A close up of a logo

Description automatically generated

**A Study on Application of NLP practices for detection of fake reviews for Hotels**

**Submitted as part of the 2020 Summer internship at SmartBridge**

**Mohammad Kharoof** (Computer Engineering)

**Obada Issa** (Computer Science)

**Yara Rashed** (Computer Engineering)

Table of Contents

[Introduction 3](#_Toc47805540)

[Objectives 4](#_Toc47805541)

[Problem Statement 5](#_Toc47805542)

[Review of Literature 6](#_Toc47805543)

[**Source (1) [1]:** 6](#_Toc47805544)

[**Source (2) [2]:** 7](#_Toc47805545)

[**Source (3) [3]:** 9](#_Toc47805546)

[**Source (4) [4]:** 11](#_Toc47805547)

[Data Collection 13](#_Toc47805548)

[Data Analysis and Methodology 15](#_Toc47805549)

[Findings and Suggestions 21](#_Toc47805550)

[Conclusion 22](#_Toc47805551)

[Table of figures 23](#_Toc47805552)

[References and Bibliography 24](#_Toc47805553)

## Introduction

In the past few years, the effect of online feedback on business has evolved dramatically to assess business performance in a wide variety of industries, from restaurants, hotels and e-commerce. Some entities use unethical means to enhance their reputation by posting fake reviews of their companies or competitors. The detection of true and false reviews has become of great importance for the hotel industry to survive. The proposition here is a feature framework for analyzing fake feedback evaluated in the user domain of the hotels.

Data science has improved exponentially over the last years and it has opened the doors to multiple ways of engineering all of that data. One way was to apply machine learning to the data in order to predict something based on an existing dataset in a set of algorithms called unsupervised machine learning algorithms.

Sentiment analysis is part of the Natural Language Processing ( NLP) techniques consisting of extracting emotions related to some raw texts. Typically this is based on social media posts and customer feedback to immediately understand why other people are positive or negative, and why. The aim of this study is to show how python can be used to perform sentiment analysis.

The goal here is to process the dataset given and come up with an NLP model that can predict how good a certain review is.

## Objectives

1. To understand the type of review shared by the customers.
2. To process the dataset provided.
3. Apply NLP on the dataset.
4. Perform sentiment analysis on the reviews.
5. Discussion of Results.

## Problem Statement

Detection of fake customer review has drawn significant attention in recent years due to the growing number of Internet transactions. Existing methods for detecting fake customer reviews use the review material, product and review information, and other features to identify fake reviews. However, as shown in recent studies, for text classification, the semantic sense of the reviews may be especially important. Additionally, the feelings hidden in the comments can be another possible indicator of fake content.

Analyzing reviews left to you by customers helps keep your feet on the ground in respect to overall customer satisfaction, as they can provide your business with feedback regarding what your customers truly want. By using this information as input, you will be able to improve customer service by quickly and efficiently resolve the issues that consumers faced, thereby creating a positive experience for the consumer and keeping your focus on their needs.

No doubt, we are social creatures since the moment we come to this world and we are interested in knowing what other say before we make our buying decisions. Much like we would ask friends and family for recommendations, review sites allow us to do this online with just some clicks.

Reviews enable new businesses to stand shoulder to shoulder with more established competition, and potentially gain a positive niche in people’s estimation and expectations. Look at it this way…which company would you rather buy from: one with 50 3-star reviews or one with 5 5-star reviews? Voila! You just took the discussion away from the discount and price!

## Review of Literature

### **Source (1) [1]:**

**Problem:**

This project is similar to the one this paper is presenting. The researcher, Anupam Chugh (2019), was trying to figure out a method to classify movie reviews. His goal was to train the model so that he can make an iOS application to find out whether a certain review is positive or negative. The difficulty comes from the computers, as they do not understand the human language as people do.

**Solution and Methodology:**

Chugh has noticed that Apple has had some advancements in the field of NLP in 2019. These advancements were mainly in text classification and word tagging. Apple has also released transfer learning technique for text classifiers model training. This allows the model to differentiate the meaning of a word depending of the context of the sentence, which was always a struggle for machine learning researchers. As for the dataset, Chugh has gathered his data from Rotten Tomatoes dataset. Starting off with the EDA, Chugh changed the csv files into train and test dataframes. He used 0.8 train to 0.2 test ratio. He then used a Text Classifier Model Project in Create ML and added the training data there. After deployment, the accuracy score was 80% and was deployed through SwiftUI and the Core ML transfer learning.

**Limitations:**

Though transfer learning is better equipped for such semantic analysis, it takes more time to train than a maximum entropy algorithm. The researcher has almost no control over what is wanted be done as everything is already premade and it is hard to change anything to make any impact on the final results.

**Comparison:**

Chugh’s model was created using the Apple’s transfer learning. It has achieved an accuracy of 80% approximately. Although it was easy to make, it had many flaws, it needed a much longer time for almost one tenth of the data the team had to train. It took the team 2 hours to fully train the model, while Chugh’s model took 4 hours to train. This means that this approach is advisable when the dataset is smaller, semantic analysis is needed and there is no worry about the time the model takes to train.

### **Source (2) [2]:**

**Problem:**

Nowadays purchasing products online has become the norm as most people do not have the time physically wait and pay in queues. Since customers cannot inquire about a product before buying online, they read reviews to help them decide. There are generally two types of spam reviews. The first one consists of those that intentionally mislead readers which provide undeserving positive opinions to some products in order to promote them or by giving unfair negative reviews to some other products in order to lower their reputation. The second type consists of nonreviews which contain no opinions on the product and only scores or ratings. However, reviews containing true negative feedback should not be treated as spam. Thus, to make online reviews reliable, it has become a critical issue to distinguish spam reviews from real ones.

**Solution and Methodology:**

The proposed model is divided into four phases. The first phase comprises acquiring the data and preprocessing it. The second phase involves Active Learning Algorithm, through which gradually all the unlabeled data becomes labeled, while the learner measures the probability difference for correct classification to assure the quality of the dataset. Third stage is applying TF-IDF and n-grams techniques and for deep learning methods we have used TF -IDF for MLP and Word Embeddings techniques for both CNN and LSTM to represent texts as numerical values. The last phase is the spam detection phase. SVM, KNN and NB classifiers are applied to detect spam reviews on behalf of traditional machine learning techniques. MLP, CNN and RNN are the deep learning techniques used for spam detection. Finally, compare the performance of each classifier for both traditional machine learning and deep learning.

**Limitations:**

Deep learning algorithms need much more training data than conventional machine learning algorithms for spam review detection. Traditional machine learning algorithms like SVM and NB, on the other hand, reach a certain threshold where adding further training data does not increase their accuracy. The downside of deep learning for this issue is that only 1600 Ott Dataset review instances and first 2000 Yelp Dataset reviews instances are used. Because of this limited amount of data, the problem of overfitting could affect the accuracy of the detection. Word2Vec also lacks pre-training, which can significantly affect the accuracy of the test results.

**Comparison:**

Shahariar et al tried to do comprehensive trials to assess and compare how different algorithms of deep machine learning and traditional machine learning can result in different outcomes and accuracy scores. Our aim is much simpler than theirs but our approach proves more reliable in huge datasets (515k entries in our case) whereas the could not work with more than 2000 entries due to constraints in the methods they used and the power in the machines they had.

### **Source (3) [3]:**

**Problem:**

The problem was similar to the one this paper is researching. The researcher, Prateek Joshi (2018), was looking for an approach to mine online reviews. His goal was to train the model so that later he can automate moving the good comments to the top while the bad ones get to the bottom. This was hard to implement as computers do not understand spoken language the way humans do. Hence, the problem was inability to get the computer to understand and classify these reviews and ratings.

**Solution and Methodology:**

Joshi has used Amazon reviews to gather the data. Since each review is wired to a rating out of 5, that was simple enough to put it in a dataset. He also had other data such as ID of the user, ID of the item, how much this rating was helpful to the other customers, the actual rating, summary of the rating and the time of the review. After some feature engineering and cleaning, Joshi was able to start with the next most important step. He then decided to use the Natural Language Toolkit (NLTK) module, to tokenize the words. Tokenization is the key as machines are much better when dealing with numbers than words themselves. Finally, he has used the Latent Semantic Analysis (LSA), which makes a table of the tokenized data and classifies them based on their type (good or bad review).

**Limitations:**

LSA, even if it is easy to implement, fast and outputs great results, it is still a linear model which means it may not do just as great on datasets with non-linear dependencies. Not to mention that LSA is a Gaussian distribution of the terms in the documents, hence it may not be true all the time for all documents. Finally, when it comes to computational power, LSA involves a linear algebra term, Singular Value Decomposition (SVD), this term means the rotation, factorization, scaling and adding until the matrix is reduced to its desirable shape. This takes a lot of time and computational power, as the search can be from as little as O (1) to O (n!) which can be devastating for large datasets.

**Comparison:**

Joshi’s LSA model has worked despite the limitations and got an accuracy of 89.6% which is fascinating when talking about NLP models. However, this paper has the accuracy of 95.25% which was achieved using an ANN model. It is not that much in difference. Nevertheless, Joshi’s dataset was much cleaner the hotel\_reviews.csv that the team had to work on, the hotel reviews dataset had a much higher rating for positive than negative 95 positives to 5 negatives in every 100. This has obviously affected the team’s final accuracy score by a lot, since the model may have chosen to give a rate higher than 5 each time. Thus, giving it the accuracy of 95%.

### **Source (4) [4]:**

**Problem:**

It has become very common for people to read the opinions on the Web to address various decision-making issues. Most individuals and organizations often use online feedback to make their decisions, and fake reviews have become a big problem for them. The credibility of online review websites is also impaired by false reviews. As most recommendation systems depend on user feedback being right, these opinion spams often decrease the trustworthiness of the recommendation systems. Thus, opinion spam detection is a very important area of research which can also be considered a very active field of natural language processing science. This research work aims to identify whether or not a review is fake using an opinion spam detection model and to analyze the effect of different text features on the identification of opinion spam.

**Solution and Methodology:**

It is found that state-of-the-art methods give substantial attention to particular words. Additionally, this study uses Word Embedding as dense vectors which represents words. Since CNN architectures on several text classification datasets have obtained state-of-the-art results, this study proposes a model based on a recently proposed CNN model. Following are the stages proposed in this study for the opinion spam detection model: data preprocessing, feature representation, model construction and training, performance improvement.

**Limitations:**

In the real world, there is a huge number of reviews available in an online review portal. It is well known that CNNs functions best with large training sets and can make good generalizations compared to logistic regression, which is much simpler in terms of generalization. Some interesting results were shown by conducting experiments with a particular combination of function types. By concatenating the text-based features, it was possible to increase the efficiency of the proposed CNN model by some minor degree. Therefore, it needs to be further studied about the proportion of increase in computational ability and performance.

**Comparison:**

K. Archchitha and E. Charles use a similar approach to ours in terms of using ML to solve the issue of fake reviews, but in their implementation, they used CNN for a simpler and as a brute force solution for all reviewing platforms. Our project differs in a way where we are using NLP and with more accurate and consistent results. Also, we are taking the sentimental factor into consideration as well, meaning that we are not only depending on key words and phrases to reach our results, but rather using the human factor as well.

## Data Collection

This dataset contains 515,000 customer reviews and scoring of 1493 luxury hotels across Europe. Meanwhile, the geographical location of hotels are also provided for further analysis.

The dataset contains a collection of hotel reviews data. Each observation consists in one customer review for one hotel. A review is composed of a textual feedback of the customer’s experience at the hotel and an overall rating. Variables in the dataset include:

|  |  |
| --- | --- |
| **Variable** | **Meaning** |
| Hotel\_Address | Address of the hotel |
| Review\_Date | Date when review was posted |
| Average\_Score | Average score of hotel in the last year |
| Hotel\_Name | Name of the hotel |
| Reviewer\_Nationality | Nationality of the reviewer |
| Negative\_Review | Negative review given by reviewer; “No Negative” if left empty |
| ReviewTotalNegativeWordCounts | Total words in a negative review |
| Positive\_Review | Positive review given by reviewer; “No Positive” if left empty |
| ReviewTotalPositiveWordCounts | Total words in a positive review |
| Reviewer\_Score | Score given by reviewer |
| TotalNumberofReviewsReviewerHasGiven | Number of reviews given in the past |
| TotalNumberof\_Reviews | Total reviews of hotel |
| Tags | Tags given by reviewer |
| Dayssincereview | Duration between review date and scrape date |
| AdditionalNumberof\_Scoring | Number of scores gotten without reviews |
| Lat | Latitude of the hotel |
| Lng | Longitude of the hotel |

Figure 1: Dataset variable meanings

## Data Analysis and Methodology

Perform Sentiment analysis using the Natural Language Processing (NLP) techniques to extract emotions related to the raw texts. Analyze the customer reviews to determine whether they are positive or negative. Using ‘NLTK’ library along with Tensorflow and Keras.

We start with loading the csv file containing the data (the reviews we want to process).

After that we use some feature engineering to grab the features we are interested in:

* Review: Negative review or Positive review
* Review Type: Bad (score < 5) and Good (score >= 5)

A brief description of our dataset:

A close up of a logo

Description automatically generated

Figure 2: Dataset description

Then we perform some data exploration:

A picture containing screenshot

Description automatically generated

Figure 3: Good and bad reviews ratio visualized

We can see that we have a huge ratio between the good and bad reviews, with the good ones being many multiples of the bad ones.

Now taking a look at the most common words in the positive reviews, we obtain the following:

A picture containing newspaper

Description automatically generated

Figure 4: Most common words in positive reviews, scaled based on frequency

Doing the same thing with negative reviews we get:

A picture containing newspaper, parked

Description automatically generated

Figure 5: Most common words in negative reviews, scaled based on frequency

To deal with the imbalance in the number of good and bad reviews, we equate them to each other. We get the following distribution now:

A screenshot of a cell phone

Description automatically generated

Figure 6: Number of positive and negative reviews equated to each other for fair processing

Now we have around 80,000 samples of both review types.

We then perform One-hot encoding to transfer the two types into numerical types for our training sets and model later on.

After that a split our dataset into the following splits with 90% for training and 10% for testing:

* Reviews for training
* Reviews for testing
* Training results
* Testing results

After doing all the above, we convert the reviews we have int embedding vectors.

Now comes the next phase: Sentimental Analysis. We use this kind of classification when we have a binary kind of results (e.g. yes/no, true/false, good/bad, etc…)

For this part we use Keras and the tools it provides for building a model.

In our model we have the following:

* 1 input layer
* 2 hidden layers
* 1 dropout layer for regularization
* 1 output layer

For calculating the loss we are using categorical cross-entropy, Adam as the optimizer and we chose accuracy as our deciding metric.

We train our model for 10 epochs with a batch size of 16. We start with an accuracy score of just a little bellow 0.80 and end up with around 0.84.

A close up of a map

Description automatically generated

Figure 7: Accuracy scores across epochs

Using the results in our testing set we get a similar accuracy score of around 0.82:

model.evaluate(X\_test, y\_test)

>>> [0.39665538506298975, 0.82044786]

Now let us take the first and second reviews from our original dataset, which are labeled bad and good, respectively; Then and check if our trained model will give the same result:

y\_pred = model.predict(X\_test[:1])

print(y\_pred)

"Bad" if np.argmax(y\_pred) == 0 else "Good"

Output: [[0.9274073 0.07259267]]

'Bad'

y\_pred = model.predict(X\_test[1:2])

print(y\_pred)

"Bad" if np.argmax(y\_pred) == 0 else "Good"

Output: [[0.39992586 0.6000741]]

'Good'

We can safely say that our model can predict the “goodness” or user reviews.

## Findings and Suggestions

* The dataset was indeed massive. However, it had a mean of 8.4 and a standard deviation of 1.6. This has damaged the dataset and made the minor negative reviews act like noise.
* This can be treated by changing the percentage of the positive reviews and lowering the mean to ~5.
* We will lose lots of data if we do that. It is not ethical as it is not our data to do that and it is lying statistically.

Some areas of improvement and further areas of research include:

1. Word misspellings in feature engineering. Train the models with a misspelling indicator, as this could be a feature that might be important.
2. Look at this data across time -- there could be difference in detecting spam based on information around the time period -- for example, will adding month pick up information about holidays and the holiday vacation experience that could help detect fake reviews.
3. Would having transaction data on each review show interesting results -- for example, the time of each post, and location of the IP address that sent the post.

## Conclusion

The research above shows that this model is promising, our model does a great deal of work to allow the model to predict tricky and genuine feedback. Curiously, adding additional variables does not do much to change the models. This may change as we derive different features, such as the IP and posting time.

Only raw text can be used as the reference for making predictions. The other most significant thing is to be able to identify and extract characteristics from that raw data source. In data science projects this type of data may also come as a good complementary source to obtain more learning features and increase the predictive capacity of the models.

## Table of figures

[Figure 1: Dataset variable meanings 14](#_Toc47805461)

[Figure 2: Dataset description 15](#_Toc47805462)

[Figure 3: Good and bad reviews ratio visualized 16](#_Toc47805463)

[Figure 4: Most common words in positive reviews, scaled based on frequency 16](#_Toc47805464)

[Figure 5: Most common words in negative reviews, scaled based on frequency 17](#_Toc47805465)

[Figure 6: Number of positive and negative reviews equated to each other for fair processing 17](#_Toc47805466)

[Figure 7: Accuracy scores across epochs 19](#_Toc47805467)

## References and Bibliography

[1] Chugh, A., 2019. *Classifying Movie Reviews With Natural Language Framework*. [online] Towards Data Science. Available at: <https://towardsdatascience.com/classifying-movie-reviews-with-natural-language-framework-12dfe2fc3308> [Accessed 4 August 2020].

[2] G. M. Shahariar, S. Biswas, F. Omar, F. M. Shah and S. Binte Hassan, "Spam Review Detection Using Deep Learning," *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada, 2019, pp. 0027-0033, doi: 10.1109/IEMCON.2019.8936148

[3] Joshi, P., 2018. *A NLP Approach To Mining Online Reviews Using Topic Modeling*. [online] Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2018/10/mining-online-reviews-topic-modeling-lda/> [Accessed 5 August 2020].

[4] K. Archchitha and E. Y. A. Charles, "Opinion Spam Detection in Online Reviews Using Neural Networks," *2019 19th International Conference on Advances in ICT for Emerging Regions (ICTer)*, Colombo, Sri Lanka, 2019, pp. 1-6, doi: 10.1109/ICTer48817.2019.9023695.