Fake reviews detection techniques

Ioana-Ruxandra Stăncioi
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
ioanaruxandra.stancioi@gmail.com

Ștefan Trăușan-Matu
University Politehnica of Bucharest
313 Splaiul Independentei,
Bucharest, Romania
and
Research Institute for Artificial Intelligence
and
Academy of Romanian Scientists
stefan.trausan@upb.ro

ABSTRACT
In a world of a rapidly increasing e-commerce, fake reviews are a current problem for the customers and businesses. This thesis addresses the detection of fake reviews, solutions and improvements using machine learning techniques. Manual detection of fake reviews is a tedious process, which requires recognition of manipulative writing and examination of the questionable user's profile.

Author Keywords
Reviews; fake; ARI; sentiment; Wald-Wolfowitz.

ACM Classification Keywords
I.2.7 Natural Language Processing: Text analysis.

General Terms
Human Factors; Design; Measurement.

DOI: 10.37789/rochi.2021.1.1.26

INTRODUCTION
Nowadays, the internet is accessible to an enormous number of individuals, and its growth continues to be an ascending curve. A part of this massive group of people uses online platforms to purchase products and book services. This means that more than one in every four people is an online customer. Around seventy-five percent buyers purchase online goods at least once a month and twenty percent at least once a week [1]. At the time of writing, online shoppers represent almost 27.6 percent of the world population [2]. This means that one in five online consumers make weekly acquisitions, which is roughly one in forty people worldwide.

Online purchases reduce, on average, the time needed to make the same purchase in a real-life store. It also offers a large variety to the customers, because some products might not be available in their area. Having these privileges, the demand for online buying is increasing. It is more convenient to buy online, being just a few clicks away, than buying in a real-life store, where it implies time commuting and waiting queues. And most of them have delivery options which successfully meet urgent customer purchases. Moreover, most online shopping markets can suit any individual's lifestyle because it does not have opening and closing hours. This is the reason most markets opened online selling websites, the so-called e-shops. There are even some markets that exist only on the internet.

Because of this exodus to the online world, the approach on the buying process also changed. People can no longer rely on sensorial approaches, such as probing, smelling, visualizing and fitting the products. Online reviews are an important metric to be considered and the more trustworthy they are, the more satisfied a customer is going to be. Companies understand the impact the reviews have on customers and some of them tried to introduce fake opinions to benefit from them. The fake reviews can arise from two directions: biased customers who either are paid, receive a discount or get the product for free if they write a good review, or from a competing company who writes bad reviews to steal their customers.

Detection of fake reviews came as a necessity for both companies and users. For companies, this necessity appeared when rival companies were able to write negative reviews. The intention is to diminish the company image and make their customers turn their attention and trust to the rival company. With this in mind, competitors would write negative fake reviews, in the hope that users will read them. Therefore, companies needed a system to keep track of real reviews from real customers and to be able to remove the counterfeit ones. This is an important use case for the application, as the number of online selling markets and online shoppers is rapidly increasing.

Another important use case is represented by the users need for reliability. Indeed, on one hand, there are companies which try to devalue their opponents, but on the other hand, there are scamming enterprises which purposely cheat their own reviews, in order to gain trust and attention from customers. There are many ways in which fakes can be introduced, the company writes them, they have bots who write them, they pay for trained reviewers to write them, etc. The paid reviewers are usually instructed and have knowledge in how to leave the impression of genuineness. This makes it harder for the typical customer to identify the artificial reviews and a trained algorithm can do the job more accurately and in less time.

All this fraud is not easy to detect by a single person and it also requires time investing it. The following steps are required to check the veracity of a review. The steps focus on the reviewer's profile and their execution order is irrelevant [3]:

- Checking the wording

  This means checking if the reviewer uses mainly superlative words: e.g., the best/the worst, greatest, explosive, robust, and these words are mostly either on the positive side or the negative one. The review does not seem to be impartial.

- Checking the review content

  Real users of the product would talk about metrics: reliability, performance, durability, and an overall value. If the reviewer emphasises more on the features of the product, rather than the metrics, this can be a bad signal.

- Checking for the number of minimum or maximum graded reviews written

  Checking the grade which the reviewer usually selects also gives an inside on the trustworthiness of his reviews. If the individual generally gives mainly minimum or maximum graded reviews it is likely they are fake.

- Checking the time stamp of the reviews

  Usually, honest reviewers would write a review after they used the product for some time. If the timestamp is too close to the purchase date, that is a red flag. Also, if is a noticeable small difference between the reviews written by a specific reviewer, that also is a red flag. (MarketWatch, 2021)

Existing fake-reviews detectors on the market are Fakespot[14], ReviewMeta[15] and Thereviewindex[16]. Fakespot is a free application for product customers. It filters the fake-reviews and has features such as categorizing the reviews by positive and negative, and it compares the exact product on other selling platforms, by its prices, shipping, packaging. The software only works on Chromium-based browsers: Google Chrome and Microsoft Edge.

ReviewMeta is also an application for product customers. It is a free tool that can only be used on Amazon. It orders the reviews from the most trusted to the least and adjusts the rating after removing the unnatural reviews. It also gives a report based on the behaviour of the reviewers: how many of them have previously deleted their reviews, how many have posted all their reviews on a single day, how many used substantial repetitive phrases, or how many unverified purchases were found. ReviewMeta is developed as an application for Android and IOs, and as a browser extension for Google Chrome, Mozilla Firefox, Internet Explorer and Safari.

Thereviewindex is also a free tool built for buyers on Amazon. It examines the product reviews using different metrics and creates a summary based on the examination results. It has fewer features than the two softwares presented above. Thereviewindex is offered as a plugin for Google Chrome and Mozilla Firefox browsers.

This paper addresses the fake reviews detection problem, solutions and improvements using machine learning techniques. In the next section are described the existing products, their features and shortcomings. It is presented the state-of-the-art research and a comparison between them. It is also illustrated the theory behind the system and the evaluated options - concepts such as natural language processing, machine learning classifying algorithms and metrics used to weight the results. In the third section analyses the dataset, presents how the previous concepts come together and describes the technologies and framework used in the process. In the fourth section are evaluated the techniques used and presented a comparison between the results.

STATE OF THE ART

Natural language processing (NLP) is a subfield of artificial intelligence which enables the understanding of humans' natural language and the human-to-machine communications. Using NLP, a computer can unravel the contents of documents. It makes computers capable of reading text, hearing speech, interpret it, measure the sentiment, and determine the relevant components. It can be used for large corpora to search for patterns or discover unusual entries within the data. It is also used in the learning procedures of machine learning, to train the algorithms and obtain a more accurate prediction. NLP's tasks can be coarsely divided into four categories: processing of speech and text, analysis of morphological constructs, syntactic analysis, and lexical semantics. NLP's role is to manoeuvre the text in such a way that it offers a greater value.

Text and speech processing

Tokenization is the process in which a text is split into a set of elemental pieces, where each piece is called a token. The splitting is done using delimiters, for example white space or punctuation marks. A token can represent anything from a character to a sentence. The tokens are further used in other processes of lexical analysation, and they help with understanding the meaning of the text. This understanding can be done, for example, by analysing the frequency of keywords or the sequence of the words.

[14] https://www.fakespot.com/
[15] https://reviewmeta.com/
[16] https://thereviewindex.com/us

Morphological analysis

Analysing the whole text of a large corpora is time consuming and not every word's presence has equal impact. For example, in English, words like: "the", "a", "is", "are", "in", bring less value because they are the most common words in the language. Stop words are the words in a text which do not add any meaning to the sentence or the context. Hence, removing them does not affect the further text processing parts, but it increases the performance by reducing the feature dimension.

Lexical semantics (Sentiment Analysis)

The word sentiment is used to describe feelings, emotions, mood, and public opinions. Sentiment analysis is a natural language process which extracts the sentiments of text. This model focuses on polarity, which can be neutral, positive or negative, and has a low, medium or high intensity. Combining sentiment's polarity with its intensity, reviews can fall into five categories: very negative, negative, neutral, positive and very positive. Of course, for a much more accurate representation, the boundaries of these five groups can be changed. A mapping of polarities and rating:

- Very positive = 5 stars (more than 80% on a different scale)
- Positive = 4 stars (between 60% and 80% on a different scale)
- Neutral = 3 stars (between 40% and 60% on a different scale)
- Negative = 2 stars (between 20% and 40% on a different scale)
- Very negative = 1 stars (less than 20% on a different scale)

NLP processes were built on the model of the human brain, and sentiment analysis is no exception. The human's brain keeps track of the words it encountered and used, in its lifetime. It also learned to differentiate these words by strength. For this reason, a sentiment analysis system is built on a memory-structure, a library, which keeps track of words and their intensities. This library contains a mapping between n-grams and scores set by humans. The scores are the metric used by a computer to distinguish the intensity of the words and compute a final polarity result. For instance, negative scores would represent negative constructions, such as "bad", "terrible communication" or "angry manager", and positive scores would represent positive n-grams, such as: "excellent functionality", "wonderful view", "exquisite performance".

Automated Readability Index (ARI)

An interesting value to investigate in reviews is the Automated Readability Index. It represents the reader's ability to comprehend a text. A skilled review scammer would be interested in the understandability of his text and the audience it reaches, while a typical reviewer will not pay as much attention to it.

$$ARI = 4.71 * \frac{characters_{no}}{words_{no}} + 0.5 * \frac{words_{no}}{sentences_{no}} - 21.43$$

$Characters_{no}$ represents the total number of characters, $words_{no}$ represents the total number of words and $sentences_{no}$ represents the total number of sentences in the text. The overall computed score is designed to correspond to the typical US reading levels. Table 1 presents the US reading levels:

Score	Age	Grade level
1	5-6	Kindergarten
2	6-7	First/Second Grade
3	7-9	Third Grade
4	9-10	Fourth Grade
5	10-11	Fifth Grade
6	11-12	Sixth Grade
7	12-13	Seventh Grade
8	13-14	Eighth Grade
9	14-15	Ninth Grade
10	15-16	Tenth Grade
11	16-17	Eleventh Grade
12	17-18	Twelfth Grade
13	18-24	College student
14	24+	Professor

Tabel 1. US reading levels (Readability Formulas)

Hu, Boseb, Kohc and Liua firstly analysed computed the readability of the review using ARI [4].

Wald-Wolfowitz (Runs) test

Wald-Wolfowitz test is a statistical test, which inspects the randomness of a sequence. Also named Runs test, this test is a non-parametrical one. It is based on the null hypothesis that the data is in a random order, and it rejects the null hypothesis with a significance level not greater than α, which is 0.05 [4]. The first step is transforming the two-valued sequence into a sequence of '+' and '-' characters, using the following rule: '+' is assigned for one of the two values, and '-' for the other one; at every iteration of parsing the initial sequence, '+' or '-' is added to the new sequence correspondingly. For example:

Initial sequence = 0 1 1 0 0 0 1 1 0 1
Resulted sequence = – + + – – – + + – +

The resulted sequence contains 6 runs, 3 runs of '-' and 3 runs of '+'. The Wald-Wolfowitz test also works with discrete inputs, of various values, by separating the values into two groups: the group with values which are less than the median or the mean value, and the group with bigger values. The first group would either represent a '+' or a '-', and the second group would represent the opposite sign.

Hu, Boseb, Kohc and Liua used the ARI scores a calculate the randomness over time of the reviews of a product. A non-random result would suggest that some reviews have been manipulated. To test the randomness, they used the Wald-Wolfowitz test, which tests if a series of numbers is random [5].

IMPLEMENTATION

Dataset

The dataset is a free downloaded corpus from the Kaggle[17] platform and it consist of numerous reviews from Yelp[18]. The reviews are written in English for hotels located in North America. However, the hotels are not identified by hotel names, but are distinguished by ids which are enumerated in the Product_id column.

We observe that within the data, the user id values range from 923 to 161 147, so there is a total of 160 225 users. Additionally, there are approximately 360 000 reviews. This means that there are users who posted multiple reviews, with an equal distribution, there would be approximately 2 reviews posted by every user. Analysing the corpus, we find that the least number of reviews posted by an user is 1, and the most reviews posted by an user is 181, by the user with the id 3504. With further analysis, we discover that 66% of the reviewers only posted a single review, 27.5% of them posted a moderate number of revies, between 2 and 5, and 6.5% of them posted more than 5 reviews.

Product id values range from 0 to 922, so we can conclude that there are plenty of products which have numerous reviews. With an equal distribution, there will be almost 400 reviews for every hotel. By inspecting the data, we find that the minimum reviews number for is 11 for the hotel with the id 94, and the maximum number of reviews is 7378 for the hotel with the id 247.

ARI

We use the automated readability index for two reasons: to determine the readability index for each review and to spot differences between fake and genuine entries. With further examination we discovered that the mean of all ARI values is 6.21, a normal value which fits in the linguistic knowledge of an average student in the sixth grade. The review with the lowest ARI value, -16.22, was dash or a minus "–" and the review with the biggest value, 1806.55, is an incomprehensible review, with no whitespaces, only a sequence of 388 characters.

The reason for such a big abnormal value for an ARI is that this technique has some flaws: it cannot correctly rate a text which misses punctation marks or whitespaces. The resulted ARI values for this kind of text is very large. While a human would note that the text is incorrectly written and fit it in a lower ARI level, the formula does not cover this corner cases. In ARI levels, a score of around 14 is the biggest and denotes a user with high levels of linguistic education. The fractions $\frac{char\_nr}{word\_nr}$ and $\frac{word\_nr}{sentence\_nr}$ will have a greater value if there is only one word detected or one sentence.

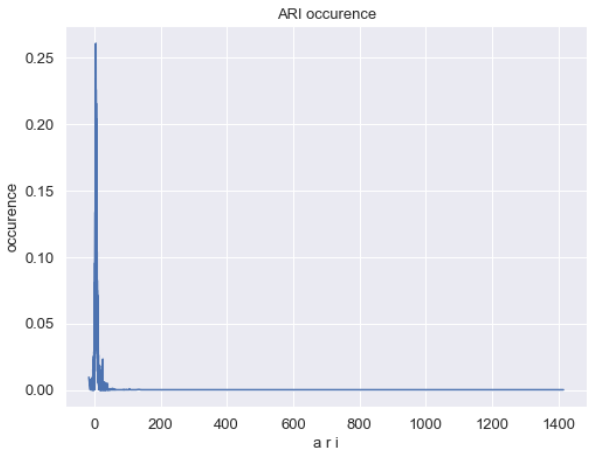

Fig 1. ARI values occurrence percentages in the big dataset

We made an attempt to solve this problem by adding corner cases, but it is tough to determine a correct correspondence between an ARI levels and those value that do not fit in these levels. A solution we came up with implies verifying the ration between the number of characters and the number of words, respectively, the ration between the number of words and the number of sentences. This was done by researching how many characters do the average English words have and how many words do the average sentences have [6].

We identified that the average length for an English word is almost five characters, and the average sentences is composed of a between fifteen to twenty words.

$$avg\_word\_len = \frac{char\_nr}{word\_nr} = 4.7 chars$$

$$15\ words \leq avg\_sentence\_len = \frac{word\_nr}{sentence\_nr} \leq 20\ words$$

When a text has no whitespaces or no punctuation marks, or both, the word_nr is proportioned to the char_nr using avg_word_len and sentence_nr is proportioned to the word_nr using avg_sentence_len and the boolean variable abnormal_ratio is set. The abnormal_ration value is initially set to 0. If only one type of writing error occurs, either missing the whitespaces or the punctuation marks, the value is set to -1. If both these typing errors occur, the value is set to -2. The ARI formula is modified to add the abnormal_ratio value.

$$ARI = 4.71 * \frac{char\_nr}{word\_nr} + 0.5 * \frac{word\_nr}{sentence\_nr} - 21.43 + abnormal\_ratio$$

[17] https://www.kaggle.com/

[18] https://www.yelp.com/

The effect of this modification is that it reduces the ARI level by one or two because of the presence of the errors. The updated maximum value is 61.62, so the error is reduced by 96.5%. The mean value is 6.05, with only 0.16 difference from the initial results.

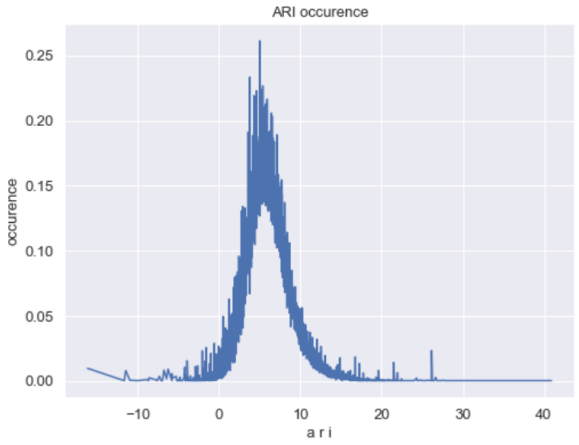

Fig 2. Modified ARI values occurrence in the big dataset

In figure 2, on the X axis there are the ARI values and on the Y axis there are the ARI values percentage occurrences. We observe in the picture above that the ARI distribution is similarly shaped to a normal distribution form. This is an indicator to the fact that the modification in the ARI formula improved the accuracy on the ARI values. In the figure 2 we present the modified ARI values, and how the abnormal_ration variable changes the distribution.

ARI differences between fake and genuine reviews

The ARI values are also used to determine the differences between the true and the fake reviews. For this comparison we choose to create a reduced version of the dataset, containing 10 000 entries and equal distribution between genuine and fake reviews. This smaller dataset keeps the original sequence between the two review types. For computing the ARI values, we used the original formula, without the abnormal_ratio variable. After computing the ARI values, the float value is rounded to two decimals, because otherwise the values are mostly unique, and the similar results are harder to spot. We then sort them and, in figure 3, we present the sorted distribution of the ARI values for the smaller corpus.

We performed a statistic test on the ARI values, which indicates the existence of fake reviews. The test we used is the Wald Wolfowitz Runs test, because it is a non-parametrical test and we work with a distribution-free dataset. In other words, the dataset is arbitrary. The test return two values: the z-stat and the p-value in a tuple format: (z-stat, p-value). Running the test on the dataset, we obtain the following results: (-99.98487738674847, 0.0). A zero value for the p-value indicates that the test rejects the null hypothesis which assumes that the data is random. This means that the ARI sequence is not random which indicates the existence of counterfeit reviews in the dataset.

For this test we used the existing method named runstest_1samp from the statsmodels.sandbox.stats.runs module [7]. The methods signature is `statsmodels.sandbox.stats.runs.runstest_1samp(`*x*, *cutoff='mean'*, *correction=True*).

Using the correspondent reviews' labels, we separated the ARI list into two lists: one containing true reviews and the other fake reviews. We computed the ARI occurrences values ipinionon both lists and plotted the results, which are shown in figure 3.

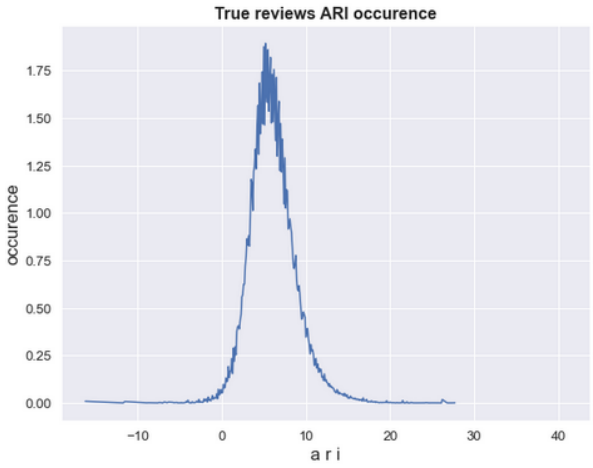

Fig 3. Aadapted ARI values occurrence in true reviews in the small dataset

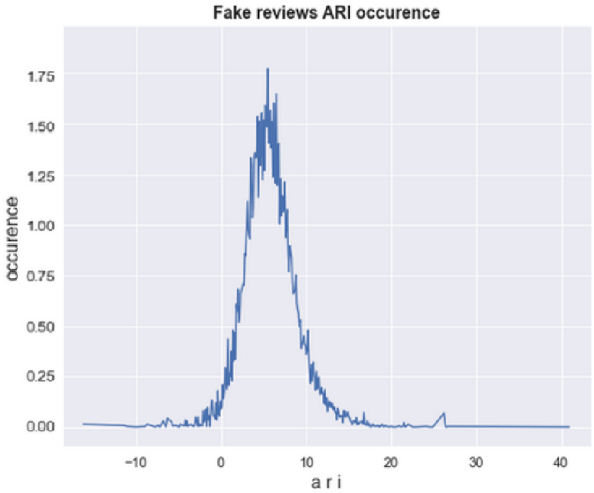

Fig 4. Adapted ARI values occurrence in fake reviews in the small dataset

VADER (Valance Aware Dictionary and sEntiment Reasoner)

Vader is a sentiment analysis tools which was purposely created for social media datasets. It is a rule-based analysis engine [8]. It is case sensitive, for example “the food was terrible” will hit a lower score than “the food was TERRIBLE”. The punctuation is also important, so “the view was astonishing!!!” will get a greater score than “the view was astonishing”. Another part of text that should not be removed in the data preprocessing part are emojis, because they are also relevant inputs for in the VADER lexicon. A downside for VADER is that it considers every word that comes after a ‘#’ or a ‘@’ irrelevant (https://predictivehacks.com/how-to-run-sentiment-analysis-in-python-using-vader/). Hashtags in social media are sometimes used to emphasis the feeling of a moment, so ignoring them might lead to inaccurate results in some reviews.

The first step is computing the polarities for each review. To do this, we firstly create a SentimentIntensityAnalyser object and then we use its polarity_score() method. The result is shown in the figure below.

	Processed_reviews	neg	neu	pos	compound	Label
0	sometimes courses. tables food selection under...	0.061	0.693	0.247	0.6486	0
1	expect little again, service. food small snott...	0.165	0.592	0.243	0.5423	0
2	snack meal. company everyone food time, nothin...	0.231	0.630	0.139	-0.2946	0
3	around little stop ambiance. bread single hear...	0.033	0.568	0.399	0.9694	0
4	snack waiters good, makes ways â taken sister ...	0.000	0.656	0.344	0.9756	0

Figure 5. Polarity scores

In figure 5, we observe that the result is a list with dictionaries. Each dictionary represents the polarity scores for a review from the dataset. The first one is {‘neg’: 0.061, ‘neu:’ 0.693, ‘pos’: 0.247, ‘compound’: 0.6486}. Each dictionary contains four keys: ‘neg’, ‘neu’, ‘pos’ and ‘compound’, where 'neg' stands for negative, 'neu' stands for neuter, 'pos' stands for positive and ‘compound’ is the score of all lexicon ratings and it is normalized to take values in the [-1;1] range. The value for each key is a float number which represents the sentiment strength of the input text. The ‘pos’, ‘neu’ and ‘neg’ probabilities add up to 1. The greatest value between them indicates the predominant polarity. For example, for the first review, we will choose the ‘neu' polarity because it has the biggest value.

The next step is using the sentiment analysis results to identify the fake reviews. One way we did this was comparing the sentiment with the rating for every review. The main idea in this comparison is that a big difference between the two features indicates a fake review. We assumed that if, for example, the review’s sentiment is negative, but its rating is highly positive, or if the review’s sentiment is positive, but its rating is negative, we mark the review as being fake. More precisely, a highly positive rated review has a 4- or 5-star rating, whereas a negative rated review receives 1 or 2 stars. We worked with two lists, sentiment_pred and sentiment_test. In sentiment_pred we added the prediction for each review after the sentiment-rating analysis, 0 for fake reviews and 1 for true reviews. In sentiment_test are the labels from the dataset, which we use to test this detecting technique.

In this part of the project, we loop through all the reviews and verify the polarities and the ratings. If the polarity is negative, but the rating is 4 or 5, we add a 0 to the sentiment_pred list to mark a fake review. Similarly, if the polarity is positive, but the rating is 1 or 2, we add a 0 to the sentiment_pred list. In any other case, we add a 1 to mark a true review.

At the end, we compare the sentiment_pred list with the sentiment_test and compute the confusion matrix which is shown is figure 6.

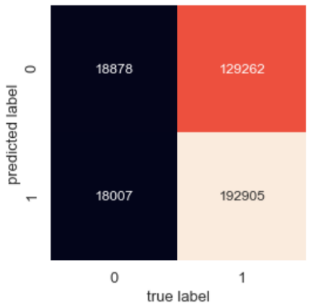

Figure 6. Sentiment-rating comparison for fake review detecting technique

The precision score is 0.127 and the recall is 0.511. The precision score indicates that the sentiment analysis works incorrectly because it misidentifies the sentiments. Recall score also tells us that half of the fake reviews were incorrectly detected as genuine.

```
Review index: 177422  Review rating: 4  Review:  Who doesn't love meat.
Review index: 177569  Review rating: 5  Review:  killer Ribs.....Priceless
Review index: 185514  Review rating: 5  Review:  Delicious but please no sweat pants!!! UGH!
Review index: 191229  Review rating: 4  Review:  A classic...but odd hours.
Review index: 201148  Review rating: 5  Review:  5
Review index: 230425  Review rating: 5  Review:  juicy dumplings.
```

Figure 7. High-rated reviews correctly detected as fake by the sentiment-rating comparison

```
Review index: 124231  Review rating: 2  Review:  good food....very pricey.....
Review index: 169698  Review rating: 2  Review:  Its okay at best. First off I've experienced better authentic cuisin
e elsewhere nearby like chang's cafe. The crab soup was good but the dumplings were tiny... oh well better luck next
time hopefully.
Review index: 177371  Review rating: 2  Review:  ok
Review index: 218616  Review rating: 1  Review:  No thanks.
Review index: 250424  Review rating: 1  Review:  I had better
Review index: 344712  Review rating: 2  Review:  love it
```

Figure 8. Low-rated reviews correctly detected as fake by the sentiment-rating comparison

In figure 7 and 8 we observe that sentiments are incorrectly detected in reviews. In the first picture, there are reviews who are positive, but are detected as negative because of some unigrams and bigrams they use, like “killer”, “doesn’t love”, “no”, “odd”. These words usually have a negative connotation, for example “killer” who has a great negative score. However, there are lexical procedures, such as metaphors and oxymorons, who use these words to emphasize and create a highly positive overall meaning.

In the same manner, because of the use of positive words, the sentiment analyser fails to detect the negative connotation of the sentence, for example in "I had better". The word "better" has a positive connotation, but used as it is in the sentence, it denotes a not so impressed feeling.

CONCLUSIONS

To conclude, this project is built to detect the online fake reviews and research different detection methods. The project's motivation resides in the customers' and markets' necessity of eliminating the fraudulent text.

The chosen dataset is a labelled collection of 360 000 reviews, which contains few fake reviews. We created a smaller dataset with only 10 000 reviews, where the fake reviews are almost the same number as the genuine ones. The new dataset preserves the initial sequence of the reviews' types, fake/genuine.

We performed the computation of the Automated Readability Index on both datasets and improved the ARI formula to adapt to corner cases which are specific to reviews. The Wald-Wolfowitz test proved the existence of fake reviews, by demonstrating that the reviews ARI sequence was not random.

We preprocessed the data using tokenization, bag of words and stop words techniques. Using the results we determined the sentiment for each review and assigned a polarity. Then, we compared the polarity with the rating to identify irregularities.

For the evaluation methods we used to compute accuracy, precision, recall and f-score. Using these four metrics we evaluated the sentiment analysis and training algorithms predictions. We compared the algorithms and created charts which show the differences between the results.

In the final analysis, the Automated Readability Index indicates the presence of fake reviews and machine learning algorithms can detect which reviews are fake with a 70% accuracy. Sentiment analysis is not accurate enough, but the results of it are biased on the dataset. On a different dataset, sentiment analysis might offer better results.

REFERENCES

1. Coppola, D. *Number of digital buyers worlwide*. Statista (2021). https://www.statista.com/statistics/251666/number-of-digital-buyers-worldwide/
2. Sabanoglu, T. *Online shopping frequency wordwide*. Statista (2020). https://www.statista.com/statistics/664770/online-shopping-frequency-worldwide/
3. Hill, C. (2018, December 10). *10 secrets to uncovering which online reviews are fake*. MarketWatch. https://www.marketwatch.com/story/10-secrets-to-uncovering-which-online-reviews-are-fake-2018-09-21
4. Minitab, 2019. *Interpret the key results for Runs Test.* https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/basic-statistics/inference/how-to/one-sample/runs-test/interpret-the-results/key-results/
5. Hu Nan, Bose Indranil, Koh Noi Sian, Liu Ling. ''Manipulation of online reviews: an analysis of ratings, readability, and sentiments''. Decis Support Syst 2012; 52:674–84.
6. Grzybek P. *History and Methodology of Word Length Studies. In: Grzybek P. (eds) Contributions to the Science of Text and Language. Text, Speech and Language Technology, vol 31*. Springer (2007), Dordrecht. https://doi.org/10.1007/978-1-4020-4068-9_2
7. Perktold J., Seabold S., Taylor J., (2009-2019). *Source code for statsmodels.sandbox.stats.runs.* https://www.statsmodels.org/stable/_modules/statsmodels/sandbox/stats/runs.html#runstest_1samp
8. Hutto, C.J. & Gilbert, Eric. (2015). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014.