UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

CARLOS MORVAN FILHO DE PAULA E SANTIAGO

Rating Prediction of Product Reviews: An approach based on the BERT language model

Work presented in partial fulfillment of the requirements for the degree of Bachelor in Computer Science

Advisor: Prof. Dr. Joel Carbonera
Co-advisor: Dr. Luan Garcia

Porto Alegre
April 2023

“A must-read, a work of art!”

— REVIEW 9

ABSTRACT

Nowadays online shopping has become increasingly common. Without the possibility to try on the products, customers became dependent on other customers' reviews. Those are usually written in natural language and can be displayed along with a rating scale. These ratings can then be easily used by a computer to filter those reviews and bring useful information for customers and sellers. However, in general, ratings are an optional field in reviews and, due to this, it is common for users to provide only textual reviews without ratings. In these cases, processing the sentiment of these reviews is not trivial for computers. In this context, recent advances in the field of Machine Learning are allowing us to develop approaches that present a promising performance for inferring the ratings from textual reviews. Recent works have investigated the adoption of fine-tuned pre-trained language models for sentiment analysis and rating prediction. The goal of this work is to investigate the performance of an approach based on BERT pre-trained language model for predicting ratings from textual book reviews. We performed four experiments, considering different variations of the original problem. Each variation involved considering different aggregations of the original set of classes of the dataset. As a general conclusion of our experiments, our BERT-based approach achieved a good performance in some of the considered experimental settings. The best performances were achieved in the experimental setting considering only 2 classes, grouping neutral and negative classes in one class (negative) and the positives as the other.

Keywords: BERT. Machine learning. Neural Networks. sentiment analysis. supervised learning. text classification.

RESUMO

Hoje em dia, compras online estão se tornando cada vez mais comuns. Sem a possibilidade de experimentar os produtos, os clientes se tornaram dependentes das avaliações de outros clientes. Essas avaliações geralmente são escritas em linguagem natural e podem ser exibidas juntamente com uma escala de classificação. Essas classificações podem ser facilmente usadas por um computador para filtrar essas avaliações e fornecer informações úteis para clientes e vendedores. No entanto, em geral, as classificações são um campo opcional nas avaliações e, devido a isso, é comum os usuários fornecerem apenas avaliações textuais sem classificações. Nesses casos, processar o sentimento dessas opiniões não é trivial para computadores. Nesse contexto, avanços recentes no campo do Aprendizado de Máquina estão permitindo o desenvolvimento de abordagens que apresentam um desempenho promissor para inferir as classificações de avaliações textuais. Trabalhos recentes investigaram a adoção de modelos de linguagem pré-treinados e posteriormente especificamente ajustados para análise de sentimento e previsão de classificação. O objetivo deste trabalho é investigar o desempenho do abordagens baseadas no modelo de linguagem pré-treinado BERT na previsão de classificações de avaliações textuais sobre livros da Amazon. Para alcançá-lo, realizamos quatro experimentos considerando quatro diferentes variações do problema original. Cada variação envolveu considerar diferentes agregações das classes originais do problema. Como conclusão geral, a abordagem baseada em BERT atingiu uma boa performance em alguns dos cenários considerads. O melhor resultado foi alcançado no experimento que considera apenas 2 classes, agrupando as classes neutras e negativas como uma e as positivas como outra.

Palavras-chave: Aprendizado de máquina. Analize de sentimento. Aprendizado supervisionado. BERT. Classificação de texto. Redes neurais.

LIST OF FIGURES

Figure 2.1 Supervised learning diagram 13
Figure 2.2 Pre-training and fine-tuning procedures for BERT in a question-answering task. 14
Figure 2.3 Confusion Matrix 15

Figure 4.1 Dataset sample 23
Figure 4.2 Dataset class distribuition 23
Figure 4.3 Dataset Review length (in terms of word count) distribution in logarithmic scale 24
Figure 4.4 Classifier architecture 27
Figure 4.5 Learning curve of Experiment 3 29
Figure 4.6 Confusion matrix for experiment 1 31
Figure 4.7 Confusion matrix for experiment 2 31
Figure 4.8 Confusion matrix for experiment 3 32
Figure 4.9 Confusion matrix for experiment 4 32

LIST OF TABLES

Table 3.1 Related works summary 21

Table 4.1 Pre-processing steps 25
Table 4.2 Class mapping 26
Table 4.3 Instances Distribution 26
Table 4.4 Result of the experiments 30

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial intelligence
ALBERT	A Lite BERT
BERT	Bidirectional Encoder Representations from Transformers
Bi-LSTM	Bi-directional Long Short Term Memory
CNN	Convolutional Neural Network
k-NN	K-nearest neighbor
LR	Linear Regression
Ml	Machine Learning
NB	Naïve Bayes
NLP	Natural Language Processing
RNN	Recurrent Neural Network
RoBERTa	A Robustly Optimized BERT Pretraining Approach
SGD	Stochastic Gradient Descent
SVC	Support Vector Machine Classifier
SVM	Support Vector Machine

CONTENTS

1 INTRODUCTION 10
2 BACKGROUND 12
2.1 Artificial Intelligence 12
2.2 Machine Learning 12
2.3 BERT 13
2.4 Model Evaluation 15
2.4.1 Metrics 15
2.4.1.1 Accuracy 16
2.4.1.2 Precision 16
2.4.1.3 Recall 16
2.4.1.4 F1-measure 16
2.4.2 Cross-Validation 17
3 RELATED WORK 18
4 EXPERIMENTS 22
4.1 Methodology 22
4.1.1 Dataset 22
4.1.2 Dataset undersampling 24
4.1.3 Dataset preprocessing 25
4.1.4 Experimental scenarios 25
4.1.5 Experimental settings 27
4.1.6 Environment Configuration 29
4.2 Results 29
5 CONCLUSION 33
REFERENCES 34
APPENDIX A — EXAMPLE OF LITERARY REVIEW 36

1 INTRODUCTION

In recent years, the volume of online shopping has grown in a fast way. Without the possibility to try on the products, customers became dependent on other customers' reviews. Those are usually written in natural language and can be paired or not with a rating system, to represent the sentiment of that rating. The ratings that represent the review sentiment, in general, can be easily processed by computers for allowing different tasks, such as retrieving reviews with some specific class, for example.

However, analyzing such a large amount of unstructured data poses a significant computational challenge for conventional approaches. In recent years, Machine learning approaches have been showing great performance in dealing with these challenges in tasks of text classification and sentiment analysis. For example, works such as Neethu and Rajasree (2013) achieve impressive performance for predicting the sentiment of tweets using traditional machine learning approaches.

More recently, several works have been exploring pre-trained language models, such as BERT (DEVLIN et al., 2018), for dealing with these tasks. BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained deep learning model that has achieved great results on a wide range of NLP tasks. Its ability to capture the context and semantics of language has made it one of the most popular language models used in different downstream tasks.

The main advantage of adopting models such as BERT is the possibility of fine-tuning them for different problems. This allows us to leverage the knowledge learned by them in large amounts of texts (during pre-training) and use this knowledge as a starting point for learning important features in other contexts.

The general goal of this work is to investigate the performance of BERT-based approaches in classifying Amazon book reviews. The selected dataset comprehends more than 27 million reviews originally classified into 5 ratings (1-5 star scale). In this work, we aim to evaluate the performance of BERT in 3 different settings of rating prediction: considering the original classification in five classes (1-5 star scale), considering three classes (positive, neutral, and negative), and two 2 classes (negative and positive), using different aggregations of the original classes.

From the experiments performed, the worst setup was considering the five classes, which is the original problem. It achieved 59% of macro F1-measure, indicating that it is hard to classify text samples in the original five classes. The best scenario was considering

only two classes, aggregating all negative and neutral reviews together. It achieved a Macro-F1 score of 88%. Our alternative scenario, considering also only two classes, but aggregating the neutral and positive reviews achieved 82% of macro F1-measure. Those results indicate that neutral reviews can present a bias towards negative sentiments.

The remainder of this work is structured as follows. Chapter 2 presents the fundamental knowledge required to understand this work, ranging from broad content such as AI and ML to more specifics such as the metrics and dataset used in this work. Chapter 3 presents relevant works in sentiment analysis and review prediction. Chapter 4 details the experiments done, describing the setup, configuration and also discussing the results. Chapter 5 concludes this work, summarizing what was done and presenting possibilities for future works.

2 BACKGROUND

In this chapter, we present the main concepts that support this work. We start by describing the fields of AI in section 2.1 and Machine learning in section 2.2. In section 2.3 we present BERT models architecture. Finally in section 2.4 we describe the metrics and techniques used to evaluate the performance of the experiments.

2.1 Artificial Intelligence

Artificial Intelligence(AI) is a field of Computer Science that aims to develop machines capable of performing tasks that would require human intelligence.

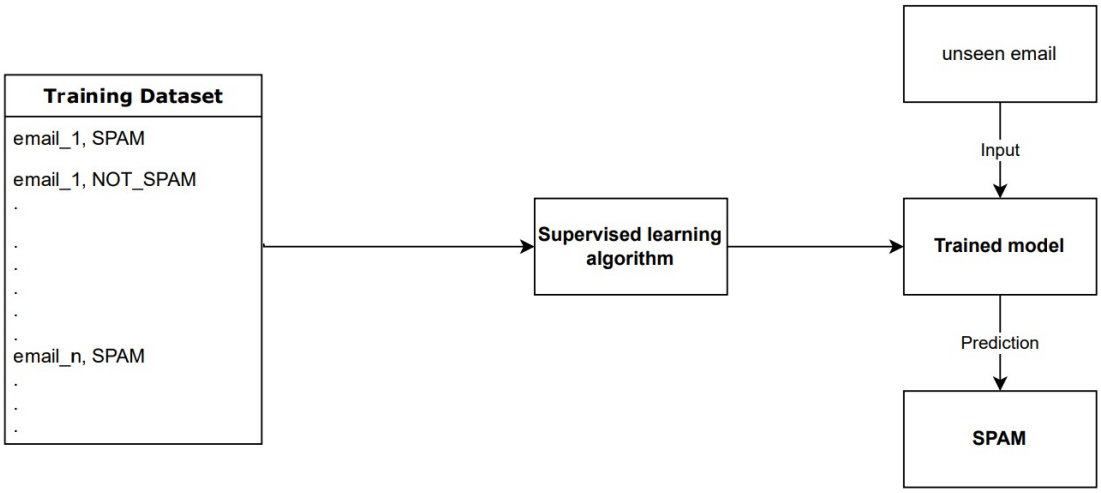
One of the most famous definitions of AI was made in POOLE, MACKWORTH and GOEBEL (1998), they described it as *the study of "intelligent agents": systems that can perceive their environment and take actions that maximize their chances of achieving their goals*. In this context, agents can be any systems that can act upon receiving and reasoning about a given interaction with its environment.

The goal of developing intelligent systems created branches of AI, each of them adopting different perspectives regarding the notion of intelligence or being specialized in solving different kinds of problems. This work adopts concepts and techniques of a specific branch of AI called Machine Learning.

2.2 Machine Learning

Machine Learning(ML) is a subfield of AI that studies the ability to improve performance based on experience (RUSSELL; NORVIG, 2009). It is a field that is rapidly growing in recent years. In this field, there are three main kinds of learning approaches: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning focus on training a machine learning model using labeled data, later using it to make predictions on new, unseen data. Unsupervised learning trains the model using unlabeled data, allowing it to identify patterns within that data. Finally, Reinforcement learning involves learning from the feedback received from its environment. This work is focused on supervised learning tasks. In figure 2.1 we present a scenario that describes the training of an email spam filter model using supervised learning. A supervised learning

Figure 2.1 – Supervised learning diagram



Source: The Author

algorithm is trained on a labeled dataset, which contains e-mails labeled as *spam* or not spam. As the output of this learning process, the supervised learning algorithm generates a model that can be used for classifying unseen emails.

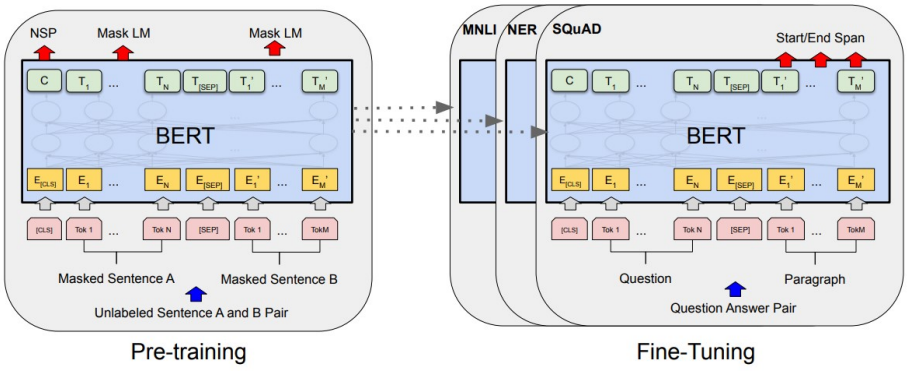
One of the main advantages of supervised ML, and the main reason it is being used in a wide range of industries, is its capability to analyze large datasets and build predictive models from this data. The resulting models can be used for supporting a wide number of complex tasks, allowing users to make decisions based on insights and patterns represented by the models.

2.3 BERT

BERT (Bidirectional Encoder Representations from Transformers) is an open-source pre-trained model developed by Google and released in 2018. At the time of its publication, the model obtained state-of-art results on at least 11 NLP Tasks (DEVLIN et al., 2018). BERTs architecture uses a multi-layer bidirectional Transformer, proposed by Vaswani et al. (2017).

Transformer is a kind o Neural Network (NN) architecture that uses a self-attention mechanism to allow the NN to understand a word using the context defined by the words around it. Before it was proposed, the traditional model for language processing was the Recurring Neural Network (RNN). This model is designed for processing sequential data,

Figure 2.2 – Pre-training and fine-tuning procedures for BERT in a question-answering task.



Source: Devlin et al. (2018)

but when dealing with long sequences it would lose the context. Besides that, it is very limited in the sense of how parallelizable it could be. In that sense, Transformers were developed as a better and more scalable solution. Instead of processing each word based on the output of the word before, it uses the mentioned self-attention mechanism to weigh the relationship between each word to every other word of a sequence, then it can process them in parallel since their relationship is already established.

The main innovation of BERT is its bidirectional processing. That is, it can consider both the left and right context of a word in a sentence. This makes BERT capable of capturing complex relationships between words and a better understanding of the context. BERT is pre-trained in two different tasks. Firstly, it is trained by masking some percentage of the input tokens at random and then predicting those masked tokens. After that, BERT is trained in a task of *next sentence prediction* that allows the model to learn the relationship between sentences. This process is carried out considering a large corpus of training data, which includes the English Wikipedia and the BookCorpus dataset.

After the pre-training BERT is ready to be fine-tuned for some specific downstream task. In this stage, a novel training process is carried out in a new task (with a suitable novel dataset for this task), adopting the weights resulting from the pre-training phase of BERT as a starting point. In this process, BERT can be used as a module within a more sophisticated architecture, with an additional output layer that fits the problem. That method is called transfer learning (GOODFELLOW; BENGIO; COURVILLE, 2016), where we use a pre-trained model as the starting point for a new model on a different task. Figure 2.2 presents an example of transfer learning where BERT is fine-tuned in a task of question-answering.

2.4 Model Evaluation

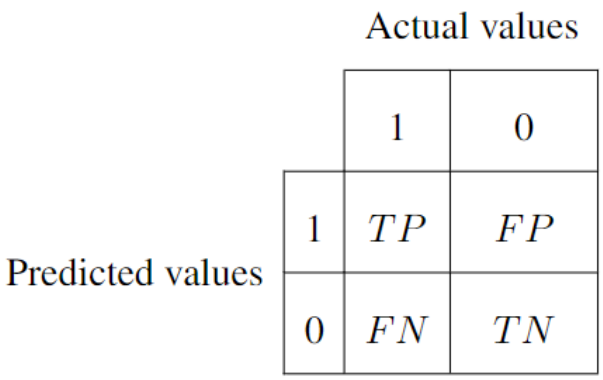
In this section, we present the different metrics and techniques used to evaluate the performance of our model in our goal task.

2.4.1 Metrics

For this work, we chose to use four metrics to evaluate the models, accuracy, Macro precision, Macro recall, and macro-f1. They are all derived from the True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values, which can be identified in a confusion matrix, as illustrated in Figure 2.3. True Positive is an instance of positive prediction that is actually positive. True negative is an instance of negative prediction that is actually negative. A false positive is an instance of a positive prediction that is actually negative. And False negative is an instance of negative prediction that is actually positive.

Precision, recall and f1-measure are originally defined for binaray classification settings. However, in this work we are dealing with some multiclass settings. In these cases, we are considering macro averages of precision, recall and f1-measure, which are, respectively, the arithmetic average considering the precision, recall, and f-measure of each class.

Figure 2.3 – Confusion Matrix



Source: the Author

2.4.1.1 Accuracy

Accuracy is the number of correct predictions divided by the total of predictions, as seen in equation 2.1.

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2.4.1.2 Precision

Precision is the number of positive predictions that were really positive, as seen in equation 2.2.

$$\textit{Precision} = \frac{TP}{TP + FP} \tag{2.2}$$

And its macro average is define in equation 2.3.

$$\textit{Macro-Precision} = \frac{\sum_{i=1}^{n} Precision_i}{n} \tag{2.3}$$

where *i* is a given class and *n* is the total number of classes

2.4.1.3 Recall

Recall refers to how many actual positives were predicted correctly, as seen in equation 2.4.

$$\textit{Recall} = \frac{TP}{TP + FN} \tag{2.4}$$

And its macro average is defined in equation 2.5.

$$\textit{Macro-Recall} = \frac{\sum_{i=1}^{n} Recall_i}{n} \tag{2.5}$$

where *i* is a given class and *n* is the total number of classes

2.4.1.4 F1-measure

F1-measure is the Harmonic mean of precision and recall for a more balanced summarization of model performance, it is calculated using equation 2.6.

$$\textit{F1} = 2 * \frac{\textit{Precision * Recall}}{\textit{Precision + Recall}} \tag{2.6}$$

And its macro average is defined in equation 2.7.

$$\textit{Macro-F1} = \frac{\sum_{i=1}^{n} F1_i}{n} \tag{2.7}$$

where *i* is a given class and *n* is the total number of classes

2.4.2 Cross-Validation

Cross-validation is a process that can be used to estimate the quality of a classifier model. This procedure is based on the idea of repeating the training and testing computation on different chosen subsets or splits of the original dataset (GOODFELLOW; BENGIO; COURVILLE, 2016).

The most common technique is k-fold cross-validation. In this approach, the original dataset is divided into k equal-sized subsets (folds), and the model is then trained and evaluated k times. In each iteration, one of the folds is used as test data, and the other $k - 1$ are used as the training dataset. The performance of the model is then evaluated based on the average of the performance achieved in each test fold.

3 RELATED WORK

In this section, we discuss related works that were relevant to the development of this work.

In Balakrishnan et al. (2022) the authors present a comparison of deep learning models applied for rating prediction to a women's clothing dataset. In this paper, two experiments were carried out. The first compared the performance of Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Bi-directional Long Short Term Memory (Bi-LSTM), in two different setups, 3-class (negative, neutral, and positive), and 5-class (extremely negative, negative, neutral, positive, and extremely positive). The second compared BERT, RoBERTa(LIU et al., 2019), and ALBERT(LAN et al., 2020), also with 3-class, and 5-class setups. Both experiments were performed considering the original dataset and also an augmented version of it. The highest F-score for their models was RNN for the 3-class setup with 89.77% while the best Bert variation was RoBERTa for the 3-class setup with 73.09% F-score. Their work served as the primary inspiration for our experiments, in which we compare BERT in different configurations.

In Taparia and Bagla (2020) the authors compared the performance of three approaches: Multinomial Naïve Bayes, Logistic Regression Classifier, and Linear SVC. The dataset used was from Amazon cellphone and Accessories review, the same source as the one used in this work, the difference is that they use both review summary and review text to train the model. The best result in their experiments was achieved by Logistic regression, with a Macro-f1 of 54.1%.

In WU et al. (2020), the authors proposed a new model called SenBERT-CNN that combines the BERT model with CNN structure, aiming to surpass baseline models in the sentiment analysis task. The model first uses BERT to perform word vector coding to represent the semantic information, then uses the CNN structure to further extract the text features in depth. The author's model was trained with a dataset of 9600 smartphones reviews from JD.com. The reviews were classified into two classes: positive and negative reviews. Considering the accuracy, the resulting model outperformed both CNN (85.03% acc) and BERT (92.47% acc) individually, obtaining an accuracy of 95.72%, showing that the hybrid model had a better performance than the baseline ones.

In Pota et al. (2021), the authors propose a different approach for Twitter sentiment analysis applied to tweets in Italian, involving a two steps approach to sentiment analysis using BERT. The first step is to translate tweet jargon such as emojis, hashtags, and ASCII

emoticons to plain text, making it easier for the model to get the context of the tweets. The second step is to pre-train BERT using plain text, instead of just tweets like previous state-of-art approaches to this problem. The results show that both steps helped to increase the model performance. Compared to the best system, ALBERTo (POLIGNANO et al., 2019), the proposed model was 3% better on average. But given that the proposed one was trained in a multilingual text corpus, there is a huge gain in not having to train it again for every other language of interest. Besides that, this work shows many cases where transforming Hashtags and emojis into plain text turned reviews that got a wrong prediction into right predictions.

In Bilal and Almazroi (2022) the authors compare 3 bag-of-words based classifiers, K-nearest neighbor (k-NN), Naïve Bayes (NB), and Support Vector Machine (SVM) with BERT, considering different sequence lengths. Those models were trained with Yelp reviews separated into two classes, Helpful and Unhelpful. This work has given some good insights into how to analyze the dataset for better fine-tuning with BERT. They concluded that usually, helpful reviews will have around 190 words, and in their experiment, the best BERT performance was with reviews with a sequence length of around 320, which got an F-score of 71.7%, 4% higher than the best baseline model, SVM.

In Haque, Saber and Shah (2018), the authors compare the performance of several different approaches applied to three Amazon product datasets. In their work, they compared algorithms like Naïve Bayes, Support vector Machine (SVM), Stochastic Gradient Descent (SGD), Linear Regression (LR), Random Forest, and Decision Tree. All of them trained with a mobile phone, an electronic, and a musical instruments Review dataset from Amazon. Different from our work, on their dataset, the instances were not labeled, so the author used Active learning, a semi-supervised technique in which the model is first trained with a fully trained part of the dataset and then alternates between trying to evaluate the dataset, selecting instances that by being labeled would add the most value to the model, querying an expert to manually label those instances and then adding them back to the model for the next iteration. Another interesting choice for this work was that, since the goal was for the models to classify the reviews as positive and negative, the author grouped 1-star and 2-star ratings as negative, 4-star and 5-star as positive, and completely discarded all neutral(3-star) reviews. For the three datasets, SVM was the best-performing model with an average 97% F1-Score.

In Karthika and Palanisamy (2016), the authors compared the performance of a Naïve Bayes model using different feature extraction methods in a sentiment analysis

task to classify Amazon book, Music, and Camera reviews. The Book Dataset used in this work is very similar to ours in the structure of reviews. The main difference is in their instances, which are classified as positive and negative instead of using the 1-5 star scale. In their work, they trained the Naïve bayes model using phrase-level feature extraction, single-word, and multi-word methods. The best highest F-score obtained by their model varied between the datasets, for the book dataset the best was the multi-word method, with an F-score of 75%. For camera and music, the best method was single-word, with an F-score of 80.3% for both datasets.

A summary of all the related works can be seen in table 3.1

Table 3.1 – Related works summary

Work reference	*Year*	*Dataset*	*Best Model*	*of Classes*	*F-Score*
(BALAKRISHNAN et al., 2022)	2022	Women's E-commerce clothing reviews	RNN	3	89.77
			RoBERTa	3	73.09
(TAPARIA; BAGLA, 2020)	2020	Amazon cellphone an accessories	Logistic regression	5	54.1
(WU et al., 2020)	2020	jd.com Customer reviews	SenBERT-CNN	2	95.32
(POTA et al., 2021)	2021	Tweets in Italian	ALBERTo	2	75
(BILAL; ALMAZROI, 2022)	2022	Yelp shopping reviews	BERT(312 sequence length)	2	71.7
(HAQUE; SABER; SHAH, 2018)	2018	Amazon Cellphones & accessories reviews	SVC	2	97
		Amazon Musical instruments reviews	SVC	2	98
		Amazon Eletronics Reviews	SVC	2	98
(KARTHIKA; PALANISAMY, 2016)	2016	Amazon Book reviews	Naïve bayes(multi-word feature extraction)	2	75
		Amazon Music reviews	Naïve bayes(single-word feature extraction)	2	80.3
		Amazon Camera Reviews	Naïve bayes(single-word feature extraction)	2	80.3

Source: The Author

4 EXPERIMENTS

This chapter presents the main contributions of this work. Section 4.1 presents the methodology adopted in our experiments, while Section 4.2 presents and discusses the results obtained in our experiments.

4.1 Methodology

In this section, we describe how our experiments were designed. It consists of first presenting the dataset used. Then we describe the undersampling process performed on the Amazon review dataset and the class grouping for experiments that need it, as described in section 4.1.2. That undersampled dataset then passes through a standard preprocessing step, described in section 4.1.3. We then present the Experiments scenarios in section 4.1.4 and the settings for the models in section 4.1.5. Finally, we describe the environment configuration, in section 4.1.6.

4.1.1 Dataset

In this work, we adopted the Amazon review dataset provided by Ni (2018). This dataset contains reviews of Books purchased from Amazon.com. It includes 27,164,983 reviews, with 11 features each. Each review includes the following features:

- reviewerID - ID of the reviewer
- asin - ID of the product
- reviewerName - the name of the reviewer
- vote - helpful votes of the review
- style - a dictionary of the product metadata
- reviewText - text of the review
- overall - rating of the product on a scale of five stars
- summary - summary of the review
- unixReviewTime - time of the review (Unix time)
- reviewTime - time of the review (raw)
- image - images that users post after they have received the product

An instance of this dataset can be seen in figure 4.1.

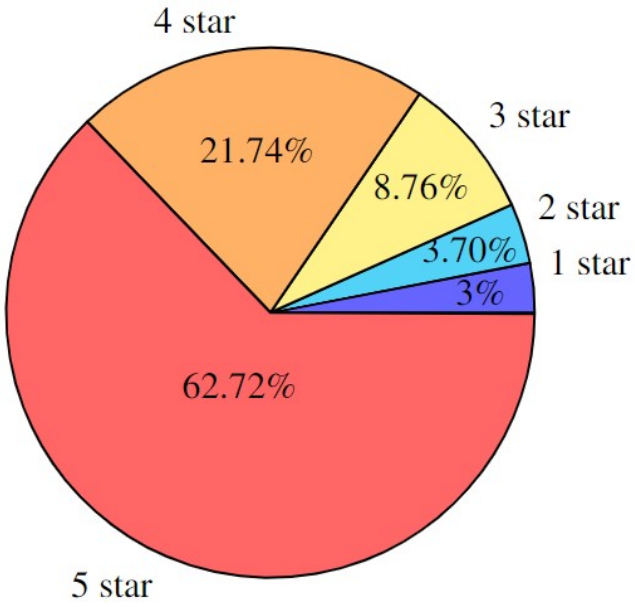
Figure 4.1 – Dataset sample


```
{
"overall": 5.0,
"verified": true,
"reviewTime": "11 9, 2012",
"reviewerID": "A2M1CU2IRZG0K9",
"asin": "1713353",
"style": {"Format:": " Paperback"},
"reviewerName": "Terri",
"reviewText": "My students (3 & 4 year olds) loved this book!  Definitely recommend it to other teachers.",
"summary": "Amazing!",
"unixReviewTime": 1352419200
}
```


Source: the Author

The instances in this dataset are classified into 5 classes corresponding to a rating scale from 1-5 stars. Classes 1 and 2 contain reviews that classify the books as very bad and bad, class 3 contains neutral reviews, and classes 4 and 5 contain reviews that classify the books as good and very good. The distribution of those instances can be seen in figure 4.2.

Figure 4.2 – Dataset class distribuition



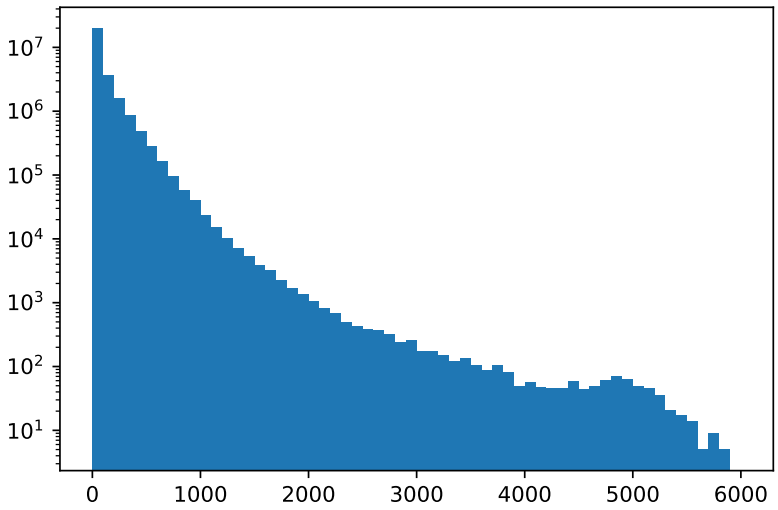
Source: the Author

And finally, after analysing the dataset, we identified two main types of reviews. The first is product review, which classifies the books as a product, and evaluates them according to their quality, material, and layout, having a similar vocabulary among them-

selves. The second type is literary review, which classifies books according to their plot. Those can have a specific vocabulary for each book, and classifying them can be challenging. For shorter reviews, the two types are mixed together, for longer ones, it's more common to see the literary ones. One example of a literary review can be seen in appendix chapter A

Figure 4.3 presents a histogram that represents the distribution of reviews according to their word count, where each bin represents a range of 50 words. According to this analysis, we can observe that short reviews dominate the dataset.

Figure 4.3 – Dataset Review length (in terms of word count) distribution in logarithmic scale.



Source: the Author

4.1.2 Dataset undersampling

Due to the time and resource constraints that provide the frame of this work, it was not possible to consider the complete dataset in our experiments. Thus, as a first step, we performed a downsampling of the original dataset in order to build a manageable subset of data that was used in our experiments. The main goal was to reduce the dataset for making the training process of or classifier faster and enable us to run it with different configurations.

Based of the conclusions of Bilal and Almazroi (2022), we tried to keep the upper limit of words per review higher than 320 words, but ensuring a balance of reviews per class. Besides that, we wanted to focus more on the product reviews rather than the

literary reviews, which usually are the bigger ones. In order to meet these criteria, we first discarded all reviews with more than 400 words, representing 5% of our dataset. After that, from the remaining dataset, we randomly selected 20,000 reviews of each class for building the dataset used in our experiments. From the attributes of the original dataset, we selected only the overall score (the class that we want to predict) and the review text.

4.1.3 Dataset preprocessing

The undersampling was followed by the preprocessing step, whose main goal was to normalize the textual content of the reviews and remove some elements from the text. All steps can be seen in table 4.1. Diacritics were removed using *gensim library*[1] and the reviews were lemmatized using *nltk library*.

Table 4.1 – Pre-processing steps

Pre-processsing steps	*Examples*
Original Text	These books are AMAZING!!!
Convert the review to lowercase	these books are amazing!!!
Remove leading and trailing spaces	these books are amazing!!!
Remove punctuations	these books are amazing
Lemmatization	these book are amazing

Source: The Author

At this point, it is important to notice that in our preliminary tests, removing stopwords had a negative impact on the performance of our BERT-based classifier. This result is aligned with the conclusions of Saif et al. (2014) that suggest that some methods of removing stopwords can harm the performance of classifiers in sentiment analysis. Due to this, we decided to keep the stopwords.

4.1.4 Experimental scenarios

In this work, we performed four experiments, in each of them we trained a BERT model with different ways of aggregating the original five classes.

In Experiment 1, we considered the dataset in its original classification. That is, in this experiment, we considered the five original classes, where each class represents the sentiment of the reviews. Class 1 represents the very bad reviews, class 2 the bad

[1] <https://github.com/RaRe-Technologies/gensim>

reviews, class 3 the neutral reviews, class 4 the good reviews, and class 5 the very good reviews. For each class, we used 20,000 instances, with a total of 100,000 instances for the experiment.

In Experiment 2, we transformed the dataset into a 3-class problem, aggregating classes 1 and 2 as negative reviews, leaving class 3 as neutral reviews, and aggregating classes 4 and 5 as positive reviews. For each resulting class, we used 20,000 instances. Thus, from our undersampled dataset, we randomly sampled 10,000 reviews of classes 1, 2, 4, and 5, and 20,000 for class 3. This experiment considered a total of 60,000 reviews.

For Experiments 3 and 4, we aggregated the neutral reviews with either the negative or the positive class from experiment 2, transforming the problem into a binary problem, with reviews being either negative or positive. For both experiments, we used 20,000 instances for both classes, totalizing 40,000 instances.

In Experiment 3, for the negative portion, we used 6,666 instances from original classes 1,2, and 3. For the positive portion, we used 10,000 instances from original classes 4 and 5.

In Experiment 4, for the negative portion, we used 10,000 instances from original classes 1, and 2. For the positive portion, we used 6,666 instances from original classes 3, 4, and 5.

The aggregation and class distribution of each experiment are summarized in tables 4.3.

Table 4.2 – Class mapping

Original rating	*Experiment 1*	*Experiment 2*	*Experiment 3*	*Experiment 4*
1 star	1	1	1	1
2 star	2	1	1	1
3 star	3	2	1	2
4 star	4	3	2	2
5 star	5	3	2	2

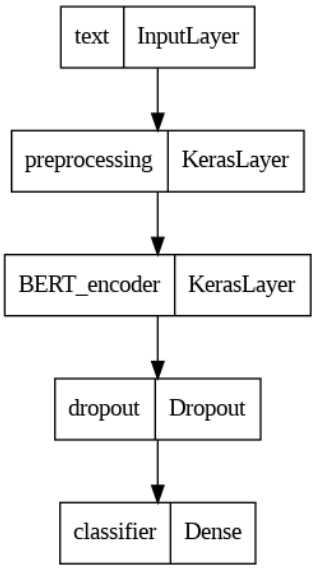
Source: The Author

Table 4.3 – Instances Distribution

Original rating	*Experiment 1*	*Experiment 2*	*Experiment 3*	*Experiment 4*
1 star	20000	10000	6667	10000
2 star	20000	10000	6667	10000
3 star	20000	20000	6666	6666
4 star	20000	10000	10000	6667
5 star	20000	10000	10000	6667

Source: The Author

Figure 4.4 – Classifier architecture



4.1.5 Experimental settings

In all experiments, we adopted the same basic BERT-based neural network architecture as our classifier. Figure 4.4 represents the architecture of the classifier adopted in our experiments. The first layer is the input layer, it accepts the textual data and passes it to the next layer. The next one is the pre-processing layer, which transforms the input into numeric token ids and arranges it in several Tensors that are used as input by BERT. Then we have the BERT_encoder layer that includes the pre-trained BERT model and outputs the embeddings for the entire input review. After that, we have the dropout layer, which randomly selects neurons to be ignored, helping to prevent overfitting in the model. Finally, we have a dense layer with the activation function, which produces the prediction of the model. Notice that the last layer can have different number of outputs, depending on the number of classes considered in the experiment.

In all experiments, we evaluate our BERT-based classifier in a cross-validation process consisting of 5-fold cross-validation. In each iteration, our dataset had a distribution of 80% of the instances in the training set and 20% on the test set. The validation set was derived from the training set, using 10% of its instances.

Since we are dealing with a multiclass classification task with a balanced dataset, we evaluated our model in all experiments according to the following metrics (discussed in Section2.4.1): Accuracy, Macro-f1, Macro recall, and Macro precision.

To run the four experiments, we used the *Tensorflow library*[2] using the AdamW optimizer and a similar configuration of hyperparameters for all experiments. The selected configuration was:

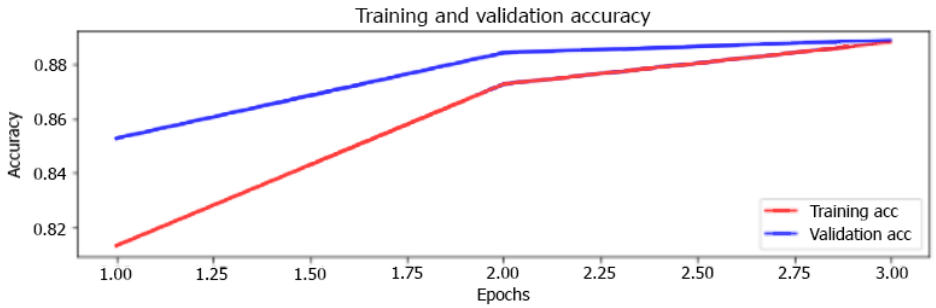
- Model: The BERT model to fine-tune. We chose bert_en_uncased[3]
- Batch size: The number of training examples used in each iteration of the training process. For all models, the batch size was 32, due to environment limitations and following (DEVLIN et al., 2018) recommendation.
- Learning rate: The step size used to update the model weights during training. For all models, the learning rate used was 3e-5. This was chosen among the recommended learning rates from (DEVLIN et al., 2018) and was empirically tested as the best for our work.
- Number of epochs: The number of times the entire training dataset is passed through the model during training. All models were trained with 3 epochs, following (DEVLIN et al., 2018) recommendation. Besides that, this value also was empirically justified, since by observing the learning curves of the models we can notice that there is no improvement in the validation accuracy after 2 or 3 epochs. This can be seen in image 4.5.
- Dropout rate: The probability of randomly dropping out a neuron during training. All models used 0.1 for dropout rate, following (DEVLIN et al., 2018) recommendation.
- Warmup steps: The number of initial training steps during which the learning rate is gradually increased. All models used 10% as their warmup percentage. In our experiments, changing this value did not impact significantly the results.
- Steps per epoch: The number of batches for each epoch. This value was defined as

$$\texttt{Steps per epoch} = \frac{\texttt{Number of samples}}{\texttt{Batch size}} \tag{4.1}$$

- Loss function: The loss function computes the distance between the expected output and the actual output. For Experiments 1 and 2, the categorical cross-entropy function was used, and for Experiments 3 and 4, Binary Cross-entropy was used.

[2]<https://www.tensorflow.org/?hl=pt-br>
[3]<https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4>

Figure 4.5 – Learning curve of Experiment 3



Source: The Author

4.1.6 Environment Configuration

The models were trained in the *Google Colaboratory pro* [4] virtual environment, with 12.7 GB RAM and 15GB GPU.

4.2 Results

The summary of the results can be seen in Table 4.4, which presents the performance of our BERT-based approach in the four performed experiments. The results support two major conclusions about the model's performance and their relation with the dataset. The first conclusion is that, as expected, by reducing the number of classes and aggregating them our approach achieves a higher performance. When comparing the performance in experiment 1, considering 5 classes, with the performance in experiment 2, considering 3 classes, we observe an increase of 14% in macro-f1. When comparing the performance in experiment 2 with the performances in experiments 3 and 4, we observe an increase of 14% and 8%, respectively. This effect is expected because by aggregating classes that often share many instances incorrectly classified with each other implies the reduction of classification errors. Figures 4.6-4.9 support the analysis of how many incorrect instances are shared among classes.

The second major conclusion is that, by comparing experiments 3 and 4, we can conclude that the neutral class is biased towards negative sentiments. This conclusion is supported by comparing the performance of experiments 3 and 4, where experiment 3 reaches 88% macro-f1 compared with 82% from experiment 4. This conclusion is

[4] <https://colab.research.google.com/>

Table 4.4 – Result of the experiments

Experiment	*Accuracy*	*Macro-f1*	*Macro-Precision*	*Macro-Recall*
Experiment 1	0.59	0.59	0.57	0.61
Experiment 2	0.74	0.74	0.74	0.74
Experiment 3	**0.88**	**0.88**	**0.88**	**0.88**
Experiment 4	0.82	0.82	0.77	0.87

Source: The Author

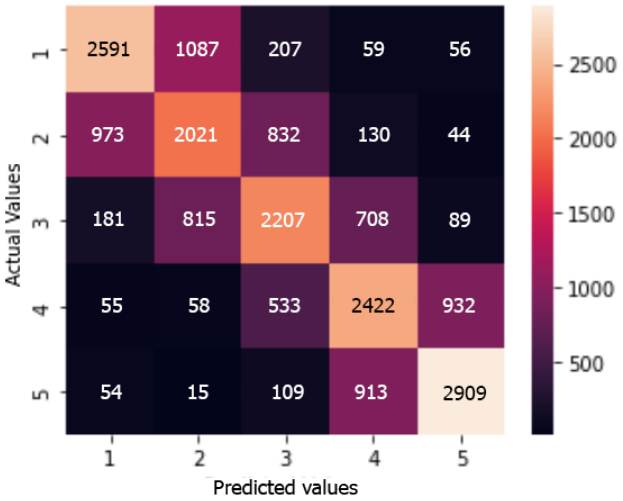
evidenced also when we analyze the confusion matrices of the experiments. Figures 4.6-4.9 represent the confusion matrices produced by experiments 1-4, respectively. These confusion matrices were generated using the fold that achieved the macro f1-measure that was closest to the average macro f1-measure of the model, considering all the folds.

In Figure 4.6 it is possible to see how the positive classes, 4 and 5, are wrongly predicted as each other, suggesting a strong similarity of textual patterns presented in these two classes. The same phenomenon can be observed When we analyze the negative classes 1 and 2. Besides that, we can observe also that the number of samples of class 3 that are wrongly classified in classes 1 and 2 is higher than those wrongly classified as 4 and 5, which evidences the negative bias of class 3.

Figure 4.7 represents the confusion matrix for Experiment 2, where we aggregate the classes 1 and 2 for representing an overall negative class and we group the classes 4 and 5 for representing an overall positive class. This confusion matrix also shows that the neutral class is predicted as negative more often than positive.

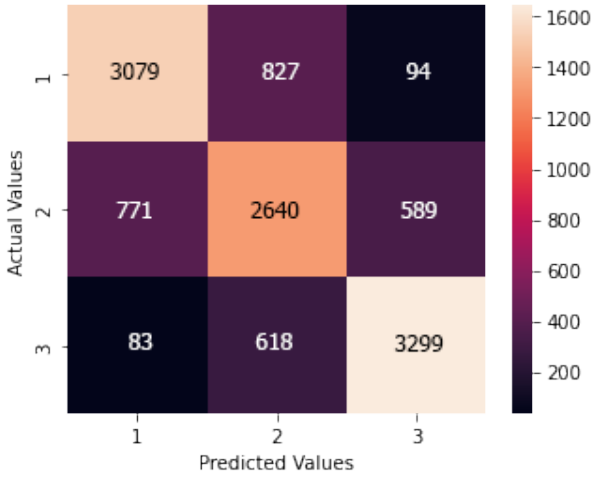
Finally, when comparing figures 4.8 and 4.9 the first one aggregating the neutral reviews with negatives and the second aggregating it with positives, we see that indeed there is a significant difference between the predictions for neutral reviews. In experiment 4 there are 3 times more false positives than in experiment 3. This strongly indicates a bias in the samples of the neutral class towards negative sentiments.

Figure 4.6 – Confusion matrix for experiment 1



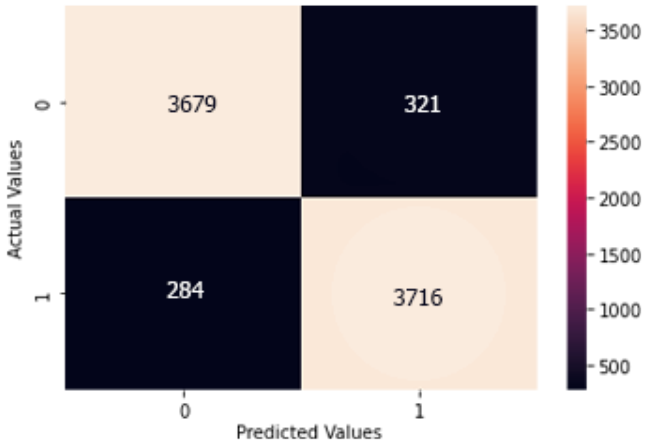
Source: The Author

Figure 4.7 – Confusion matrix for experiment 2



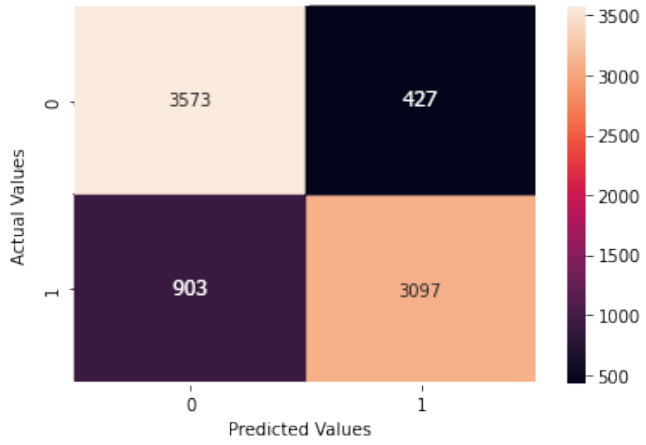
Source: The Author

Figure 4.8 – Confusion matrix for experiment 3



Source: The Author

Figure 4.9 – Confusion matrix for experiment 4



Source: The Author

5 CONCLUSION

In this work, we investigated the performance of a BERT-based classifier in the task of sentiment analysis of product reviews in four different in an Amazon book review dataset, which includes reviews rated according to a 1-5 scale.

We performed four experiments, using the same basic BERT-based classifier with a similar training configuration. In each experiment, we aggregated the original classes in different ways. As a general result, our BERT-based classifier achieved good performances in some scenarios. The results showed that, as expected, decreasing the number of classes increased the performance of our classifier. The best performance was achieved in Experiment 3, with an F1 score of 88%. In this experiment, we aggregated the negative and neutral classes (classes 1,2, and 3) into one single negative class and the positive classes (4 and 5) into a single positive class. Our experiments suggest also that neutral reviews are biased toward negative sentiments. Besides that, we can also observe that it is relatively easier to predict the ratings of reviews with a strong sentiment associated, such as classes 1 and 5, than it is for the neutral class, 3.

As part of future work, the summary of the reviews can be incorporated to give more context to the reviews. The results from these models could be compared with other traditional machine learning approaches, such as SVM. Future works can also compare the performance of our BERT-based classifier with classifiers built using other pre-trained language models available in the literature. Moreover, in the future, we can also investigate the performance of classifiers created as ensembles of different pre-trained language models.

REFERENCES

BALAKRISHNAN, V. et al. A deep learning approach in predicting products' sentiment ratings: a comparative analysis. **J Supercomput**, v. 78, 2022.

BILAL, M.; ALMAZROI, A. A. **Effectiveness of Fine-tuned BERT Model in Classification of Helpful and Unhelpful Online Customer Reviews**. 2022.

DEVLIN, J. et al. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. arXiv, 2018. Available from Internet: <https://arxiv.org/abs/1810.04805>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

HAQUE, T. U.; SABER, N. N.; SHAH, F. M. Sentiment analysis on large scale amazon product reviews. In: **2018 IEEE International Conference on Innovative Research and Development (ICIRD)**. [S.l.: s.n.], 2018. p. 1–6.

KARTHIKA, P.; PALANISAMY, K. **Feature Selection Methods in Sentiment Analysis and Sentiment Classification of Amazon Product Reviews**. 2016. 225-230 p.

LAN, Z. et al. **ALBERT: A Lite BERT for Self-supervised Learning of Language Representations**. 2020.

LIU, Y. et al. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. 2019.

NEETHU, M. S.; RAJASREE, R. Sentiment analysis in twitter using machine learning techniques. In: **2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)**. [S.l.: s.n.], 2013. p. 1–5.

NI, J. **Amazon Review Data (2018)**. 2018. Available from Internet: <https://nijianmo.github.io/amazon/index.html>.

POLIGNANO, M. et al. **AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets**. CEUR, 2019. Available from Internet: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074851349&partnerID=40&md5=7abed946e06f76b3825ae5e294ffac14>.

POOLE, D.; MACKWORTH, A.; GOEBEL, R. **computational intelligence: a logical approach**. 1998.

POTA, M. et al. **An Effective BERT-Based Pipeline for Twitter Sentiment Analysis: A Case Study in Italian**. 2021. Available from Internet: <https://www.mdpi.com/1424-8220/21/1/133>.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: a modern approach**. 3. ed. [S.l.]: Pearson, 2009.

SAIF, H. et al. **On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter**. Reykjavik, Iceland: European Language Resources Association (ELRA), 2014. 810–817 p. Available from Internet: <http://www.lrec-conf.org/proceedings/lrec2014/pdf/292_Paper.pdf>.

TAPARIA, A.; BAGLA, T. **Sentiment Analysis: Predicting Product Reviews' Ratings using Online Customer Reviews**. 2020. Available from Internet: <https://ssrn.com/abstract=3655308>.

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

WU, F. et al. **Sentiment Analysis of Online Product Reviews Based On SenBERT-CNN**. 2020. 229-234 p.

APPENDIX A — EXAMPLE OF LITERARY REVIEW

The Fellowship of the Ring, the first part of The Lord of the Rings Trilogy by J.R.R. Tolkien, is a fantasy focusing on a hobbit by the name of Frodo Baggins. Frodo is the favorite nephew of his Uncle Bilbo Baggins, the legendary hobbit who set foot upon the most storied journey of all hobbit folklore. Bilbo's story is told in The Hobbit, which is considered the prelude to this trilogy. Near the end of Bilbo's travels he came upon the most powerful ring in all of the land, known as the Ring of Power.

Being Bilbo's favorite relative, Frodo inherited the Ring, among various other items, when Bilbo decided to "retire" and move to a different land after his 111th birthday. Frodo had known of the Ring through stories relayed by Bilbo, however he learned all that was to be known of the Ring from the mighty wizard, Gandalf the Grey. What Frodo discovered was that very little was known of the One and that its mystery was only exceeded by its power. Gandalf told of the very evil implications about the Ring and that the dark Sauron was in pursuit of the One. Frodo must set foot on a journey to dispose of the Ring in the only place which it can be destroyed at the very center of Sauron's evil kingdom in Mordor atop Mt. Doom. Frodo is joined initially by his faithful servant, Sam, along with Gandalf. Along the travels, the crew encounters several new characters and conquests, both advantageous and perilous and is continuously having new light shone on the mystery of the Ring and the true meanings of the journey. This book is the classic story of Good versus Evil.

The one aspect that really sets this book apart from the pack is Tolkienś excellent language and diction. His language is very descriptive, yet to-the-point. The reader can see what is happening without actually viewing the actions. Tolkien's language is very poetic and would be better served had the book been printed in calligraphy. The Lord of the Rings Trilogy will hit the big screen in successive Christmases starting in 2001 with The Fellowship of the Ring. However, this story is sure to be far better using one's own imagination rather than the impressions of someone else. Read the book before seeing the movie.