features of game reviews and to create a labelled game review dataset based on different features. We were able to create a labelled dataset that can be used to identify spam game reviews in future research. Our method resulted in 5,021 of the 33,450 unlabelled Steam reviews being labelled as spam reviews, or approximately 15%. This falls within the expected range of 10-20% and maps to the Yelp figures of 14-20% of reviews are spam.

CCS Concepts: • **Computing methodologies** → **Natural language processing**; • **Software and its engineering** → **Interactive games**; • **Applied computing** → **Computer games**.

Additional Key Words and Phrases: game reviews; natural language processing; NLP; video games; qualitative analysis; text mining; Steam

ACM Reference Format:
Pengze Bian, Lei Liu, and Penny Sweetser. 2021. Detecting Spam Game Reviews on Steam with a Semi-Supervised Approach. In *International Conference on the Foundations of Digital Games (FDG '21), August 3–6, 2021, Montreal, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

There are a multitude of popular websites that enable and encourage consumers to leave their opinions or views on products that they have bought or used. For example, as of Q3 2020[1], the USA review site Yelp has more than 220 million online reviews of local businesses. Reviews reflect user experiences, evaluations, and opinions [33]. This information can influence other consumers' initial impressions of products as well as purchasing decisions [35]. The potential value of online reviews prompts many people to write spam reviews [23]. Some organisations even hire professionals to promote their own products or defame their competitors' products [13]. Spam reviews can be categorised into three types [9]: 1) untruthful reviews, 2) brand-only reviews, and 3) non-reviews. Untruthful reviews include intentionally deceptive or maliciously defamatory comments. Brand-only reviews comment only on the brands, manufacturers, or sellers of the products, rather than the product itself. Non-reviews include advertisements and irrelevant comments without meaningful opinions (e.g., random text or symbols).

Spam reviews are widely distributed, harmful, and difficult to identify manually [8, 16, 38]. According to Fei et al. [4], deceptive reviews account for 14-20% of Yelp reviews. This large number of spam reviews not only harms the interests of merchants and the market more broadly, but also causes consumers to distrust online reviews [27]. Ott et al. [22]

[1]https://www.yelp-press.com/company/fast-facts

Manuscript submitted to ACM

organised three volunteers to manually identify spam comments and found that volunteers tended to misjudge fake reviews as real reviews, with recognition accuracy of 53.1%-61.9%. This research demonstrates that annotating fake reviews manually is not feasible due to the low accuracy, in addition to being time-consuming. As a result, identifying spam reviews effectively is an open research challenge.

Previous research has investigated the analysis and detection of spam reviews, starting with Jindal and Liu [9] in 2008. Their work focused on identifying untruthful reviews. Subsequently, researchers investigated constructing review text features [11, 15] and reviewer behavior features [17, 19] to find the differences between deceptive and truthful reviews. Researchers have also applied neural network models, such as Convolutional Neural Networks [40] and Recurrent Neural Networks [29], to learn features automatically. The core idea of detecting deceptive reviews is to build a classifier to classify reviews as fake or true reviews, or to generate the probability that they are fake or true. The machine learning method of learning to classify can be divided into supervised learning, unsupervised learning, and semi-supervised learning. Previous research using supervised learning mainly used logistic regression [10], Naïve Bayes [8], and Support Vector Machines [19] as classifiers. Unlike supervised models, unsupervised learning methods [28] do not require labelled samples, as it can be hard to accurately build the labelled dataset. Researchers [6, 31] adopted semi-supervised learning methods when it was possible to gain a small amount of reliable labelled data. For review datasets, existing research mainly focuses on shopping, hotel, and restaurant reviews. After Ott et al. [21, 22] created a Gold-standard dataset, in which deceptive reviews were written by online hired workers on Amazon Mechanical Turk, many studies of review spam detection were conducted on this dataset [5, 30, 39].

In the research reported in this paper, we aimed to detect spam reviews in the domain of video games. Since online reviews are rich resources for collecting feedback about player experience for the video game industry [37], it is important to filter those fake reviews first. We crawled a video game review dataset from the popular game platform Steam. We analysed the unique features of video game reviews in the process of feature extracting. We labelled a small amount of positive data and used a semi-supervised learning method to identify the deceptive reviews. The results revealed that our method performed well with only a few positive samples.

2 RELATED WORK

Previous research has involved developing methods to detect spam reviewers on e-commerce websites and social media, as this can alleviate negative impacts imposed on customers and product manufacturers. As mentioned in the Introduction, Jindal and Liu [9] defined three different types of spam reviews: 1) untruthful reviews, 2) brand-only reviews, and 3) non-reviews. Type 2 (brand-only) and type 3 (non-reviews) can be collected manually due to obvious features and applied Naïve Bayes, Support Vector Machines, and logistic regression to identify these two types of deceptive reviews. Type 1 (untruthful) is difficult to detect as one can carefully craft a deceptive review to imitate genuine reviews. Jindal and Liu [9] solved this problem by regarding duplicate and near-duplicate reviews as almost certainly spam reviews and using them to build spam detection models. The results showed an average AUC value of 78%. The features they used to represent a review included the content of the review, the reviewer, and the product.

Other research has focused on finding effective features to represent deceptive reviews [25]. These features can be categorised into review-centric features and reviewer-centric features. The review-centric features include n-grams, part-of-speech (POS) tagging, textual features, linguistic features, sentiments, and rating-related features [34]. Some research [9, 21] modelled the content of the review with n-gram features. Other studies [15, 19, 22] combined n-gram with POS tagging and Linguistic Inquiry and Word Count (LIWC) features [24]. The text analysis tool LIWC can be used to extract multiple features according to the research requirements, such as linguistic features and sentiments.

Li et al. [16] introduced a neural network based model for assigning weights to reviews at the whole sentence and document level. Their method revealed the latent semantic meaning of the user's review. The results showed their method outperformed the state-of-the-art model.

The reviewer-centric features include the behaviors and characteristics of the person who wrote the review. Wang et al. [36] created a heterogeneous review graph to capture relationships among reviewers, reviews, and online stores. Based on this, they developed a computation method to calculate the reliability of those nodes in the graph. The elements of their computation model included ratings, the posting time of the review, and the number of the reviewer's comments. Hussain et al. [8] utilised thirteen different spammer's behavioral features (the ratio of negative reviews, rating deviation, and so on) to calculate the review spam core so as to detect spam reviews and spammers. Li et al. [14] concluded that reviewers' posting rates (number of reviews written in a period of time) are bi-modal and many spammers like to collectively write reviews for the same set of products within a short time frame (co-bursting). They applied a Hidden Markov Model to model spamming by only utilising reviewers' comment posting times. Dematis et al. [3] combined spam indicators of the rating deviation, the number of the reviewer's reviews, the content similarity of reviews, as well as burst patterns to distinguish spam reviews from honest ones.

3 POSITIVE-UNLABELLED LEARNING

Positive-Unlabelled Learning is one kind of semi-supervised approach. It is different to supervised learning methods, which need positive samples and negative samples. Nigam et al. [20] identified that unlabelled data is helpful in classifier building. They used a small set of labelled documents and a large set of unlabelled documents to build classifiers. They showed that this approach is better than using the small set of labelled documents alone. It provides the basic theory of PU-learning. Since PU-learning is ideal for cases where vast amounts of unlabeled data exist, some studies [3, 5, 7] detected deceptive reviews based on PU-learning approaches. Rout et al. [30] utilised four semi-supervised learning methods (Co-training, expectation maximisation algorithm, label propagation and spreading and PU learning) for online deceptive review detection. They concluded that high performance levels were achieved when using PU learning-based classification.

In this paper, we extracted stylistic features, POS tagging, and sentiment features. We also considered the semantic features of the whole review by using the doc2vec method. In addition, we created unique features based on the characteristics of Steam reviews and employed them to build a small number of positive samples. Features from the five approaches were used to train a Decision Tree classifier. In our research, in order to supplement current frameworks of spam review detection, we adopted a semi-supervised learning method, PU learning, to detect deceptive video game reviews.

4 FEATURE EXTRACTION

Previous research has extracted various features from deceptive reviews, including stylistic features, part-of-speech features, sentiment features, raw data features, and doc2vec. In this section, we analyse and discuss these features for detecting deceptive reviews. We also discuss some unique features of Steam game reviews.

4.1 Stylistic Features

Stylistic features are mainly used to describe the user's writing style. According to early analysis of truthful writing and imaginative writing [1, 26], deceptive and truthful reviews are different in writing style. For example, deceptive reviewers use simpler, short, and fewer average syllables per word in compare with real reviewers [2]. Therefore, finding

Table 1. Lexical Features

Lexical Features	Description	How to detect spam reviews
Total number of numbers	All numbers (e.g., 1, 10, 100)	The more numbers, the less semantic information the review contains, and the more likely to be a deceptive review.
Length of review tokens (T)	All tokens including words, symbols, punctuation	The shorter the length of review, the weaker the credibility.
Ratio of total number of first-person words	(The number of first-person words: I, my, mine, our, ours, we, us, me) / T	The more first-person words, the more likely it is a spam review.
Total number of characters (N)	All letters Aa-Zz, all single digits, all punctuation marks, all symbols	Same as above.
Ratio of total number of uppercase letters	(The number of upper letters A-Z) / N	Capitalised letters generally indicate emphasis or strong tone which express a strong emotion. The bigger the ratio, the stronger the emotion.
Ratio of total number of digits and symbols	(The number of digits, punctuation, special symbols ($#%&̂)) / N	The larger the ratio, the less semantic information the review contains. A ratio of 1 means the review only contains symbols or digits. We can regard it as a spam review.
Total number of short words	All short words (1-3 characters): if, the, how	Same as above.

Table 2. Syntactic Features

Syntactic Features	Description	How to detect spam reviews
Total number of punctuation symbols	. ? ! : ; ‘ “	The higher the number, the less semantic information the review contains.
Total number of function words	Using list of words in Appendix A	Same as above.

features of different writing styles is the main goal of stylistic features. Stylistic features can be categorised into two types [32], lexical features and syntactic features. Table 1 presents lexical features and Table 2 shows syntactic features we used in this research.

4.2 Part-of-Speech Features

Part-of-speech features are generated by tagging words by part-of-speech and counting the frequency. Li et al. [15] found that truthful and deceptive reviews vary by the frequency of different part-of-speech features used. Truthful reviews contain more words that are tagged as noun, adjective, preposition, qualifier, and conjunction. Deceptive reviews contain more words that are tagged as verb, adverb, pronoun, and pre-qualifier. These characteristics are consistent with early analyses of truthful writing and imaginative writing [1, 26]. However, part-of-speech features have certain limitations. Deceptive reviews fabricated by experts do not satisfy this rule. The purpose of imitating

truthful reviews is stronger when experts write reviews. They imitate characteristics of truthful reviews from details of product information, consumption experience, and so on, which is more convincing.

4.3 Sentiment Features

According to Li et al. [15], deceptive reviews contain more emotional words than truthful reviews, which means that deceptive reviews are more positive or negative than truthful reviews. The goal of spam reviewers is to advocate or discredit an object. The use of emotional words in reviews can show and enhance the emotional polarity. Therefore, sentiment analysis is necessary for detecting deceptive reviews: reviews with strong emotion are more likely to be deceptive. Mainstream sentiment analysis methods include sentiment dictionaries and machine learning techniques.

Sentiment dictionary methods utilise a dictionary which contains emotional words marked with a polarity value. For each input sentence, all emotional words in it will be picked out and the polarity value for the whole sentence is returned by certain mathematical calculation methods. This method is intuitive and easy to operate, while it is limited by the initial sentiment dictionary. However, it ignores the rules of word order and syntax. Sentiment analysis based on machine learning methods mainly utilises classification technologies to judge the sentiment orientation of sentences. It trains on labelled datasets. This method has performs differently in various review datasets that come from different domains and sources.

4.4 Raw Data Features

The raw data of the review refers to characteristics of the review apart from the text content. Each review record in our game review dataset includes nickname, rating, play duration, the number of people who think the review is help or funny, and the content of review. After extracting features of the text content by using the above methods, we can also obtain potential raw data. In this research, the raw-data features can detect deceptive reviews directly. Raw-data features include play duration and sentiment polarity, rating and sentiment polarity, and confidence.

(1) Play duration and sentiment polarity: according to Steam user policy, players can request a refund if the play duration for one game is less than 2 hours. As few spam reviewers will actually buy the product, the play duration can be helpful for detecting deceptive reviews.
(2) Rating and sentiment polarity: generally, the rating can represent the attitude or emotion of players. When there is a contradiction between the rating and sentiment polarity, this review is more likely to be deceptive.
(3) Confidence: this value is calculated by considering the factor of play duration, the number of people who think the review is helpful or funny, and the length of the review. By inspecting the game review dataset, we found that longer reviews generally have more helpful votes. Confidence is a composite value which is used to indicate the credibility of each review.

4.5 Doc2Vec

Doc2vec, or paragraph2vec, sentence embeddings is an unsupervised algorithm which can get the vector expression of sentences/paragraphs/documents. Le and Mikolov [12] proposed doc2vec as an extension of word2vec [18] in 2014. The training results of word2vec, word embeddings, can measure the similarity between words and words. Likewise, the training results of doc2vec can measure the similarity between documents and documents. Doc2vec also has two different implementations called distributed memory model and distributed bag of words model. The core idea of the

former is the centre word can be predicted according to other words near it, while the latter predicts a random word in the paragraph by inputting the paragraph vector.

In this section, we covered all features we have used in this research. Different features can be applied to detect different types of spam reviews, which will be introduced and explained in the next section.

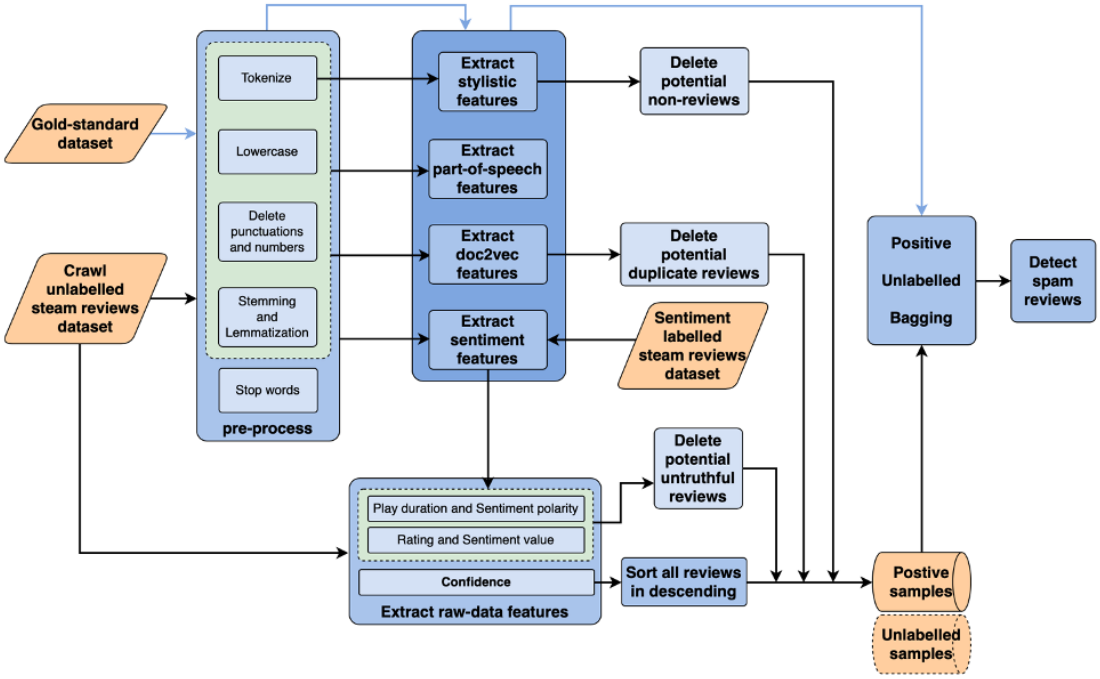

Fig. 1. Framework for feature-based spam game reviews detection.

5 FRAMEWORK

Figure 1 illustrates the framework of our method for identifying spam game reviews. In this research, we made use of three review datasets: (1) 33,000 unlabelled Steam game reviews we collected from 8 different games, (2) the Gold-Standard deceptive opinion dataset comprised of 1,600 labelled reviews from various websites, and (3) 100,000 Steam game reviews with sentiment labels.

The unlabelled steam review dataset was applied as the main dataset for analysing features and detecting spam game reviews. It was collected from 8 different games: GTA, PUBG, Oxygen Not Included, Total War: Three Kingdoms, NBA 2k19, Just Cause 4, Scum, and For Honor. The reasons for selecting these games were:

(1) They are very popular, so we can crawl enough reviews.
(2) They are different types of games. The game review dataset we crawl should cover different game types as much as possible so as to represent the whole game dataset.
(3) In order to balance the effect of positive and negative reviews, the eight games have different ratings ranging from very good to very poor.

There are 33,450 reviews in this dataset. For each review, it contains nickname, rating, play duration, the number of people who think the review is helpful or funny, and the content of review.

The Gold-standard dataset[2]is publicly available. All the reviews in this dataset are labelled with truthful or deceptive. The dataset consists of 1,600 reviews, including:

- 400 truthful positive reviews from TripAdvisor
- 400 truthful negative reviews from TripAdvisor, Orbitz, Expedia, Hotels.com, Priceline and Yelp
- 400 deceptive positive reviews from Mechanical Turk
- 400 deceptive negative reviews from Mechanical Turk.

The Steam game reviews with sentiment label dataset[3]is publicly available and is labelled with positive and negative labels. We used a sub-dataset, which has 100,000 reviews, and each review includes game id, content of reviews, sentiment label, and number players who think the review is helpful. We used this dataset to train the sentiment value of reviews.

5.1 Pre-processing Dataset

After constructing the raw dataset, we performed preliminary processing so that it could be used to extract various features. The pre-processing methods included tokenising, lowercase, replacing punctuation and numbers with space or no-space, deleting stop words, stemming, and lemmatisation.

5.2 Extracting Stylistic Features

Stylistic features are based on tokens in the content of reviews. The numbers, punctuation, and some special should be analysed in stylistic features. Therefore, we did not apply lowercase, replacing punctuation and numbers with space or no-space, deleting stop words, and stemming and lemmatisation techniques for stylistic features.

5.3 Extracting Part-of-Speech Features

According to Li et al. [15], the ratios of different part-of-speech features are different in truthful and deceptive reviews. However, the stop word list contains some words that we need to utilise. Therefore, we applied all pre-processing techniques except deleting stop words to generate part-of-speech features. We defined all words that are tagged as noun, adjective, preposition, qualifier, and conjunction as true words. The verb, adverb, pronoun and pre-qualifier are false words. We then counted the number of true words and false words for each review. The part-of-speech features include frequency of true words (FT), frequency of false words (FF), and the ratio of false to true word frequency (FF/FT).

5.4 Extracting Sentiment Features

In this research, we applied sentiment dictionary (Python inbuilt package TextBlob) and machine learning method (XGBoost) together to obtain the final polarity value for each review. TextBlob returned a polarity value. We randomly chose 100,000 reviews from the labelled steam dataset to train XGBoost. The classifier returned a label of 1 (positive) or 0 (negative). We balanced the two methods' results by adjusting the polarity value of those reviews:

(1) Reviews in which the polarity value is bigger than 0 but the label is 0: we reduced its polarity value by 0.3.
(2) Reviews in which the polarity value is smaller than 0 but the label is 1: we increased its polarity value by 0.3.

[2]https://myleott.com/op-spam
[3]https://zenodo.org/record/1000885#.X3wpny-1GJ8

5.5 Extracting Raw-Data Features

In this research, we extracted three raw-data features: (1) Play duration and sentiment polarity, (2) rating and sentiment polarity, and (3) confidence.

5.5.1 Play duration and sentiment polarity. We searched all reviews in which the play duration is less than 2 hours and set two thresholds, T1 and T2, to 0.5 and -0.75, respectively (see Algorithm 1). The reasons are:

- Generally, the play duration reflects the player's attitude towards the game. When the play duration is less than 2 hours, it usually means the game does not attract players and the polarity of the review may be negative. Thus, the review is more likely to be untruthful if the polarity value of the review is bigger than 0.5. Considering that some players just cannot wait to write a review, their polarity values can be positive. But if the value is bigger than 0.5, it should be abnormal.
- Meanwhile, we still need a threshold T2 even if this kind of review is generally negative. Thus, if the polarity value is lower than -0.75, we still believe that the player has malicious intent.

Algorithm 1 Play duration and sentiment polarity

```
for each review i do
    if play duration of i < 2 then
        if polarity value of i >= T1 or polarity value of i <= T2 then
            Set a label 1 for i
        else
            Set a label 0 for i
        end if
    end if
end for
```

Algorithm 2 Rating and sentiment polarity

```
for each review i do
    if rating i == 1 then
        if polarity value of i < T1 then
            Set a label 1 for i
        else
            Set a label 0 for i
        end if
    end if
    if rating i == 0 then
        if polarity value of i > T2 then
            Set a label 1 for i
        else
            Set a label 0 for i
        end if
    end if
end for
```

5.5.2 Rating and sentiment polarity. Rating has two types: recommended (1) and not recommended (0). The purpose of rating is to show the player's attitude towards the game. We selected all reviews where the rating is different than its polarity value and set two thresholds, T1 and T2, to -0.5 and 0.5, respectively (see Algorithm 2. The reasons are:

- In real commentary activities, it may appear that users complain or suggest certain aspects of the product in reviews, but this does not mean that they do not recommend the product. Imagine that only one update of the game dissatisfied the player and they left a negative review. However, they still love this game and would like to recommend it. In this case, we cannot regard this review as an untruthful review. Similarly, users might not recommend a game, but might consider one or two highlights of the game and are willing to mention those in reviews.
- The above cases can exist in game reviews. However, the polarity value opposite to the rating should be within a reasonable range. Therefore, we set T1 to -0.5 and T2 to 0.5.

5.5.3 Confidence. Confidence is a combination of several raw factors. According to the description of stylistic features, the length of reviews can influence their credibility. Meanwhile, the voting factors (helpful/funny) also need to be considered when detecting untruthful reviews. Therefore, we defined the confidence for each review as:

confidence = play duration * N(length of review) * (helpful + 0.1) * [(funny + 0.1) / 2]

N(x) means that *x* should be normalised into the range (0, 1). We added 0.1 for helpful and funny to avoid a value of 0. Moreover, we set the weight of helpful to be higher than that of funny as the helpful factor is more important when determining the credibility of reviews.

5.6 Extracting Doc2Vec Features

We applied all pre-processing techniques except deleting stop words to train two doc2vec models: Doc2vec_dm and Doc2vec_dbow. The training results of the two models were combined by stacking the array horizontally (in column order) to get the vector for each review. This vector contains characteristics of both models. We regard it as the doc2vec feature for each review. Meanwhile, we made each model return the most similar review for each review. We then applied an algorithm as shown to find duplicate reviews (see Algorithm 3.

In this algorithm, we set t1 and t2 as 0.95 and 0.9, respectively, after several tests on different values. We found that many reviews in Steam are very short, like "nice game" and "good game". Although these reviews contain very limited information due to the length, they cannot simply be regarded as duplicate reviews. Therefore, we set the length of reviews greater than 5 in this algorithm. Meanwhile, we also defined that the difference in the length of similar reviews should not be greater than 5.

5.7 Applying PU Learning to Detect Spam Reviews

The non-reviews, duplicate reviews, and explicit untruthful reviews are easy to find, while implicit untruthful reviews cannot be detected directly through the features above. Therefore, the final step of this research is applying the classifier to detect those implicit untruthful reviews. Normally, we would utilise a supervised learning approach with truthful and untruthful samples to train the classifier. But in this case, we cannot obtain reliable spam reviews and only have truthful reviews detected by a high confidence factor. As such, we applied a semi-supervised learning technique, Positive Unlabelled (PU) Learning, to detect implicit untruthful reviews and regarded reliable truthful reviews as positive samples.

Algorithm 3 Detecting duplicate reviews

```
for each review i do
    if length of i > 5 then
        Get the most similar review x based on Doc2vec_dm
        Get the most similar review y based on Doc2vec_dbow
        if similarity between i and x > t1 then
            if length |x-i| < 5 and polarity of x is same as i then
                if nickname of x is different from i then return i.index, x.index
                end if
            end if
        end if
        if similarity between i and y > t2 then
            if length |y-i| < 5 and polarity of y is same as i then
                if nickname of y is different from i then return i.index, y.index
                end if
            end if
        end if
    end if
end for
```

Using the features defined in the previous sections, we generated a positive samples dataset using the following process:

(1) We sorted all reviews in descending order of the confidence value. Therefore, the top reviews are highly credible comments.
(2) Then, we applied stylistic features and raw-data features to delete all potential non-reviews and untruthful reviews, respectively. The duplicate reviews were deleted after extracting doc2vec features.
(3) Finally, we selected 2,000 reviews from the first quarter of reviews, 1,000 reviews from the second quarter of reviews, and 500 reviews from the third quarter of reviews. The reason we selected reviews from different parts was to avoid the influence of the confidence factor on the final result.

All the remaining samples were regarded as the unlabelled samples dataset. We utilised the PU-bagging technique to train decision tree classifiers. We applied all features (except the raw-data features) to reviews in the Gold-Standard dataset. We performed the PU-bagging on the Gold-standard dataset with different sizes of positive samples and 520 unlabelled samples to verify the validity of PU-bagging.

The PU-bagging technique can be summarised in the following steps:

(1) Create a training dataset by combing all positive samples and random unlabelled samples.
(2) Use "bootstrap" samples to build a classifier. Bootstrap includes all positive samples and a random unlabelled samples dataset that has the same number as the positive dataset.
(3) Apply a classifier to classify the samples that are not in bootstrap and record the probability.
(4) Repeat the above three steps. The final probability is the average value of all iteration.

6 EXPERIMENTAL EVALUATION

As mentioned previously, we divided spam game reviews into three different types: non-reviews, duplicate reviews, and untruthful reviews. In this section, we report our experimental results of using features to detect these three types of spam reviews.

6.1 Non-Reviews

For non-reviews, the stylistic features show a high number in ratio of digits, symbols, and punctuation. Example non-reviews from our dataset include: ".", "!!", "10/10", and "123818237980801312/1". There were 337 reviews that belong to non-reviews. Such reviews account for a small percentage of the dataset and are easy to detect manually.

6.2 Duplicate Reviews

In this research, we mainly use doc2vec features to detect duplicate reviews. The performance of detecting duplicate reviews is greatly influenced by the settings in the experiment. If we changed the limited length from >5 to >0, there would be far more duplicate and short reviews in the results. Therefore, the result of detecting duplicate reviews is also affected by the overall distribution of review length. There were 1,221 groups of reviews labelled as duplicate reviews. One group contains two (matching) reviews.

6.3 Untruthful Reviews

Untruthful reviews can be divided into two different types: explicit untruthful reviews and implicit untruthful reviews. We can detect explicit untruthful reviews by raw-data features. For implicit untruthful reviews, it is much more difficult to detect. All defined features were used as input for our semi-supervised learning technique to detect implicit untruthful reviews.

For explicit untruthful reviews, there were 3 records. The play duration of first two reviews were less than 2 hours. Their content is short and contains no word after pre-processing. Moreover, negative reviews without explanations also have low credibility. The third review was detected as untruthful reviews as the recommendation is different to the polarity value. The player does not recommend this game, while they write a positive review with a polarity value larger than 0.3. Therefore, we regarded it as an untruthful review.

6.4 PU-Bagging

Since there is no publicly labelled game review dataset (labelled with truthful and untruthful), we applied the PU-bagging method based on a decision tree to the Gold-Standard dataset to verify the validity of our proposed method. All features except the raw-data features (because the Gold-Standard dataset has no attributes for generating raw-data features) were used to present the reviews in the Gold-Standard dataset. The dataset includes 400 truthful reviews and 400 deceptive reviews. We used different sizes (20, 40, 60, 80, 100, 120) of positive samples to train the PU-bagging method. The results are shown in Table 3.

We compared our experimental results with Hernandez et al.'s [5] work. Our best result was obtained when the size of positive samples was 100, which was the same as their evaluation result. While Hernandez et al.'s maximum F1 score (83.7%) was better than ours (80.0%), our proposed method still outperformed theirs in the remaining cases (20, 40, 60, 80, 120 positive samples). It is worth noting that our method performed better when there were fewer positive samples. Our method also outperformed human accuracy of around 60% for all sample sizes. Finally, the precision scores were

Table 3. The performance of PU-bagging when using 20, 40, 60, 80, 100, and 120 examples of deceptive reviews for training. D refers to deceptive reviews and U to unlabeled reviews.

Training Set	Work	Precision	Recall	F1
20-D	Hernandez et al. [5]	100.0%	2.5%	4.9%
520-U	**Proposed Method**	56.4%	93.8%	**70.4%**
40-D	Hernandez et al. [5]	77.8%	28.8%	15.7%
520-U	**Proposed Method**	62.7%	86.3%	**72.6%**
60-D	Hernandez et al. [5]	91.7%	27.5%	42.3%
520-U	**Proposed Method**	66.4%	91.2%	**76.8%**
80-D	Hernandez et al. [5]	86.8%	41.3%	55.9%
520-U	**Proposed Method**	69.2%	92.5%	**79.1%**
100-D	Hernandez et al. [5]	78.3%	90.0%	**83.7%**
520-U	**Proposed Method**	72.0%	90.0%	80.0%
120-D	Hernandez et al. [5]	78.9%	78.1%	78.0%
520-U	**Proposed Method**	72.4%	88.7%	**79.8%**

worse than recall scores in this experiment. This means our method can identify almost all positive samples, but it also treated some negative samples as positive samples.

The features in this research cannot best represent reviews in the Gold-Standard dataset. All these features were generated by considering game reviews, not hotel reviews. For example, there were no short reviews and irregular statements (many symbols and punctuation) in the Gold-Standard dataset. Therefore, some features for detecting deceptive hotel reviews were not helpful and perhaps even counterproductive. Moreover, the research has demonstrated that raw-data features can improve the accuracy of the classifier [13]. In the last stage, we applied the proposed method to predict the unlabelled Steam review dataset. There were 5,021 of the 33,450 reviews labelled as spam reviews, or approximately 15%. This falls within the expected range of 10-20% and maps to the Yelp figures of 14-20% of reviews are spam.

7 CONCLUSIONS AND FUTURE WORK

In this paper we analysed features of game reviews and utilised a semi-supervised method to detect spam reviews. By comparing with the previous research, we verified the validity of our method. When there are not many positive samples, our proposed method has much better performance. Although this method incorrectly labels negative samples as positive samples, it performs well at identifying positive samples. We think this is more important in the context of detecting spam reviews. The spam reviews have a great negative impact on user behavior and the reliability of the review platform. Therefore, we should try to identify all the spam reviews, even if some truthful reviews are regarded as spam. In general, our method is more suitable for scenarios where there are few positive samples and the positive ones have a substantial impact on the system.

In future work, we plan to further analyse reviews that only contain symbols or numbers. In this research, we detected many abnormal reviews, like "**********" and "ksadfklasjdfkljsdfasd fasdfasdfasdf" as non-reviews, while some reviews did have their own unique meaning. For example, the review "10/10" can be interpreted to mean the same as the sentence "the game is perfect". Therefore, some special numbers and emojis will be considered as special reviews. We also plan to manually annotate the Steam game review dataset with spam and truthful labels. In the future work,

we will evaluate the performance of our method on the Steam game review dataset and try to build a spam labelled game review dataset.

REFERENCES

[1] David B Buller and Judee K Burgoon. 1996. Interpersonal deception theory. *Communication theory* 6, 3 (1996), 203–242.

[2] Judee K Burgoon, J Pete Blair, Tiantian Qin, and Jay F Nunamaker. 2003. Detecting deception through linguistic analysis. In *International Conference on Intelligence and Security Informatics*. Springer, 91–101.

[3] Ioannis Dematis, Eirini Karapistoli, and Athena Vakali. 2018. Fake review detection via exploitation of spam indicators and reviewer behavior characteristics. In *International Conference on Current Trends in Theory and Practice of Informatics*. Springer, 581–595.

[4] Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. 2013. Exploiting burstiness in reviews for review spammer detection. *Icwsm* 13 (2013), 175–184.

[5] Donato Hernández Fusilier, Rafael Guzmán Cabrera, Manuel Montes, and Paolo Rosso. 2013. Using PU-learning to detect deceptive opinion spam. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*. 38–45.

[6] Donato Hernández Fusilier, Manuel Montes-y Gómez, Paolo Rosso, and Rafael Guzmán Cabrera. 2015. Detecting positive and negative deceptive opinions using PU-learning. *Information processing & management* 51, 4 (2015), 433–443.

[7] Daojing He, Menghan Pan, Kai Hong, Yao Cheng, Sammy Chan, Xiaowen Liu, and Nadra Guizani. 2020. Fake Review Detection based on PU Learning and Behavior Density. *IEEE Network* (2020).

[8] N. Hussain, H. Turab Mirza, I. Hussain, F. Iqbal, and I. Memon. 2020. Spam Review Detection Using the Linguistic and Spammer Behavioral Methods. *IEEE Access* 8 (2020), 53801–53816. https://doi.org/10.1109/ACCESS.2020.2979226

[9] Nitin Jindal and Bing Liu. 2008. Opinion Spam and Analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining* (Palo Alto, California, USA) *(WSDM '08)*. Association for Computing Machinery, New York, NY, USA, 219–230. https://doi.org/10.1145/1341531.1341560

[10] Nitin Jindal, Bing Liu, and Ee-Peng Lim. 2010. Finding Unusual Review Patterns Using Unexpected Rules. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (Toronto, ON, Canada) *(CIKM '10)*. Association for Computing Machinery, New York, NY, USA, 1549–1552. https://doi.org/10.1145/1871437.1871669

[11] Seongsoon Kim, Hyeokyoon Chang, Seongwoon Lee, Minhwan Yu, and Jaewoo Kang. 2015. Deep semantic frame-based deceptive opinion spam analysis. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. 1131–1140.

[12] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*. 1188–1196.

[13] Fangtao Huang Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. 2011. Learning to identify review spam. In *Twenty-second international joint conference on artificial intelligence*.

[14] Huayi Li, Geli Fei, Shuai Wang, Bing Liu, Weixiang Shao, Arjun Mukherjee, and Jidong Shao. 2017. Bimodal distribution and co-bursting in review spam detection. In *Proceedings of the 26th International Conference on World Wide Web*. 1063–1072.

[15] Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. 2014. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1566–1576.

[16] Luyang Li, Bing Qin, Wenjing Ren, and Ting Liu. 2017. Document representation and feature combination for deceptive spam review detection. *Neurocomputing* 254 (2017), 33 – 41. https://doi.org/10.1016/j.neucom.2016.10.080 Recent Advances in Semantic Computing and Personalization.

[17] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting Product Review Spammers Using Rating Behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management* (Toronto, ON, Canada) *(CIKM '10)*. Association for Computing Machinery, New York, NY, USA, 939–948. https://doi.org/10.1145/1871437.1871557

[18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[19] Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie S Glance. 2013. What yelp fake review filter might be doing?. In *Icwsm*. 409–418.

[20] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell. 2000. Text classification from labeled and unlabeled documents using EM. *Machine learning* 39, 2-3 (2000), 103–134.

[21] Myle Ott, Claire Cardie, and Jeffrey T Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*. 497–501.

[22] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. *arXiv preprint arXiv:1107.4557* (2011).

[23] Dhairya Patel, Aishwerya Kapoor, and Sameet Sonawane. 2018. Fake Review Detection using Opinion Mining. *International Research Journal of Engineering and Technology (IRJET)* 5 (2018).

[24] James W Pennebaker, Ryan L Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. Technical Report.

[25] Ajay Rastogi, Monica Mehrotra, and Syed Shafat Ali. 2020. Effective Opinion Spam Detection: A Study on Review Metadata Versus Content. *Journal of Data and Information Science* 5, 2 (2020), 76–110.

[26] Paul Rayson, Andrew Wilson, and Geoffrey Leech. 2002. Grammatical word class variation within the British National Corpus sampler. In *New Frontiers of Corpus Research*. Brill Rodopi, 295–306.

[27] Y. Ren and D. Ji. 2019. Learning to Detect Deceptive Opinion Spam: A Survey. *IEEE Access* 7 (2019), 42934–42945. https://doi.org/10.1109/ACCESS.2019.2908495

[28] Yafeng Ren, Lan Yin, and Donghong Ji. 2014. Deceptive reviews detection based on language structure and sentiment polarity. *Journal of Frontiers of Computer Science and Technology* 8, 3 (2014), 313–320.

[29] Yafeng Ren and Yue Zhang. 2016. Deceptive opinion spam detection using neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. 140–150.

[30] Jitendra Kumar Rout, Anmol Dalmia, Kim-Kwang Raymond Choo, Sambit Bakshi, and Sanjay Kumar Jena. 2017. Revisiting semi-supervised learning for online deceptive review detection. *IEEE Access* 5 (2017), 1319–1327.

[31] Jitendra Kumar Rout, Smriti Singh, Sanjay Kumar Jena, and Sambit Bakshi. 2017. Deceptive review detection using labeled and unlabeled data. *Multimedia Tools and Applications* 76, 3 (2017), 3187–3211.

[32] Somayeh Shojaee, Masrah Azrifah Azmi Murad, Azreen Bin Azman, Nurfadhlina Mohd Sharef, and Samaneh Nadali. 2013. Detecting deceptive reviews using lexical and syntactic features. In *2013 13th International Conference on Intellient Systems Design and Applications*. IEEE, 53–58.

[33] Gina A. Tran and David Strutton. 2020. Comparing email and SNS users: Investigating e-servicescape, customer reviews, trust, loyalty and E-WOM. *Journal of Retailing and Consumer Services* 53, C (2020). https://doi.org/10.1016/j.jretconser.2019

[34] Dushyanthi U Vidanagama, Thushari P Silva, and Asoka S Karunananda. 2020. Deceptive consumer review detection: a survey. *Artificial Intelligence Review* 53, 2 (2020), 1323–1352.

[35] Bettina von Helversen, Katarzyna Abramczuk, Wiesław Kopeć, and Radoslaw Nielek. 2018. Influence of consumer reviews on online purchasing decisions in older and younger adults. *Decision Support Systems* 113 (2018), 1 – 10. https://doi.org/10.1016/j.dss.2018.05.006

[36] Guan Wang, Sihong Xie, Bing Liu, and S Yu Philip. 2011. Review graph based online store review spammer detection. In *2011 IEEE 11th International Conference on Data Mining*. IEEE, 1242–1247.

[37] Xiaohui Wang and Dion Hoe-Lian Goh. 2020. Components of game experience: An automatic text analysis of online reviews. *Entertainment Computing* 33 (2020), 100338.

[38] Lan You, Qingxi Peng, Zenggang Xiong, Du He, Meikang Qiu, and Xuemin Zhang. 2020. Integrating aspect analysis and local outlier factor for intelligent review spam detection. *Future Generation Computer Systems* 102 (2020), 163–172.

[39] Wen Zhang, Chaoqi Bu, Taketoshi Yoshida, and Siguang Zhang. 2016. CoSpa: A co-training approach for spam review identification with support vector machine. *Information* 7, 1 (2016), 12.

[40] Siyuan Zhao, Zhiwei Xu, Limin Liu, Mengjie Guo, and Jing Yun. 2018. Towards accurate deceptive opinions detection based on word order-preserving CNN. *Mathematical Problems in Engineering* 2018 (2018).

A FUNCTION WORDS

a between in nor some upon about both including nothing somebody us above but inside of someone used after by into off something via all can is on such we although cos it once than what am do its one that whatever among down latter onto the when an each less opposite their where and either like or them whether another enough little our these which any every lots outside they while anybody everybody many over this who anyone everyone me own those whoever anything everything more past though whom are few most per through whose around following much plenty till will as for must plus to with at from my regarding toward within be have near same towards without because he need several under worth before her neither she unless would behind him no should unlike yes below i nobody since until you beside if none so up your