Sentiment Analysis and Classification of Hotel Opinions in Twitter With the Transformer Architecture

Sergio Arroni, Yeray Galán, Xiomarah Guzmán-Guzmán, Edward Rolando Núñez-Valdez*, Alberto Gómez

Department of Computer Science, University of Oviedo, Oviedo (Spain)

Received 7 May 2022 | Accepted 25 October 2022 | Early Access 7 February 2023

Abstract

Sentiment analysis is of great importance to parties who are interested is analyzing the public opinion in social networks. In recent years, deep learning, and particularly, the attention-based architecture, has taken over the field, to the point where most research in Natural Language Processing (NLP) has been shifted towards the development of bigger and bigger attention-based transformer models. However, those models are developed to be all-purpose NLP models, so for a concrete smaller problem, a reduced and specifically studied model can perform better. We propose a simpler attention-based model that makes use of the transformer architecture to predict the sentiment expressed in tweets about hotels in Las Vegas. With their relative predicted performance, we compare the similarity of our ranking to the actual ranking in TripAdvisor to those obtained by more rudimentary sentiment analysis approaches, outperforming them with a 0.64121 Spearman correlation coefficient. We also compare our performance to DistilBERT, obtaining faster and more accurate results and proving that a model designed for a particular problem can perform better than models with several millions of trainable parameters.

Keywords

Artificial Intelligence, Machine Learning, Natural Language Processing, Sentiment Analysis, Transformer Architecture.

DOI: 10.9781/ijimai.2023.02.005

I. Introduction

In the last few years, there has been immense growth in the field of Natural Language Processing (NLP) and, especially, in the application of machine learning methods to NLP problems.

With the increase in popularity of deep learning, for some years the focus was on the research of neural network structures that make use of convolutional layers [1] and recurrent layers [2] for language understanding and processing. These architectures brought about a big wave of research for their application to NLP, since they allowed much more detailed representations that the standard feed-forward models [3]. Convolutional networks allow looking at text by parts through filters, which then are aggregated for a global interpretation. Recurrent networks, on the other hand, process the text input sequentially but, aside from the part of the input being currently processed, they take into consideration outputs from previous parts as an additional input, hence the name recurrent networks.

A few years ago, however, the proposal of attention-based neural network models by Vaswani et al. [4] has shifted great part of the research in deep learning for NLP towards the development of transformer structures and pre-trained models with hundreds of millions of trainable parameters that only require some fine-tuning training to be applicable to a wide range of NLP tasks [5]–[7].

* Corresponding author.
E-mail address: nunezedward@uniovi.es

However, NLP can be of special interest to businesses or parties who are interested in knowing the public opinion about something in general purpose social networks (e.g., a restaurant might be interested in knowing whether customers like or dislike its food, but not so much in being able to generate AI-written text or in artificial question answering). For that purpose, a simpler and smaller model might suffice or even obtain more accurate results than models which are pre-trained for multiple NLP tasks.

We explore sentiment analysis, which is a subfield in NLP that deals with processing a piece of text and obtaining the general sentiment included in it. Several advances have taken place lately towards more precise sentiment analysis, ranging from basic and rudimentary approaches [8] to complex neural network systems.

In this work, we implement a neural network system using the state-of-the-art attention-based Transformer structure, with a dramatically lower number of trainable parameters and size than those of the previously mentioned pre-trained models. In contrast to other machine learning NLP approaches like recurrent [2] and convolutional [1] neural networks, which tend to lose information when the text is too large, this structure manages to process the text input in a single iteration, which increases the speed and the ability to understand the context of the whole text.

Thus, our aim is to make use of the Transformer neural network architecture and obtain a model that greatly improves the prediction accuracy of those more basic methods, while not resorting to the complexity of training millions or billions of parameters, proving that a simpler and faster model crafted for the task at hand can perform better if trained for a particular problem. We tackle the problem of

predicting the general opinion about hotels in Las Vegas from Twitter data, making use of the dataset provided by Philander & Zhong [8].

The study by Philander & Zhong [8] served as a motivation for the possibility of improving the obtained results, so we came up with the idea of trying to use a neural network to predict the sentiment in the tweets. Our first approach to this work consisted in following a similar approach to the original study while adding the computation of bigrams and trigrams, but the results obtained were not significantly better compared to those of the original work. We improved on this by adding a machine learning algorithm. Knowing the real-world applications of sentiment analysis, we want to obtain the best possible result for a problem we found interesting.

In a first instance, we opted for using a HuggingFace pre-trained DistilBERT [6] model, which can tokenize and prepare the input and uses a transformer neural network model. However, as it is a pre-trained model, its structure and parameters are not modifiable, which in combination with the fact that the model contains millions of parameters for general NLP tasks that are not needed for our specific problem, motivated our decision to create our own attention-based model from scratch. We then compared our model to DistilBERT for evaluation against a strong state-of-the-art model.

The remainder of this work is structured as follows: Section II discusses the related work in the state of the art about NLP, sentiment analysis, machine learning applied to NLP and the transformer architecture. Section III presents our proposal, including the dataset used, the model architecture and the evaluated metrics. Section IV details the experiments carried out and the results obtained in terms of the metrics and execution time. Lastly, Section V offers our conclusions and some proposed future work in relation to our study.

II. Literature Review

In this section, we discuss the main aspects that we deal with in this work: sentiment analysis, deep learning, and the transformer architecture for NLP.

A. Natural Language Processing and Sentiment Analysis

NLP has attracted a lot of attention in recent times, and great advances are being achieved in this field. NLP is a field of study that is being researched since more than 50 years ago and it is one of the most widely spread topics in which artificial intelligence is applied. Its purpose is to enable computers to understand words written by humans and process them to reach conclusions related to the problem at hand.

NLP approaches usually involve several linguistic aspects, like semantics, phonology [9], morphology [10] or syntax [11] of written or spoken natural language. Nowadays, however, most of the research is oriented towards the application of machine learning to NLP problems [3]. Some deep learning models are developed almost exclusively for NLP tasks [4], considering the needs for text processing and sequence generation, and brought a breakthrough to the field achieving great results in several NLP tasks like translation or question answering.

We can find two different cases in NLP: natural language understanding and natural language generation. A particular case of natural language understanding is text classification, which deals with the problem of assigning a category to a text. Sentiment analysis can be seen as a generalization of text classification, as it attempts to analyze a piece of text and find the general sentiment included in it. Other subfields in text classification are topic detection or language detection. We will focus on sentiment analysis, as our goal is finding the general opinion present in a text, that is, assigning a label to a text.

Going further into the field of sentiment analysis, it tries to identify and obtain subjective information from a given text as input. This analysis provides us with information that gives us a result of emotional tone, such as positive, negative, happy, sad, angry, etc.

There are mainly two state-of-the-art approaches to sentiment analysis. One of them is the lexicon and rule-based method, which consists in making a decision on the sentiment in the tweet according to whether specified conditions are met [12]–[14]. However, most state-of-the-art approaches to sentiment analysis nowadays include the previously mentioned deep learning models [5], [15], [16], one of which is the transformer architecture that we study in this work. We will conduct experimentation comparing methods from both groups and attempt to show that our deep learning model has greater potential.

By obtaining this result we can categorize the text within an emotional spectrum, being able to group them by feelings. This has a wide range of direct applications, including product or service reviews [17], analysis of social network data [18], marketing and branding [19], financial analysis and forecasting [20], detection of emotions in conversations [21] and many others.

As shown in the previously mentioned studies, sentiment analysis also finds great use in fields that do not necessarily have a direct relationship with computer science and is often used for the processing and analysis of Big Data.

As Philander & Zhong [8] say in their work, this analysis that we do can have a great impact on the hotels we analyze, as it provides them with very useful information that they would take a long time to get by hand. Sentiment analysis as a whole possesses great applications in industry, in fields where customer opinions are of great relevance such as product or service reviews on the web [22]–[25] or prediction of stock markets and prices [26], [27] and even in fields like opinion analysis in politics [28] or medicine [29].

B. Deep Learning for Natural Language Processing

Machine learning is being applied to NLP tasks for over two decades. Some years ago, standard machine learning approaches like random forests [30] and support vector machines [31] were the state-of-the-art methods for learning text representations.

However, with the increase in computational power and the popularity of neural network models, deep learning soon took over the field. Apart from the conventional feed-forward models, the development of convolutional and recurrent neural network models brought about a whole new world of approaches to the processing of natural language, both in the form of text and voice [32].

Convolutional neural networks, as first proposed by Kunihiko Fukushima [1], process an input by looking at different parts of the whole through a filter and shifting the filter through the input. Those results are then aggregated in different ways for obtaining the desired output. As could be expected by this brief description, convolutional neural networks have found the most success in the processing of images or computer vision [33]. However, the model can be applied to NLP in the same way, as the processing of text greatly benefits from a partial look at different parts of it for computing a representation of the input [34].

On the other hand, recurrent neural networks, as proposed by Rumelhart et al. [2], process an input iteratively one by one, but compute each representation by using the previous output as well. Thus, the final output contains information about the whole sequence. However, the earliest information is sometimes lost because of the vanishing gradient problem. To address this problem, researchers come up with models like LSTM (Long Short-Term Memory) [35], which introduces an additional input that contains the previous

unprocessed inputs. As expected, this model has found great success in the processing of sequential data, one of which is text.

One of the problems with those models, however, is the concept of distance in the sequences, e.g., the first word in a sentence will always be furthest away from the last word when being processed by recurrent models, despite that not being necessarily the case for meaning in language processing, since two words can be closely related even if there are several words between them. For solving this, the attention mechanism is introduced, which is able to compute dependencies equally between every single element in the sequences. This mechanism is mostly used in combination with recurrent or convolutional models until 2017, when the transformer architecture is introduced by Vaswani et al. proving that using attention is enough and that recurrency and convolutions are not needed [4].

C. Transformers

Transformers are a neural network architecture based solely on the attention mechanism which was introduced first in a work by Vaswani et al. in 2017 called Attention Is All You Need [4].

Transformers, like other neural network architectures [36], are based on an encoder-decoder model, where the encoder is responsible for analyzing a sequence of input data and obtaining an encoding, and the decoder is responsible for obtaining an output from the encoding. As its structure suggests, this model is mainly aimed towards the translation or transformation of the input into a similar output [37], that is, computing different representations of the input data. However, the decoding process can be adapted to a much greater variety of tasks.

In the field of NLP, the transformer architecture is used to solve multiple tasks including text classification [38], translation [39], question answering [40], summarization [41] or text generation [42]. In this study, we focus on text classification.

As described in the study by Vaswani et al. and applied to our problem, given an embedding matrix E with embeddings of size d_e for tokenized texts, self-attention is calculated as described in (1).

$$a(E) = \sigma\left(\frac{EE^T}{\sqrt{d_e}}\right)E \tag{1}$$

As we can see in equation (1), σ is the softmax function. For multi-head attention, tree matrices Q, K, V will be linearly projected from E for each head $i \in \{1, ..., h\}$ by means of transformation matrices W_i (three in total for each head). Therefore, there will be $3h$ matrices that will be learned by the model. Attention is computed then not as self-attention but as regular attention, as shown in (2).

$$a(Q_i, K_i, V_i) = \sigma\left(\frac{Q_i K_i^T}{\sqrt{d_e}}\right)V_i \tag{2}$$

The results are then concatenated and projected back to the original shape. The original proposal uses 8 attention heads. This process does not imply a big increase in the total operation time as the size of the computations in each head is reduced by the projections. Finally, the outputs are passed through densely connected layers until the final probabilities for text classification are obtained.

In other architectures like recurrent networks or convolutional networks, the text input is analyzed sequentially or by segments, which often causes the loss of early information due to the problem of vanishing gradient [43]. In the transformer architecture, by only making use of attention as described above, the whole text is processed in one go, which allows greater precision for analyzing the relative position and meaning of big texts.

Not only that, but transformers have also found great success in other fields that do not involve the processing of text, like the processing of audio, mainly in the field of speech recognition [44], the processing of images [35], [36] or even other kinds of sequence generation like chemical chains [45]. Lin et al. [46] present a survey of the most relevant contributions to the attention mechanism and the transformer architecture over the past few years.

In the field of NLP, most research has been centered around developing pre-trained models [47]. These models are still being researched as of today for the creation of more accurate language representation models that can be fine-tuned for a large number of different problems.

In this work, our goal is to craft a transformer architecture model much simpler and smaller than the mentioned models, and show that for specific problems, there is no need to obtain a model that can contain a whole language representation, just the information needed for said problem. Additionally, we want to show that the transformer architecture is very powerful no matter the size of the model and that it also works for smaller problems without millions or billions of trainable parameters.

III. Proposal

This section details our proposal, offering a view of the process of preparing the dataset, an in-depth description of the model created and the details of the computation of ratio score and evaluation metrics used for comparisons.

A. The Datasets

This subsection details the datasets used for this work: the tweet dataset, the Datafiniti review dataset [48] and the Amazon review dataset [49].

1. Tweet Dataset

In our study, we use the tweet dataset offered by Philander & Zhong [8]. The dataset contains the tweets tagging hotels in Las Vegas within two periods of time: from August 16, 2013, to September 13, 2013, and from October 25, 2013, to November 15, 2013. However, no kind of labelling was included, so we opted for manually classifying some of them to be able to employ a machine learning algorithm. We started with 2701 classified tweets, 2014 of which are positive, 250 are negative and 437 are neutral.

We use this dataset as the main dataset so that we can compare our method with the method proposed by Philander & Zhong under similar data, so that the results are not biased towards our model, and we can make a fair comparison between their manual classification method and our deep learning model.

We want to clarify that the manual classification took into account emojis, exclamation marks, ironies and other colloquial expressions. In order to carry out this classification, a standard was followed as it was done by several members of the team in different periods of time. This standard is divided into four different classifications for each tweet:

Positive tweets: we decided to classify as positive tweets those that had a clear and undoubtedly positive feeling towards their stay at the hotel. We encountered many tweets of people that are just happy or attracted by some celebrity or contest that is popular at that moment. Those tweets do not necessarily have anything to do with their opinion on the hotel and therefore have not been classified as positive. Some other tweets talk about their excitement of staying in the hotel for the first time, these tweets have not been categorized as positive either as we understand that they do not provide any information on features of the hotel or their stay as they have not been there yet. Tweets that speak positively about the hotel's facilities or the hotel's service have been classified as positive.

Negative tweets: following the pattern of positive tweets, both events and famous people mentioned negatively in the tweets have

not been classified as negative. There was a problem mentioned in the work by Philander & Zhong [8], which alerted us of the low number of negative tweets. Due to this, in this classification we were less strict in classifying a tweet as negative, classifying one as such at the slightest hint of doubt or discomfort from a customer.

Neutral tweets: in this category are all tweets that do not meet the conditions for the other sentiments since we understand as neutral all tweets that do not talk about the hotel or do not say anything significant about it. Tweets that only tag the hotel but do not explicitly talk about it are classified as neutral since they do not reflect a sentiment on the hotel but are useful nonetheless for learning to separate opinions about the hotel from opinions about something else. Tweets that say both positive and negative things about the hotel but do not clearly emphasize one of the two are also classified as neutral.

Tweets that did not fit in any field were deleted, as there are some that are either empty or do not include any type of information, neither about the hotel nor about any other topic, or were written in a language other than English. There were also tweets that were too ambiguous, and we would not be able to add them to any of the sentiments described, so in order not to include unnecessary noise in the training data, these tweets were also deleted. Table I shows some examples of tweets as manually classified by us.

TABLE I. Examples of Tweets Classified by Hand

Positive
"@AriaLV loved every minute about staying at the Aria very safe modern and overall great atmosphere will stay there again!!" *"So excited for Vegas now! Looking forward to staying at the best hotel on the strip @TheMirageLV"* *"What a beautiful day in #Vegas. The sun is shining our pool is #Shimmering and we have #rooms to sell @TropLV ! What could be better?"*
Negative
"Hey @HardRockHotelLV your customer service leaves MUCH to be desired. If #Pubcon is smart you won't be the partner hotel next year." *"@RivieraLasVegas Did you ever replace the lamp in room 3533? Might wanna clean the puke off the walls too. So gross!"* *"@AriaLV @myVEGAS no.. but a nice stay would thankful for once :)"*
Neutral
"*Hey @TropLV I love your hotel, but the service in your beach cafe was atrocious tonight. You guys are better than that."* *"Went to @Rock_Vault @LVHHotelCasino...photo with @RobinMcauley from Survivor #80srock"* *"The Eiffel Tower at @parisvegas will award lucky 10 Millionth visitor with trip for 2 to Paris, France. Learn..."*

2. Datafiniti Review Dataset

The Datafiniti reviews [48] in the dataset were categorized into star ratings between 1 and 5. We refer to the Likert scale [51], which stipulates that on a 5-point scale, each extreme represents an opposing opinion, in this case, positive and negative, at the intersection of these extremes, point 3, represents an opinion that is indifferent to the subject or, in many cases, neutral.

Thus, following this scale, we decided to classify 1-to-2-star reviews as negative, 3-star reviews as neutral and 4-to-5-star reviews with positive sentiment. In the end, the dataset has 6048 negative reviews, 5709 neutral reviews and 22429 positive reviews.

3. Amazon Review Dataset, Hugging Face

Similarly, to the Datafiniti dataset [48], the Amazon dataset [49] contained reviews with star ratings. We used the same method of classification as before, marking reviews from 1 to 2 stars as negative, reviews with 3 stars as neutral and those from 4 to 5 stars as positive.

B. The Attention-based Transformer Model

We use a neural network structure based on self-attention as proposed by Vaswani et al. [4]. The transformer architecture makes use of self-attention, which by computing the dot product of every pair of tokens, is able to process relationships between elements at any distance, something that other models like recurrent and convolutional networks struggle with. This is a very important feature in the representation and understanding of natural language, which is the main reason why we opt for the transformer architecture in this study [4].

According to Vaswani et al. [4], the computational efficiency per layer and precision also improves that of other models, which added to the fact that our available computational power is not great, and tweets are short and require higher precision, serves as another reason for our choice. We consider using other architectures, classical recurrent networks like LSTM or convolutional networks, but since those are nowadays less efficient and, unlike transformers, have trouble scaling to bigger inputs, we in the end opted for the transformer architecture being a successful newer and very interesting approach.

The network structure consists of a single transformer block with 11 attention heads as described in their study, with the main difference being that we use learned embeddings of dimension 12 for positional encoding (as well as token encoding) instead of sine and cosine functions. These vectors of dimension 12 are able to capture the information and context of the tokens in a continuous space without unnecessarily over-increasing the complexity of learning the transformation. Token embeddings and position embeddings are added and fed to the multi-head attention layer, the output of which is then passed through a dropout layer and a normalization layer before being used as an input for the feed-forward layers. The fully connected layers inside the transformer blocks contain 768 neurons as proposed by Devlin et al. [25]. After experimenting with different sizes, we decided against decreasing the size of these layers, since they contribute greatly to the mapping of features extracted by the attention layers to the outputs. Residual connections are used around the multi-head attention layer and the feed-forward layer inside the transformer block. The output is fed to an average pooling layer to reduce its dimensionality to a 1d vector of size 12 (the dimension of the learned embeddings) which integrates all the information obtained from the transformer block, and through a 16-neuron feed-forward layer that is fully connected to the final 3 probabilities, which are obtained by a SoftMax function. Fig. 1 offers a visualization of the described model structure.

The data goes through a preprocessing stage before entering the neural network. The tweet texts are first cleaned of all tags starting with "@" and hyperlinks starting with "http" since these only add noise to the tweets, and then we make use of the Keras [52] text tokenizer for tokenizing the tweets in our dataset. We decide to leave the hashtags starting with "#" since they can carry information related to the sentiment expressed in the tweets [53]. The vocabulary size used for tokenizing the tweets is 20,000 words, which we found to be an appropriate size for our problem, given that tweets are usually not very long, but they contain a large range of words that are not necessarily part of the English language (like abbreviations, Internet slang, emoticons, and so on). Tokenized tweets are then padded to a length of 20 tokens. This length was chosen for allowing a slightly above average number of words in a maximum-length tweet (the average for the English language was 17 words when the maximum tweet length was 140 before being increased to 280 in 2017) while not dramatically increasing the neural network input size. For the analysis of longer format reviews or posts, the text will still be padded to 20 tokens, possibly causing some loss of information in some cases where the sentiment is expressed in the latter part of the text. In those cases,

we recommend increasing the maximum input length and slightly increasing the complexity of some layers so that the whole text can be processed with similar representational power.

Our contribution consists in the use of a manually created neural network based on the transformer architecture. We perform a study of the different parameters and structures of the model using a grid search method and arrive at the model described above. With this approach, we manage to achieve better results than those obtained in the study by Philander & Zhong [8] and other more complex models while drastically lowering the number of trainable parameters and thus, the time needed for training and making predictions.

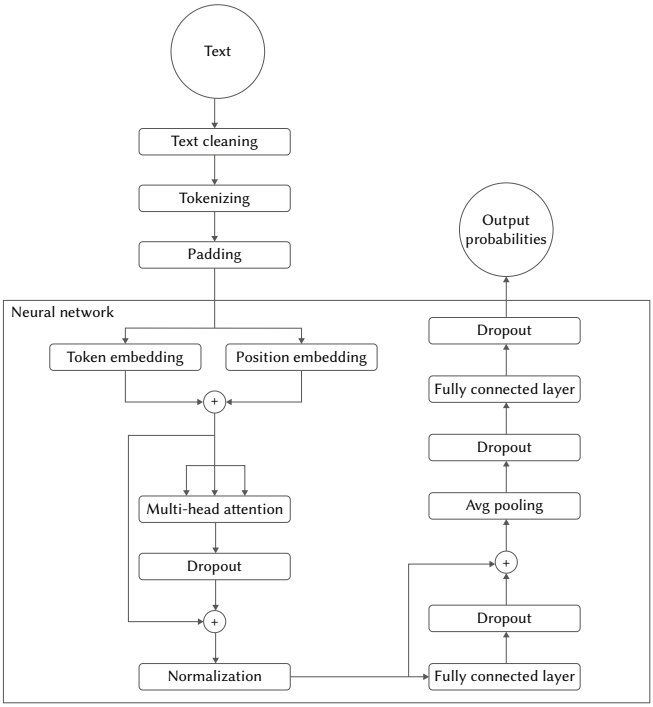

Fig. 1. Model architecture.

C. Ratio Score

The resulting classified tweets are used for computing a ratio score for each of the hotels in the dataset. The calculations are similar to the ones proposed by Philander & Thong [8]. They propose the ratio score of a hotel as the quotient between the total number of positive tweets and the total number of negative tweets related to a hotel. It is formally defined in (3), where n is the total number of tweets related to hotel h, p_i is the number of positive words in tweet t and n_i is the number of negative words in tweet i. For a predicate p, the function 1_p is defined in (4).

$$score(h) = \frac{\sum_{i=1}^{n} \mathbf{1}_{p_i - n_i > 0}}{\sum_{i=1}^{n} \mathbf{1}_{p_i - n_i < 0}} \tag{3}$$

$$\mathbf{1}_p = \begin{cases} 1 & p \\ 0 & \neg p \end{cases} \tag{4}$$

We use this definition in some of our experiments for a fair comparison between the method proposed by Philander & Thong [8] and our model.

However, since our model does not only obtain the label of the text, but instead obtains the probability that the model considers for each of the labels, we propose a new method for calculating the ratio score of each hotel that is much more flexible and makes use of the probabilities to obtain a much more informative representation of the ratio score.

Instead of computing the ratio between the number of positive tweets and the number of negative tweets for each hotel, we consider the probability obtained by the model of each label for each tweet. That is, the new ratio of a hotel is calculated as shown in (5), where n is the total number of tweets related to hotel h and l_i is the label obtained for tweet i.

$$score(h) = \frac{\sum_{i=1}^{n} P(l_i = positive)}{\sum_{i=1}^{n} P(l_i = negative)} \tag{5}$$

The regular ratio score is a very rigid method of scoring hotels since it can only classify tweets as positive or negative and every tweet has the same weight towards the final score. Since not every positive tweet is equally positive and not every negative tweet is equally negative, we put forward this method that computes "how positive" and "how negative" each tweet is, regardless of its final classification. Thus, we intend for this method to perform better at predicting relative performance between hotels than the regular ratio score.

D. Evaluation Metrics

In the following experimentation, we propose the usage of two metrics to evaluate and compare the performance of different models: accuracy on the validation set and the Spearman correlation coefficient with a TripAdvisor ranking.

Validation accuracy is a very popular metric for the evaluation of machine learning algorithms in classification problems. It computes the ratio of correctly predicted samples among all the samples in the validation set, that is, not including the samples used for training. More formally, the validation accuracy is defined in (6), where V is the validation set, y_i is the expected output of sample i and $\hat{y}_i$ is the predicted output for sample i.

$$accuracy = \frac{\sum_{i \in V} \mathbf{1}_{y_i = \hat{y}_i}}{|V|} \tag{6}$$

We make use of this metric as a method of gauging how reliable the models are at correctly predicting the sentiment in individual tweets.

On the other hand, for a more global evaluation, we compute the Spearman correlation coefficient. We choose this metric for a fair comparison to the study by Philander & Zhong, since it is the metric that they propose for comparing rankings. This coefficient is defined as the Pearson correlation coefficient between the ranking of the values in each variable. We can take the Pearson coefficient as the expression in (7), where σ is the standard deviation and cov is the covariance.

$$r_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \tag{7}$$

Equation (8) then shows the definition of the Spearman correlation coefficient, where $R(x)$ is the ranking of variable x ordered by its value.

$$\rho_{X,Y} = r_{R(X),R(Y)} \tag{8}$$

We compute this coefficient by taking as X the ratio score of every hotel, calculated as described in the previous section, and taking as Y the TripAdvisor score obtained from the study of Philander and Zhong [8]. The coefficient is always contained in the interval [-1, 1], meaning a weak correlation in values close to 0, a strong negative correlation (rankings move in opposite directions) in values close to -1 and a strong positive connection (rankings move in the same direction) in values close to 1.

This metric obtains a more general view on how accurate the models are at predicting the relative quality of hotels. We opt for using both proposed metrics as they serve as a means of evaluating different aspects of the goodness of the models. However, in the experimentation phase, we put more focus on the Spearman coefficient for comparison purposes, given that the more general goal of the study [8] we aim to compare our models with, is obtaining an accurate overview of the relative performance of hotels between themselves.

IV. Experimentation

Our goal is the application of machine learning to tweet labelling to improve the results obtained by Philander & Zhong [45]. This study makes use of a random selection of tweets from the dataset presented by them, containing a total of 2701 tweets, of which 2014 are positive, 250 are negative and 437 are neutral. Only part of the tweets has been selected so that enough of them are left for predicting. We use a random selection of 80% of those tweets as training samples for our machine learning system and the remaining 20% for testing.

After training, the remaining tweets are classified and labelled by the system. In another approach for attempting to train our model with more data, we also make use of an external dataset from Datafiniti [48] containing hotel reviews and classify our tweets by the model trained with said dataset. After all the tweets are classified, we obtain the ratio score of each of the hotels with the two methods described in the proposal. With these ratios we compute the proposed metrics, and, in the end, we compare the complexity and time needed for each model. In brief, we train the same model with the two described datasets separately and predict the remaining tweets for comparing the results.

As the datasets are imbalanced towards positive labels, we attempt to mitigate the difference in number of samples by class by applying greater training weights to the samples that have "negative" and "neutral" labels. However, we find that despite not affecting the average results much, it has a negative impact on the prediction for relative performance of hotels. We believe this is possibly caused by overfitting on negative and neutral tweets due to applying greater weights to each of them, which would in turn cause the model to mistakenly predict some tweets as negative or neutral that are related to the overfit samples. When this happens for a tweet that is closely related to one particular hotel, it can cause its ratio score to drop dramatically. For this reason, we leave the number of tweets for each label as is, since we observe that the models still manage to correctly learn to predict negative and neutral tweets as well.

A. Hardware

We run the computational experiments on a 64-bit Windows Server 2016 with two Intel Xeon Silver 4208 CPUs at 2.1GHz and 6GB of RAM memory. It should be noted that no CUDA-compatible graphics unit is present, so all computing is done in the specified CPU. Every instance of the experiments is run on the same machine for a fair comparison of time and efficiency.

B. Metric Measurement

This section presents the results obtained in terms of validation accuracy and Spearman correlation coefficient.

1. Single Results

In a first instance of experimentation and for a better adjustment of our model, we carry out a single execution of the training and evaluation process several times. In order to ensure that the results are deterministic and that they are not influenced by random variation, we employ a random seed and execute every instance over it.

Philander & Zhong [8] offer the Spearman correlation coefficient (ρ from now on) between their obtained score for each hotel and the TripAdvisor score at the time as the validation metric. For a fair comparison, in a first instance, we compute the ratio score as proposed by them. With the same TripAdvisor score (obtained from their article [8]) and our own obtained ratio score for each hotel, we manage to obtain a ρ of 0.601 (p = 0.0001) between our hotel ranking and the TripAdvisor ranking, which we consider a high correlation considering that the mean score among TripAdvisor reviews does not always necessarily match the general sentiment expressed by people on Twitter. Table II shows some examples of the classification of tweets by our model.

TABLE II. Examples of Tweets Classified by Our Model

Positive
"*VEGAS I just got back and stayed at and it was AMAZING Vegas is perfect for bachelorette parties*" - Pos: 86.93%; Neg: 2.29%; Neu: 10.78%. "*loved every minute about staying at the Aria very safe modern and overall great atmosphere will stay there again*" - Pos: 82.36%; Neg: 3.56%; Neu: 14.08%. "*Absolutely loved the rooms Luxury*" - Pos: 69.1%; Neg: 7.22%; Neu: 23.68%. "*I've stayed in a lot of nice places but might just be the nicest... Only problem is I get lost ALL THE TIME. #itshuge*" - Pos: 41.99%; Neg: 22.26%; Neu: 35.73%.
Negative
"Where is the 'cl*ean window*' button in my room? ;)" - Pos: 23.77%; Neg: 41.06%; Neu: 35.17%. "*Good morning from It's a beautiful day, but I think the windows need washing :/*" - Pos: 33.14%; Neg: 36.08%; Neu: 30.78%. "*#APALAcon13 has joined workers ; No contract, no peace #CantStopWontStop*" - Pos:18.37%; Neg:47.56%; Neu:34.06%. "*on my do not serve list. Doorman talked 2 fares into a shuttle who we're asking for a cab. Then told me to shut the fuck up*" - Pos: 0.71%; Neg:82.28%; Neu:10.59. "*No microwave AND no fridge? I know this is only the Manor Motor Lodge at but STILL...*" - Pos: 18.63%; Neg:46.46%; Neu:34.90%.
Neutral
"Hey-o happy hour Drinking a 312 Urban Wheat Ale by atwashing :/" - Pos: 28.95%; Neg: 32.6%; Neu: 38.44%. "See you in November :))" - Pos: 30,44%; Neg: 33.09%; Neu: 36,46%. *"Caught some of the set. Awesome* voice. #tingling" - Pos: 38.31%; Neg: 19.79%; Neu:41.88%. "The commercial on MTV with ML's original don playing in the background literally just blew my mind #ilovevegas" - Pos: 28.57%; Neg: 30.41%; Neu: 41.00%.

We make a ratio score calculation as they did in the original work, in which we add one to the total score of that hotel if the tweet mentioning that hotel is classified as positive and subtract one if the tweet is negative, if the tweet is neutral, we neither add nor subtract, it remains the same, as seen in the ratio score formula given by Philander & Zhong [8].

Further experimentation was carried out for more reliable validation. We applied the pre-trained state-of-the-art transformer model developed by Sanh et al. [6] to our problem, following two different approaches: fine-tuning the model with further training on a dataset containing Amazon reviews [49] and their sentiment and fine-tuning it with our own manually classified tweets. This model was created following the template provided by HuggingFace [7] and adapted to the problem by us. The fine-tuning on the Amazon dataset [27] manages a 0.735 accuracy on our classified tweets, and a borderline non-significant ρ of 0.338 (p=0.05) with the TripAdvisor ranking. The fine-tuning on our own tweets reports a 0.808 accuracy and a ρ of 0.447 (p=0.007). As these results show, the accuracy on the validation data is slightly higher in the fine-tuned model than in our own model, but the Spearman correlation with the TripAdvisor ranking is significantly lower. Despite the accuracy being roughly the same, we can observe that the fine-tuned model makes more negative predictions. This greatly influences the score since the positive-

negative ratio is usually greater than 1, and so this model does not do as well as our own in terms of the Spearman coefficient.

For our own model, we studied the option of training the neural network with a different bigger dataset from Datafiniti [48]. This dataset contains around 35000 hotel reviews that we classify as positive, negative, and neutral according to their star rating. We decided to observe the result that this new training obtains to rule out one of the possible areas for improvement, which would be increasing the number of manually classified tweets and having a larger dataset. The results were slightly worse than those of the tweet dataset but still better than the other models and methods, obtaining a validation accuracy of 0.706 on our own tweets and a ρ of 0.569 (p = 0.0004) between the obtained ranking and the TripAdvisor ranking, which a priori rules out the problem of having a small dataset used for training in our approach. Table III shows the comparison between all the results presented thus far.

TABLE III. Validation Accuracy and Spearman Correlation for Each Model

Model	Spearman p	Val accuracy
Philander & Zhong	0.501	
DistilBERT (Amazon)	0.338	0.735
DistilBERT (tweets)	0.447	0.808
Our model (tweets)	0.601	0.803
Our model (Datafiniti)	0.569	0.706

However, we suspect that the improvements achieved by the adjustments made over a single deterministic instance, while relevant to the problem at hand, might not completely reflect the general performance of the model for other cases, since we are optimizing the parameters for only that one case. Thus, for further validation, we make several random executions and compute the average results obtained.

2. Average Results

To get more representative data, we run 25 iterations of each model, under the same conditions. As expected, we obtain somewhat lower results in some models than those previously mentioned using a specific random seed. We obtain a mean of the iterations for making more representative comparisons between models. Table IV shows the average validation accuracy and average Spearman correlation coefficient obtained.

TABLE IV. Average Validation Accuracy and Spearman Correlation for Each Model

Model	Spearman ρ	Val accuracy
Philander & Zhong	0.501	
DistilBERT (Amazon)	0.2855	0.8288
DistilBERT (tweets)	0.4386	0.8288
Our model (tweets)	0.4713	0.7639
Our model (Datafiniti)	0.5763	0.7108

In terms of the Spearman coefficient, the best result is obtained by our model trained with the Datafiniti dataset [48] with a ρ of 0.5763, the next would be our model trained with the tweets dataset [4] with a result of ρ=0.4713, and finally, both models that use DistilBERT [8] either trained with our data [4] or with Amazon reviews [49], for which Spearman results are ρ=0.4385 and ρ=0.2855 respectively.

After observing the data, we can conclude that our approach is superior to the one used in the original work [4], which obtains a ρ of 0.501, and superior to DistilBERT's model [6] which, when trained with our data [4], does not reach the spearman result obtained in the original work [4] (ρ=0. 4385). And lastly, within our model, the one trained with Datafiniti [48] (ρ=0.5763) is superior to the one trained with our tweets [8] (ρ=0.4713), so we can suggest that the results would be greatly benefitted from possessing more training data and that the formal review format could potentially be more accurate for predicting the general opinion on hotels than tweets. A possible reason for this phenomenon might be that formal language is more standardized and carefully written than informal language seen in social networks, which usually includes slang and misspelt words are very frequent, making training harder.

After these experiments, we obtain the average ratio score for each hotel across all executions for obtaining a more representative ranking of the hotels. In Table V we present a view of the TripAdvisor scores, the ratio scores by Philander & Zhong [8] and our ratio scores.

TABLE V. Ratio Score of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	88	7.04	97.4168	48.6633
Bellagio Las Vegas	88	9.6	27.4728	35.9129
M Resort Spa Casino	87	5.96	33.4462	38.7507
Red Rock Las Vegas	85	10.22	80.2095	27.3824
Venetian Las Vegas	85	8.66	229.6565	27.6740
Aria Las Vegas	84	10.72	39.7088	36.5600
Wynn Las Vegas	84	6.79	44.8549	19.3175
South Point Hotel	84	5.88	103.8230	21.4435
The Mirage	84	4.9	75.6752	19.6242
MGM Grand Hotel	81	4.41	210.1098	29.3816
Tropicana Las Vegas	80	11.1	70.2524	22.4831
NYNY Vegas	79	5.08	48.8507	25.1627
Golden Nugget	79	3.13	14.6022	15.7709
The Cosmopolitan	77	5.2	75.4652	29.7664
Mandalay Bay Resort	74	5.78	79.9043	25.5889
Paris Las Vegas	73	9.9	18.0115	23.6796
Treasure Island	72	6.66	25.2909	18.3458
Caesars Palace	72	6.08	34.5140	24.0737
Monte Carlo Resort	70	6.81	29.6825	19.8006
Planet Hollywood	68	3.39	33.5736	16.6929
Bally's Las Vegas	68	2.62	21.1096	8.6210
Stratosphere Hotel	66	6.05	77.9540	22.8341
Palms Casino Resort	66	5.24	240.7661	29.1991
Hard Rock Hotel LV	64	2.64	46.5190	16.5550
Harrah's Las Vegas	63	4.03	14.7001	13.2363
Luxor Hotel & Casino	60	6.1	72.4207	23.3794
Circus Circus	60	5.68	28.3268	17.2063
Excalibur Las Vegas	60	5.53	18.1248	22.3499
Rio Las Vegas	59	4.61	55.3870	16.3660
Flamingo Las Vegas	55	5.42	13.6495	11.7607
Hooters Casino Hotel	55	4.64	16.1221	10.7504
Riviera Las Vegas	48	5.23	34.4220	26.4814
LVH Hotel & Casino	47	5.86	26.7072	20.7079
The Quad Las Vegas	43	2.57	5.2856	9.0395

With the scores in the fourth and fifth columns, we calculate the Spearman correlation coefficient again against the TripAdvisor scores in the second column, obtaining 0.48099 by training with our tweets and 0.60969 by training with the Datafiniti dataset [48] in this case. From these results, we can observe that the extensiveness of the training data affects the model's capability for obtaining relative performances to a great extent. Table VI offers a visualization of the rankings predicted by each model.

TABLE VI. Ranking of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	1	7	5	1
Bellagio Las Vegas	2	5	24	4
M Resort Spa Casino	3	14	21	2
Red Rock Las Vegas	4	3	6	9
Venetian Las Vegas	5	6	2	8
Aria Las Vegas	6	2	17	3
Wynn Las Vegas	9	9	16	23
South Point Hotel	7	15	4	19
The Mirage	8	25	9	22
MGM Grand Hotel	10	28	3	6
Tropicana Las Vegas	11	1	12	17
NYNY Vegas	12	24	14	12
Golden Nugget	13	31	32	29
The Cosmopolitan	14	23	10	5
Mandalay Bay Resort	15	17	7	11
Paris Las Vegas	16	4	29	14
Treasure Island	18	10	26	24
Caesars Palace	17	12	18	13
Monte Carlo Resort	19	8	22	21
Planet Hollywood	20	30	20	26
Bally's Las Vegas	24	33	27	34
Stratosphere Hotel	22	13	8	16
Palms Casino Resort	21	21	1	7
Hard Rock Hotel LV	23	32	15	27
Harrah's Las Vegas	25	29	31	30
Luxor Hotel & Casino	26	11	11	15
Circus Circus	27	18	23	25
Excalibur Las Vegas	28	19	28	18
Rio Las Vegas	29	27	13	28
Flamingo Las Vegas	30	20	33	31
Hooters Casino Hotel	31	26	30	32
Riviera Las Vegas	32	22	19	10
LVH Hotel & Casino	33	16	25	20
The Quad Las Vegas	34	34	34	33

3. Ratio Score Computation Using Probabilities

We propose a new method for calculating the ratio score of hotels, as described in the proposal. Using this method, we repeat the average measurements conducted previously, using again an average of 25 iterations. This method of calculating the ratio score will only be applied to our model as it is the one with the best ratio score results.

We do not take into account the validation accuracy and time since this change only affects the Spearman coefficient and the rest remains unchanged. We prefer to focus on the ratio score of the hotels as it is the measurement we use to compare ourselves with both TripAdvisor and the approach by Philander & Zhong [8], and for practical purposes, it is a very good indicator for a hotel to know its evaluation.

TABLE VII. Comparison of Spearman Coefficients With Different Ratio Score Calculations

System	Spearman ρ
Philander & Zhong [8]	0.5010
Our model (tweets, regular method)	0.4713
Our model (Datafiniti, regular method)	0.5763
Our model (tweets, new method)	0.5311
Our model (Datafiniti, new method)	0.6106

As can be observed in Table VII, our new method of computing the ratio score improves the average results obtained, especially for the tweet dataset. As we did before, we also obtain the average ratio scores by means of this new method for each hotel, obtaining the results shown in Table VIII.

TABLE VIII. New Ratio Score of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	88	7.04	21.3194	6.7282
Bellagio Las Vegas	88	9.6	16.9642	7.2440
M Resort Spa Casino	87	5.96	15.4200	6.5445
Red Rock Las Vegas	85	10.22	22.1321	6.4168
Venetian Las Vegas	85	8.66	22.6369	6.4919
Aria Las Vegas	84	10.72	15.5908	6.9443
South Point Hotel	84	5.88	17.3179	5.3535
The Mirage	84	4.9	19.4905	6.2074
Wynn Las Vegas	84	6.79	14.4221	5.7305
MGM Grand Hotel	81	4.41	17.9841	5.4712
Tropicana Las Vegas	80	11.1	18.5516	7.5564
NYNY Vegas	79	5.08	16.8442	5.8826
Golden Nugget	79	3.13	8.8946	4.9302
The Cosmopolitan	77	5.2	18.5545	6.0684
Mandalay Bay Resort	74	5.78	17.2724	5.8667
Paris Las Vegas	73	9.9	12.3846	5.9012
Caesars Palace	72	6.08	14.5638	6.5781
Treasure Island	72	6.66	13.3477	5.4290
Monte Carlo Resort	70	6.81	13.6617	5.9256
Planet Hollywood	68	3.39	12.8227	5.2006
Palms Casino Resort	66	5.24	22.1213	5.4528
Stratosphere Hotel	66	6.05	19.8981	5.5520
Hard Rock Hotel L	64	2.64	15.4029	4.8764
Bally's Las Vegas	68	2.62	10.3494	3.8094
Harrah's Las Vegas	63	4.03	10.1939	4.7356
Luxor Hotel & Casino	60	6.1	17.3746	5.6681
Circus Circus	60	5.68	14.2205	5.6515
Excalibur Las Vegas	60	5.53	11.1029	5.5503
Rio Las Vegas	59	4,61	13,4598	4,8241
Flamingo Las Vegas	55	5,42	9,7044	4,7637
Hooters Casino Hotel	55	4,64	10,4948	4,1472
Riviera Las Vegas	48	5,23	14,9554	6,4636
LVH Hotel & Casino	47	5,86	15,1780	5,4890
The Quad Las Vegas	43	2,57	4.3803	4,2604

Lastly, we compute the Spearman coefficient with these new ratio scores, obtaining 0.57978 for the tweet dataset and 0.64121 for the Datafiniti dataset [48] . With these results, we can conclude that our model performs much better in terms of accurately predicting the relative positiveness of opinions about hotels. Table IX offers a visualization of the rankings obtained by different models compared to the actual TripAdvisor ranking.

TABLE IX. New Ranking of Hotels by Each Model

	TripAdvisor	Philander & Zhong	Our model (tweets)	Our model (Datafiniti)
Palazzo Las Vegas	1	7	4	4
Bellagio Las Vegas	2	5	13	2
M Resort Spa Casino	3	14	16	6
Red Rock Las Vegas	4	3	2	9
Venetian Las Vegas	5	6	1	7
Aria Las Vegas	6	2	15	3
South Point Hotel	7	15	11	25
The Mirage	8	25	6	10
Wynn Las Vegas	9	9	21	16
MGM Grand Hotel	10	28	9	22
Tropicana Las Vegas	11	1	8	1
NYNY Vegas	12	24	14	14
Golden Nugget	13	31	33	27
The Cosmopolitan	14	23	7	11
Mandalay Bay Resort	15	17	12	15
Paris Las Vegas	16	4	27	13
Caesars Palace	17	12	20	5
Treasure Island	18	10	25	24
Monte Carlo Resort	19	8	23	12
Planet Hollywood	20	30	26	26
Palms Casino Resort	21	21	3	23
Stratosphere Hotel	22	13	5	19
Hard Rock Hotel LV	23	32	17	28
Bally's Las Vegas	24	33	30	34
Harrah's Las Vegas	25	29	31	31
Luxor Hotel & Casino	26	11	10	17
Circus Circus	27	18	22	18
Excalibur Las Vegas	28	19	28	20
Rio Las Vegas	29	27	24	29
Flamingo Las Vegas	30	20	32	30
Hooters Casino Hotel	31	26	29	33
Riviera Las Vegas	32	22	19	8
LVH Hotel & Casino	33	16	18	21
The Quad Las Vegas	34	34	34	32

C. Complexity Results

Table X shows the average execution time of every model. As we can observe, our model obtains again the best results with an average of 19.28 seconds for the model trained with our tweets [8] and an average of 124.94 seconds for our model trained with Datafiniti [48] , while the DistilBERT model [6] takes 2092 seconds for training with our tweets and 2228 seconds for training with the Amazon dataset [49].

TABLE X. Average Run Time

Model	Time (seconds)
DistilBERT (Amazon)	2228
DistilBERT (tweets)	2092
Our model (tweets)	19.28
Our model (datafiniti)	124.94

This gap between the models is due to the fact that the complexity of our proposed model in terms of number of parameters is a lot lower than that of general-purpose pre-trained models. Models like DistilBERT [6] greatly rely on running on CUDA-compatible graphics, which makes training on personal computer CPUs or small servers ridiculously slow even for very small datasets like our case.

As we can see in Table XI, which compares our model with some of the most used models. This comparison measures the number of parameters that each neural network uses to train each model, as we can see our model has only 0.2665 million parameters, compared to the 530000 million of Megatron-Turing NLG [7]. This is because we saw that we do not need such a complex network for our problem. Thanks to this, our execution time is negligible compared to the rest of the models.

TABLE XI. Number of Trainable Parameters in Each Model

Model	Number of parameters (in millions)
GPT [56]	110
GPT-2 [57]	1500
GPT-3 [47]	175000
BERT base [5]	110
BERT large [5]	340
DistilBERT [6]	66
Megatron-Turing NLG [7]	530000
Our model	0.2665

V. Conclusions and Future Work

We have created a simple attention-based neural network model following the Transformer architecture and applied it to the problem of analyzing the sentiment in tweets about hotels.

We can conclude that our model, despite a low number of parameters of 266500, obtains a more accurate ranking score for the hotels than both the approach by Philander & Zhong and a big pre-trained model like DistilBERT, measured by the Spearman correlation coefficient. Both training datasets, tweets and Datafiniti reviews, manage to improve the results obtained by every other option, with the more extensive dataset of Datafiniti reviews achieving superior results that the tweets dataset.

In terms of validation accuracy on our own dataset, however, the DistilBERT model achieves slightly superior performance but is greatly outperformed in terms of training and execution time for processing the same dataset in the same system.

After reviewing the results obtained, we can conclude that for the problem of classifying the positivity of opinions about hotels in tweets, a very specific problem, using a neural network model based on Transformers with few parameters and simple layers is a better alternative than some basic NLP approaches and than complex pre-trained models.

As options for future work, we propose the following areas of research. Using other external datasets or increasing the number of classified tweets, might report more accurate results, since we have found that the tweet dataset performs considerably worse than the bigger review dataset. The model could also be trained with a dataset from Facebook, Instagram, or any other social network.

For the dataset used and our aim of efficiency, we believe to have reached a refined model configuration. However, increasing the model learning capabilities at the cost of greater complexity is bound to report more accurate results, as the results have shown that the DistilBERT model does.

In a similar way, attempting to implement convolutional layers, recurrent layers or other systems for text classification is also an area that could be explored. Attempting to train a different model to include the analysis of linked pictures to support the classification of the text or taking threads into account could prove especially beneficial when

dealing with tweets. For improving on the TripAdvisor comparisons, training with reviews from TripAdvisor itself should report greater ranking accuracy than training with comments from a completely different site.

The problem could also be updated to current times, obtaining the latest tweets tagging the hotel accounts and updating the TripAdvisor scores and ranking to reflect that of the current date. This could be done by means of a Domain Specific Language to automatically extracts tweets and generate a new dataset.

Acknowledgment

This research was funded by Fundación Universidad de Oviedo grant numbers PE.065.21 & PE.066.21.

References

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: 10.1007/BF00344251.

[2] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.

[3] D. W. Otter, J. R. Medina, and J. K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, Feb. 2021, doi: 10.1109/TNNLS.2020.2979670.

[4] A. Vaswani *et al.*, "Attention is All you Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, Oct. 2018.

[6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, Oct. 2019.

[7] S. Smith *et al.*, "Using DeepSpeed and Megatron to Train Megatron-Turing NLG 530B, A Large-Scale Generative Language Model," *arXiv preprint arXiv:2201.11990*, 2022.

[8] K. Philander and Y. Y. Zhong, "Twitter sentiment analysis: Capturing sentiment from integrated resort tweets," *International Journal of Hospitality Management*, vol. 55, pp. 16–24, May 2016, doi: 10.1016/J.IJHM.2016.02.001.

[9] S. Barke, R. Kunkel, N. Polikarpova, E. Meinhardt, E. Baković, and L. Bergen, "Constraint-based Learning of Phonological Processes," *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pp. 6176–6186, 2019, doi: 10.18653/V1/D19-1639.

[10] O. Güngör, T. Güngör, and S. Uskudarli, "EXSEQREG: Explaining sequence-based NLP tasks with regions with a case study using morphological features for named entity recognition," *PLoS One*, vol. 15, no. 12, Dec. 2020, doi: 10.1371/journal.pone.0244179.

[11] E. M. Ponti, A. Korhonen, R. Reichart, and I. Vulić, "Isomorphic transfer of syntactic structures in cross-lingual NLP," *56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 1531–1542, 2018, doi: 10.18653/V1/P18-1142.

[12] C. Hutto, E. G.-P. of the international A. conference on, and undefined 2014, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," *Proceedings of the international AAAI conference on web and social media*, vol. 8, no. 1, pp. 216–225, 2014

[13] P. Chikersal, S. Poria and E. Cambria, "SeNTU: Sentiment Analysis of Tweets by Combining a Rule-based Classifier with Supervised Learning," *Proceedings of the 9th international workshop on semantic evaluation*, pp. 647–651, 2015.

[14] F. Wunderlich and D. Memmert, "Innovative Approaches in Sports Science—Lexicon-Based Sentiment Analysis as a Tool to Analyze Sports-Related Twitter Communication," *Applied Sciences*, vol. 10, no. 2, p. 431, Jan. 2020, doi: 10.3390/APP10020431.

[15] S. Lai, L. Xu, K. Liu and J. Zhao, "Recurrent convolutional neural networks for text classification," *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[16] H. Kim and Y. S. Jeong, "Sentiment Classification Using Convolutional Neural Networks," *Applied Sciences*, vol. 9, no. 11, p. 2347, Jun. 2019, doi: 10.3390/APP9112347.

[17] X. Fang and J. Zhan, "Sentiment analysis using product review data," *Journal of Big Data*, vol. 2, no. 1, pp. 1–14, Dec. 2015, doi: 10.1186/S40537-015-0015-2/FIGURES/9.

[18] M. Imran, P. Mitra, and C. Castillo, "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages," *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pp. 1638–1643, May 2016, doi: 10.48550/arxiv.1605.05894.

[19] X. Liu, H. Shin, and A. C. Burns, "Examining the impact of luxury brand's social media marketing on customer engagement: Using big data analytics and natural language processing," *Journal of Business Research*, vol. 125, pp. 815–826, Mar. 2021, doi: 10.1016/J.JBUSRES.2019.04.042.

[20] F. Z. Xing, E. Cambria, and R. E. Welsch, "Natural language based financial forecasting: a survey," *Artificial Intelligence Review*, vol. 50, no. 1, pp. 49–73, Oct. 2017, doi: 10.1007/S10462-017-9588-9.

[21] M. G. Huddar, S. S. Sannakki, and V. S. Rajpurohit, "Attention-based multi-modal sentiment analysis and emotion detection in conversation using RNN," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 112–121, 2021, doi: 10.9781/ijimai.2020.07.004.

[22] P. Dcunha, "Aspect Based Sentiment Analysis and Feedback Ratings using Natural Language Processing on European Hotels," *Doctoral thesis. Dublin, National College of Ireland*, 2019.

[23] T. Ghorpade and L. Ragha, "Featured based sentiment classification for hotel reviews using NLP and Bayesian classification," *Proceedings - 2012 International Conference on Communication, Information and Computing Technology*, 2012, doi: 10.1109/ICCICT.2012.6398136.

[24] B.-Ş. Posedaru, T.-M. Georgescu, and F.-V. Pantelimon, "Natural Learning Processing based on Machine Learning Model for automatic analysis of Online Reviews related to Hotels and Resorts," *Database Systems Journal*, vol. 11, no. 1, pp. 86–105, 2020.

[25] W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey," *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, Dec. 2014, doi: 10.1016/J.ASEJ.2014.04.011.

[26] L. C. Yu, J. L. Wu, P. C. Chang, and H. S. Chu, "Using a contextual entropy model to expand emotion words and their intensity for the sentiment classification of stock market news," *Knowledge-Based Systems*, vol. 41, pp. 89–97, Mar. 2013, doi: 10.1016/J.KNOSYS.2013.01.001.

[27] M. Hagenau, M. Liebmann, and D. Neumann, "Automated news reading: Stock price prediction based on financial news using context-capturing features," *Decision Support Systems*, vol. 55, no. 3, pp. 685–697, Jun. 2013, doi: 10.1016/J.DSS.2013.02.006.

[28] I. Maks and P. Vossen, "A lexicon model for deep sentiment analysis and opinion mining applications," *Decision Support System*, vol. 53, no. 4, pp. 680–688, Nov. 2012, doi: 10.1016/J.DSS.2012.05.025.

[29] J. Wang *et al.*, "Systematic Evaluation of Research Progress on Natural Language Processing in Medicine Over the Past 20 Years: Bibliometric Study on PubMed," *Journal of Medical Internet Research*, vol. 22, no. 1, Jan. 2020, doi: 10.2196/16816.

[30] A. Alsudais, G. Leroy, and A. Corso, "We know where you are tweeting from: Assigning a type of place to tweets using natural language processing and random forests," *Proceedings - 2014 IEEE International Congress on Big Data*, pp. 594–600, Sep. 2014, doi: 10.1109/BIGDATA.CONGRESS.2014.91.

[31] Y. Goldberg and M. E. Ben, "splitSVM: Fast, Space-Efficient, non-Heuristic, Polynomial Kernel Computation for NLP Applications," *Association for Computational Linguistics*, pp. 237–240, Jun. 2008.

[32] C. Bartz, T. Herold, H. Yang, and C. Meinel, "Language Identification Using Deep Convolutional Recurrent Neural Networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10639, pp. 880–889, 2017, doi: 10.1007/978-3-319-70136-3_93.

[33] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," *2010 IEEE International Symposium on Circuits and*

Systems: Nano-Bio Circuit Fabrics and Systems, pp. 253–256, doi: 10.1109/ISCAS.2010.5537907.

[34] A. Conneau, H. Schwenk, Y. le Cun, and L. Lo¨ıc Barrault, "Very Deep Convolutional Networks for Text Classification," *Nature*, pp. 1–11, Jun. 2016, doi: 10.48550/arxiv.1606.01781.

[35] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.

[36] T. Wang, P. Chen, K. Amaral, and J. Qiang, "An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification," *arXiv preprint arXiv:1609.03663*, Sep. 2016.

[37] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," *8th Workshop on Syntax, Semantics and Structure in Statistical Translation*, pp. 103–111, Sep. 2014, doi: 10.3115/v1/w14-4012.

[38] Z. Shaheen, G. Wohlgenannt, and E. Filtz, "Large Scale Legal Text Classification Using Transformer Models," *Computer Science ArXiV*, vol. abs/2010.12871, 2020.

[39] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," *3rd International Conference on Learning Representations*, Sep. 2015, doi: 10.48550/arxiv.1409.0473.

[40] T. Shao, Y. Guo, H. Chen, and Z. Hao, "Transformer-Based Neural Network for Answer Selection in Question Answering," *IEEE Access*, vol. 7, pp. 26146–26156, 2019, doi: 10.1109/ACCESS.2019.2900753.

[41] U. Khandelwal, K. Clark, D. Jurafsky, and Ł. Kaiser, "Sample Efficient Text Summarization Using a Single Pre-Trained Transformer," *arXiv preprint arXiv:1905.08836*, 2019.

[42] T. Wang, X. Wan, and H. Jin, "Amr-to-text generation with graph transformer," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 19–33, Jan. 2020, doi: 10.1162/TACL_A_00297/43537/AMR-TO-TEXT-GENERATION-WITH-GRAPH-TRANSFORMER.

[43] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 2, pp. 107–116, Apr. 1998, doi: 10.1142/S0218488598000094.

[44] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," *Proceedings of the Annual Conference of the International Speech Communication Association*, vol. 2020-October, pp. 5036–5040, 2020, doi: 10.21437/Interspeech.2020-3015.

[45] A. Rives *et al.*, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 118, no. 15, Apr. 2021, doi: 10.1073/PNAS.2016239118/SUPPL_FILE/PNAS.2016239118.SAPP.PDF.

[46] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A Survey of Transformers," *OpenAI*, Jun. 2021.

[47] T. Brown *et al.*, "Language Models are Few-Shot Learners," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 1877–1901.

[48] "Hotel Reviews - dataset by datafiniti | data.world." https://data.world/datafiniti/hotel-reviews (accessed Feb. 11, 2022).

[49] "amazon_reviews_multi · Datasets at Hugging Face." https://huggingface.co/datasets/amazon_reviews_multi (accessed Feb. 11, 2022).

[50] "Hotel Reviews - dataset by datafiniti." https://data.world/datafiniti/hotel-reviews (accessed Mar. 23, 2022).

[51] A. Joshi, S. Kale, S. Chandel, and D. K. Pal, "Likert Scale: Explored and Explained," *British Journal of Applied Science & Technology*, vol. 7, no. 4, p. 396, 2015, doi: 10.9734/BJAST/2015/14975.

[52] François Chollet, "Keras: the Python deep learning API," *Astrophysics Source Code Library*, 2018.

[53] A. Belhadi, Y. Djenouri, J. C. W. Lin, and A. Cano, "A data-driven approach for twitter hashtag recommendation," *IEEE Access*, vol. 8, pp. 79182–79191, 2020, doi: 10.1109/ACCESS.2020.2990799.

[54] K. Philander and Y. Y. Zhong, "Twitter sentiment analysis: Capturing sentiment from integrated resort tweets," *International Journal of Hospitality Management*, vol. 55, pp. 16–24, May 2016, doi: 10.1016/J.IJHM.2016.02.001.

[55] "Natural Language Processing with Transformers [Book]." https://www.oreilly.com/library/view/natural-language-processing/9781098103231/ (accessed Feb. 11, 2022).

[56] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI*, 2018.

[57] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," *OpenAI*, vol. 1, no. 8, p. 9, 2019.

Sergio Arroni

Sergio Arroni is a student of Software Engineering at the University of Oviedo, passionate about Machine Learning, Artificial Intelligence and NLP among other Software fields. He is currently working at the Foundation of the University of Oviedo, researching new advances in the field of Artificial Intelligence and Machine Learning.

Yeray Galán

Yeray Galán is a Software Engineering Graduate of the University of Oviedo and a Master's Degree in Mathematical Research and Modelling, Statistics and Computing student at the University of the Basque Country and the University of Zaragoza, currently doing research at Fundación Universidad de Oviedo on artificial intelligence, particularly machine learning, and Monte Carlo methods.

Xiomarah Guzmán-Guzmán

Xiomarah Guzmán-Guzmán is an Interim Professor and Ph.D. candidate in the Department of Computer Science at the University of Oviedo (Spain). She has a Master's in Website Management and Engineering from the International University of La Rioja, and B.S. in Computer Science from the Technological University of Santiago. She has published some articles in international journals and conferences. Her research interests include artificial intelligence, recommendation systems and machine learning.

Edward Rolando Núñez-Valdez

Edward Rolando Núñez-Valdez is an associate professor in the Department of Computer Science at the University of Oviedo (Spain). He has a Ph.D. in Computer Engineering from the University of Oviedo, a Master's in Software Engineering from the Pontifical University of Salamanca and a B.S. in Computer Science from the Autonomous University of Santo Domingo. He has participated in several research projects; he has taught computer science at various schools and universities, and he has worked in software development companies and IT consulting for many years. He has published several articles in international journals and conferences. His research interests include artificial intelligence, recommendation systems, decision support systems, health informatics, modeling software with DSL and MDE.

Alberto Gómez

Alberto Gómez works for the Department of Business Administration, at the School of Industrial Engineering of The University of Oviedo, Spain. His teaching and research initiatives focus on the areas of Production Management, Applied Artificial Intelligence and Information Systems. He has written several national and international papers. Journal of the Operational Research Society. Artificial Intelligence for Engineering Design, Analysis and Manufacturing. International Journal of Foundations of Computer Science. European Journal of Operational Research. International Journal of production economics. Engineering Applications of Artificial Intelligence. Concurrent Engineering- Research and Applications.