Procesamiento del Lenguaje Natural, Revista nº 67, septiembre de 2021, pp. 13-25 recibido 07-04-2021 revisado 04-06-2021 aceptado 07-06-2021

Sarcasm Detection with BERT

Detección de Sarcasmo con BERT

Elsa Scola, Isabel Segura-Bedmar
Universidad Carlos III de Madrid, Leganés, Spain
scolaelsa@gmail.com, isegura@inf.uc3m.es

Abstract: Sarcasm is often used to humorously criticize something or hurt someone's feelings. Humans often have difficulty in recognizing sarcastic comments since we say the opposite of what we really mean. Thus, automatic sarcasm detection in textual data is one of the most challenging tasks in Natural Language Processing (NLP). It has also become a relevant research area due to its importance in the improvement of sentiment analysis. In this work, we explore several deep learning models such as Bidirectional Long Short-Term Memory (BiLSTM) and Bidirectional Encoder Representations from Transformers (BERT) to address the task of sarcasm detection. While most research has been conducted using social media data, we evaluate our models using a news headlines dataset. To the best of our knowledge, this is the first study that applies BERT to detect sarcasm in texts that do not come from social media. Experiment results show that the BERT-based approach overcomes the state-of-the-art on this type of dataset.
Keywords: Sarcasm Detection, Deep Learning, BiLSTM, BERT.

Resumen: El sarcasmo se usa con frecuencia para realizar crítica o burla indirecta, a veces hiriendo los sentimientos de alguien. Algunas veces, las personas tienen dificultades para reconocer los comentarios sarcásticos, ya que decimos lo contrario de lo que realmente queremos decir. Por lo tanto, la detección automática de sarcasmo en textos es una de las tareas más complicadas en el Procesamiento del Lenguaje Natural (PLN). Además, se ha convertido en un área de investigación relevante debido a su importancia para mejorar el análisis de sentimientos. En este trabajo, exploramos varios modelos de aprendizaje profundo, como Bidirectional Long Short-Term Memory (BiLSTM) y Bidirectional Encoder Representations from Transformers (BERT) para abordar la tarea de detección de sarcasmo. Si bien la mayoría de los trabajos anteriores se han centrado en datasets construidos con textos de redes sociales, en este artículo, evaluamos nuestros modelos utilizando un dataset formado por titulares de noticias. Por tanto, este es el primer estudio que aplica BERT para detectar el sarcasmo en textos que no provienen de las redes sociales. Los resultados de los experimentos muestran que el enfoque basado en BERT supera el estado del arte en este tipo de conjunto de datos.
Palabras clave: Sarcasm Detection, Deep Learning, BiLSTM, BERT.

1 Introduction

The Cambridge Dictionary defines sarcasm as *"the use of remarks that mean the opposite of what they say, made to hurt someone's feelings or to criticize something in a humorous way"*. However, understanding sarcasm is a task that is often hard for humans, as it is highly dependent on the context and sense of humor of each person(Capelli, Nakagawa, and Madden, 1990). The perception of sarcasm can vary by multiple factors, like culture, gender or personality (Rockwell and Theriot, 2001). For example, Indians and Americans perceive sarcasm in different ways (Joshi et al., 2016). In the following sentence taken from the study presented by Joshi et al. (2016): *"Love going to work and being sent home after two hours"*, Indian annotators do not agree with Americans. Indian annotators labeled the instance as non-sarcastic as they

ISSN 1135-5948. DOI 10.26342/2021-67-1

did not have any context about long commuting to work and that 'being sent home' could mean being fired from a job.

Moreover, when sarcasm happens verbally, aspects like volume, voice tonality and speed, contribute to express it. Often sarcasm is also accompanied by various gestures, like eye and hand movement. In contrast, written sarcasm, which occurs in different environments (such as emails, social media, or product reviews) completely lacks the aforementioned features that contribute to the identification of sarcasm, and therefore, making it more difficult to detect it. This suggests that detecting sarcasm is a very challenging task for humans, and it is even harder for algorithms.

Automatic sarcasm detection is one of the most challenging tasks in Natural Language Processing (NLP) (Eke et al., 2020) and can be used in a variety of applications, ranging from knowing the customers opinions about products or services offered by a company or even identifying inappropriate or harming comments in social media to protect users.

To date, most attempts at sarcasm detection have used Twitter datasets to train and evaluate their models (Ptáček, Habernal, and Hong, 2014). However, these datasets are noisy and add difficulty to the task because tweets are short texts (280 characters). They also contain very informal language, grammatical and spelling mistakes, slang terms, abbreviations and non-standard language features such as hashtags, emoticons, hyperlinks and other ones, which do not occur in standard texts. Moreover, the lack of context could also be a problem as many tweets are replies to previous tweets (Hernandez Farias, Patti, and Rosso, 2016). However, significantly less effort has been put into exploiting other types of texts to train and evaluate models for sarcasm detection.

This study aims to explore different deep learning techniques such as Bidirectional Long Short Term Memory (BiLSTM) (Hochreiter and Schmidhuber, 1997) and Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) to address the task of sarcasm detection from texts. BERT is a model that is gaining increasing popularity due its outstanding performance for multiple NLP tasks (Lee et al., 2020; Zheng and Yang, 2019; Hakala and Pyysalo, 2019). To the best of our knowledge, this is the first study that applies BERT for sarcasm detection in texts that are not extracted from social media.

This paper is organised as follows: Section 2 discusses the main datasets used for sarcasm detection. It also presents state-of-the-art methods for this task. Section 3 describes the dataset and methods used in this work. In Section 4, we show the evaluation of the proposed methods and discuss their results. Finally, Section 5 describes conclusions and future work.

2 Related work

Sarcasm is a form of expression in which people convey the opposite of what they say to hurt someone emotionally or humorously criticize something. This implicit subjectivity to the problem makes it even harder for machines to detect. Therefore, this is one of the most challenging tasks in NLP nowadays. The task of automatic sarcasm detection has been most commonly defined, in past work, as a classification task. That is, given a piece of text, the goal is to predict whether it is sarcastic or not. In this section, we review the main datasets as well as the most recent approaches to address this task.

2.1 Datasets for sarcasm detection

Many datasets for sarcasm detection are created by using hashtag based supervision. Hashtag based supervision consists in searching tweets containing hashtags like #sarcasm, and assuming they were correctly "annotated"by their authors. The main advantage of this technique is that it allows collecting large datasets, which are automatically labeled with no manual annotation. However, the produced labels can be highly noisy. For instance, it considers all tweets without predefined tags as non-sarcastic, however, some of them could express sarcasm.

One of the earliest Twitter dataset for sarcasm detection was proposed by Riloff et al. (2013). This dataset contains a total of 3,000 tweets, of which 2,307 are non-sarcastic and 693 sarcastic. Due to Twitter's data sharing policy, only the tweet ids are permitted to share, so their tweets can be directly downloaded from Twitter by using those ids. However, many of the original tweets have been removed since 2013, and therefore, the dataset is a bit outdated.

Another Twitter dataset collected by

using hashtag based supervision is the *Irony detection in English tweets* dataset, which was part of the SemEval-2018 competition (Apidianaki et al., 2018). Although it is oriented to the task of irony detection, it can be used for sarcasm detection as well, as sarcasm often contains irony in it. The dataset contains 2,396 positive instances (sarcastic tweets) and 604 negative instances (non-sarcastic tweets). Despite the tweets being collected making use of hashtags, all the tweets were manually labeled in order to avoid noisy data. Furthermore, the corpus was cleaned by removing retweets, duplicated tweets, and non-English tweets.

Ghosh and Veale (2016) created one of the biggest datasets for sarcasm detection. The training dataset contains 39,000 tweets, of which 18,000 are sarcastic and 21,000 non-sarcastic, making this dataset evenly balanced. The test dataset contains 2,000 tweets annotated by an internal team of researchers, which is also balanced.

An innovative contribution was made by Oprea and Magdy (2020). This proposal shows an original way of collecting sarcastic tweets. They have designed an online survey where they ask Twitter users to provide links to one sarcastic and three non-sarcastic tweets. This results in the iSarcasm dataset, which contains 4,484 tweets, out of which 777 were labeled as sarcastic and 3,707 as non-sarcastic.

Apart from Twitter data, there are also some available datasets from Reddit, a discussion website. One of them is the dataset collected by Khodak, Saunshi, and Vodrahalli (2018), which contains 1.3 million comments from Reddit. It was generated by scraping comments that contained the
s tag. This tag is often used by Redditors to indicate that their comment is sarcastic and should not be taken seriously. Therefore, it may produce noise, as happen with hashtag based supervision. This dataset provides balanced and imbalanced versions.

Contributions in a dialogue context for sarcasm detection have also been made. The Discussion Forum dataset (Ghosh, Richard Fabbri, and Muresan, 2017) is a collection of posts from forums. For each post, its replies are also included. These posts were manually annotated in three categories of sarcasm: general sarcasm, hyperbole, and rhetorical questions. For the general sarcasm category, there are 3,260 posts per class (sarcastic and not-sarcastic), that is, a total of 6,520 posts. The hyperbole contains 582 posts per class and rhetorical questions 851 posts per class.

Several options can be found in the multilingual panorama for sarcasm detection. Ptáček, Habernal, and Hong (2014) created two datasets for sarcasm detection on Twitter in English and Czech. While the English dataset was obtained by hashtag based supervision (using the hashtag #sarcasm as an indicator of sarcastic tweets), the Czech dataset was manually annotated. The English dataset is provided in two options: balanced corpus (50,000 sarcastic and 50,000 non-sarcastic tweets), and imbalanced corpus (25,000 sarcastic and 75,000 non-sarcastic tweets). The Czech dataset has 325 sarcastic tweets and 6,675 non-sarcastic ones.

Recently, the IroSvA (Irony Detection in Spanish Variants) shared task (Ortega-Bueno et al., 2019) provided a dataset for irony detection in short messages (tweets and news comments) written in Spanish. The corpus consists of 9,000 short messages about different topics written in Spanish –3,000 from Cuba, 3,000 from Mexico and 3,000 from Spain- and annotated with irony. Approximately, 80 % of the corpus corresponds to the training dataset, whereas the remaining 20 % corresponds to the test set.

As can be seen from the above, most datasets for sarcams detection are collected from social media. An alternative dataset was presented by Misra and Arora (2019). They proposed a novel dataset based on new headlines to overcome the limitations of Twitter and other social media datasets. Sarcastic headlines were collected from TheOnion, [1] which is a news website whose sole purpose is to produce sarcastic content. Non-sarcastic headlines were extracted from the HuffPost,[2] which is a real news website. This dataset is described in detail in Subsection 3.1.

2.2 Approaches for sarcasm detection

We now review the main approaches that have addressed this task.

[1] https://www.theonion.com
[2] https://www.huffpost.com

2.2.1 Traditional Machine Learning

Early approaches for sarcasm detection used traditional machine learning algorithms. Liebrecht, Kunneman, and van den Bosch (2013) proposed a model for text classification, which is based on Balanced Winnow (Littlestone, 1988). This machine learning technique produces interpretable per-class weight. These weights can later be used to discover the highest-ranking features for one class. To train the classifier, the authors used a collection of tweets collected from a database provided by the Netherlands eScience Centre. To create the dataset, the authors collected a sample of tweets tagged with the hashtag #sarcasme (the Dutch word for sarcasm) and a random sample of tweets without it. They explored the effect of balanced (50-50) and imbalanced (25 % sarcastic and 75 % non-sarcastic) data. The classifier provided 75 % TPR (True Positive Rate or recall) and 16 % FPR (False Positive Rate) using the balanced dataset. However, although the use of an imbalanced dataset had a positive effect on FPR (5 %), TPR dropped markedly to 56 %. Error analysis showed that sarcasm is often indicated by the usage of intensifiers and exclamations. When these are not present in the tweet, there is often a hashtag indicating sarcasm. The authors hypothesized that explicit markers, like hashtags, are the digital equivalent of nonverbal expressions that people use in real life to express sarcasm. One of the limitations of this study is that the tweets were automatically annotated without further manual review.

The same year, Riloff et al. (2013) proposed an alternative approach to define and identify sarcasm in text. They state that sarcasm is often defined in terms of contrast or "saying the opposite of what you mean". As they stated it in their study, *"[It] is common on Twitter: the expression of positive sentiment (e.g., "love" or "enjoy") in reference to a negative activity or state (e.g., "taking an exam" or "being ignored"*. Their approach focused on trying to identify this type of contrast in the text by recognizing positive sentiments with negative situations in sentences. To achieve this, they created a novel bootstrapping algorithm that automatically learns lists of positive sentiment phrases and negative situation phrases from sarcastic tweets. They used the bootstrapped lexicons to recognize sarcasm by looking for phrases in their tweets. This system achieved an F1 of 22 %. Additionally, they tested a Support Vector Machine (SVM) (Cortes and Vapnik, 1995) classifier providing an F1 of 48 %. Finally, they combined both approaches in an attempt to improve the results. The hybrid approach obtained an improvement in recall with a slight drop in precision, which resulted in an F1 of 51 %.

In 2014, Ptáček, Habernal, and Hong (2014) proposed two classifiers: Maximum Entropy (MaxEnt) (Nigam, 1999) and SVM to address this task. This research gave more importance to feature engineering rather than to the classifiers. A Bag-of-Words approach was applied to represent the texts. A number of experiments were performed by combining previously selected n-grams and a set of language-independent features, including punctuation marks, emoticons, quotes, capitalized words, character n-grams, and skip-grams as baselines. Moreover, they did a multilingual study by using an English dataset, as well as a Czech dataset. These datasets were described above. They evaluated balanced and imbalanced datasets scenarios. For the English dataset, The MaxEnt classifier achieved an F1 of 94.7 % and 92.4 % on the balanced and imbalanced datasets respectively. The SVM classifier yielded an F1 of 91.4 % on the balanced data and 88.6 % on the imbalanced data. Experiments showed lower results for the Czech dataset. This may be due to the Czech dataset being much smaller than the English dataset, as well as to the inner grammatical complexity of the Czech language. MaxEnt obtained an F1 of 57 %, while SVM gave the best F1 (58.2 %) on the Czech dataset.

Bamman and Smith (2015) approached the problem from an original perspective by attempting to introduce one of the most relevant components to sarcasm understanding: context. To achieve this, they used Twitter data combined with extra-linguistic information from the context of the tweet. This information includes properties of the author (such as author historical salient terms, author historical topics, author historical sentiment, profile information, profile unigrams), the audience (author features of the users involved in the Twitter conversation), and the immediate communicative environment (such as unigrams and bigrams in both original and response tweets). As a clas-

sifier, a binary Logistic Regression with l_2 regularization and tenfold cross-validation was used. They evaluated different combinations of feature sets (about the tweet, author and environment), showing improvements in comparison to only using term frequencies to represent tweets. Including all the features together yielded the highest accuracy at 85.1 %. However, the most significant improvement in accuracy came from the inclusion of author features.

2.2.2 Deep Learning Approaches

We now present some of the latest work on sarcasm detection research, in which mostly deep learning techniques are used.

Amir et al. (2016) proposed to automatically learn and exploit user embeddings combined with lexical features to detect sarcasm. User embeddings are vector representations that *"encode latent aspects of users and capture homophily, by projecting similar users into nearby regions of the embedding space."* (Amir et al., 2016). More concretely, their approach captures relations between users and their content. One of the main advantages of this work is that it avoids the laborious feature engineering process. The authors only used the text of previous posts of the users to create the user embeddings. The authors used a Convolutional Neural Network (CNN), obtaining an accuracy of 87 %. Thus, user embeddings can capture relevant user attributes without the need for elaborated feature engineering.

Ghosh, Richard Fabbri, and Muresan (2017) exploited the conversation context in sarcasm detection. The authors used the Discussion Forum data (Oraby et al., 2016b), which contains sarcastic responses from a forum and their corresponding context (the original post and its replies). They also collected sarcastic and non-sarcastic tweets to create a dataset for their experiments. Several types of LSTM networks were investigated, showing an F1 of 70.56 % for the Discussion Forum data and an F1 of 73.45 % on the Twitter dataset.

The most recent studies for sarcasm detection have exploited BERT, a novel language model based on transformers to provide deep contextual representations for words. This model is gaining increasing popularity due its outstanding performance for multiple NLP tasks.

Xu and Xu (2019) presented an investigation of different models to explore the effect of contextual information on sarcasm detection. Concretely, various LSTM models and BERT were used. The implemented LSTM models were all of them unidirectional (from left to right) and used pre-trained GloVe(Pennington, Socher, and Manning, 2014) as a word embedding model. Two datasets are used for this study, Discussion Forum data (Oraby et al., 2016a)), and Reddit Sarcasm data (Khodak, Saunshi, and Vodrahalli, 2018)), which were described above. The results of this study show that BERT achieved better results than all the LSTM models for both datasets. In the case of the LSTM model, the performance varies depending on the dataset. For example, it obtains an accuracy of 73.23 % for the Discussion Forum dataset, but 67.32 % for Reddit data. This is probably due to the lack of "quality.of the Reddit dataset as it relies on self-annotated labels, which often add noise. Besides, the fact that the Discussion Forum dataset contains generally properly written English contributes to helping the model perform better. Whereas Reddit is full of typos and slang terms, which makes it hard for word embeddings to understand.

Khatri and P (2020) proposed using machine learning classifiers in combination with BERT and GloVe embeddings in order to detect sarcasm in tweets. The authors experimented with different classifiers: SVM, Logistic Regression, Gaussian Naive Bayes, and Random Forest. They used a balanced dataset containing 5,000 tweets. Their experiments showed that word embeddings are useful for sarcasm detection as they capture the meaning of the words as vector representations. The best results were obtained with Logistic Regression for both embeddings, achieving an F-score of 63 % when BERT embeddings are used, and an F1 of 69 % with Glove embeddings.

Misra and Arora (2019) proposed hybrid neural network architecture for sarcasm detection from texts, as well as a dataset of news headlines, described above, which changes the tendency of the almost exclusive usage of Twitter datasets for the task of sarcasm detection. This system combined pretrained user embeddings and a BiLSTM module, which used an attention module to update the weights of the encoded context for each epoch. As the headlines are written by

professionals in a formal tone, there are no spelling mistakes or slang terms. Moreover, the labels of the sarcastic instances are high quality (TheOnion only publishes sarcastic news). Furthermore, as news headlines are self-contained (there are not reply posts), it is easier to spot the sarcastic elements of the sentence. This hybrid architecture obtained an accuracy of 89.7 %.

3 Approaches

This section describes the dataset used in this work and the two deep learning approaches proposed to deal with the task of sarcasm detection.

3.1 Dataset

Sarcasm detection studies often make use of Twitter datasets (Cai, Cai, and Wan, 2019), collected using keyword hashtags (like #sarcasm). However, these datasets can turn out to be noisy due to the informal use of language in social media. Social media texts contain very informal language, grammatical and spelling mistakes, slang terms, abbreviations (for example, 'TBH' refers to 'To Be Honest') and non-standard language features such as hashtags, emoticons, hyperlinks and other ones that do not occur in standard texts. Furthermore, tweets are often replies to previous tweets, which would imply a lack of contextual information.

In order to avoid these drawbacks of tweets, in this study, the *News Headlines Dataset For Sarcasm Detection* (Misra and Arora, 2019) is used. This dataset contains news headlines collected from two journal websites: *The Onion* and *HuffPost*. The former produces sarcastic versions of current events, whereas the latter is a well known trustworthy newspaper. Some of the advantages of using this dataset:

- News headlines are written without any spelling mistakes and informal usage of the vocabulary.
- Pre-trained word embedding models (Mikolov et al., 2013; Pennington, Socher, and Manning, 2014; Joulin et al., 2017) trained using formal texts usually provide higher coverage of vocabulary than pre-trained models trained on tweets.
- The headlines are self-contained, whereas, tweets, on the contrary, could be responses to previous tweets, or part of threads, which would translate in a lack of context.
- Since the sole purpose of *The Onion* is to publish sarcastic news, it could provide a higher guarantee of the correctness of the data in comparison to some Twitter datasets based in keyword hashtags. In studies like (Riloff et al., 2013), it is assumed that human labeling from Twitter users is correct, however, the annotation of a sample showed that only 85 % of those tweets were indeed sarcastic. Therefore, the Twitter datasets can be noisy. Thus, headlines from newspapers like *The Onion* can be an alternative that contributes to more accurately annotated datasets for sarcasm detection. However, the implicit subjectivity in humor related tasks certainly hinders the achievement of the "perfect" dataset.

The original dataset is provided in JSON format. Each headline is represented by its text, the link to the original news article, and an value of 0 (if it is a non-sacarstic headline) or 1 (if it is a sarcastic headline). After removing duplicate headlines, the dataset is composed of 28,503 headlines, of which 14,951 are non-sarcastic and 13,552 are sarcastic. Finally, the dataset was split into 70 % for training (with 9,498 sarcastic headlines and 10,454 non-sarcastic ones), 10 % for validation (with 1,342 sarcastic headlines and 1,508 non-sarcastic ones) and 20 % for test (with 2,712 sarcastic headlines and 2,989 non-sarcastic ones). As can be seen, the three datasets are balanced, that is, the amount of positive instances (sarcastic texts) and negative ones (non-sarcastic texts) is roughly the same. These datasets were used to train the models and evaluate their performance.

3.2 Methods

3.2.1 Long Short-Term Memory (LSTM)

As a baseline, we propose the Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) architecture that recently has been successfully used for text classification (Zhou et al., 2016; Wang et al., 2018). LSTM is a unidirectional model that processes the inputs from left to right, but not from right to left. Hence, during the training, it can only preserve relevant information from the

left part of the input, but it does not know about what is the information on the right part. However, sometimes, to correctly understand a text, we need to take into account not just the previous words, but also the coming words. For example, *"Sometimes I need what only you can provide: your absence."*, and *"Sometimes I need what only you can provide: your love."*, are sentences that share the same beginning, but having completely different meanings.

To overcome this drawback, a Bidireational LSTM (BiLSTM) network is proposed. This model connects two hidden layers of opposite directions to the same output. In this way, the output layer can get information from the past (forward) and future (backward) states simultaneously. Therefore, BiLSTM can capture past (left) and future (right) contexts information. In our experimentation, we apply a BiLSTM layer with 128 units in each direction. The number of units was set according to previous literature work, such as (Garain, 2019; Garain and Mahata, 2019).

The network is initialized with word embeddings. To do this, the headlines are tokenized and each token is represented as a vector by using a pre-trained word embedding model such as Glove (Pennington, Socher, and Manning, 2014), developed by Google . In particular, we use glove.6B.200d, which was trained with the Wikipedia 2014 + Gigaword 5 corpora and contains 6B tokens. The dimension of word vectors is 200.

In deep learning models, it is important not to take the last result of each cell, but rather the best result of it. For this reason, after the BiLSTM layer, a global maxpooling layer downsamples the entire feature map to a single value. This is done by checking each sequence of results provided by each LSTM cell and retaining only the maximum result. This allows us to identify the strongest trait of a headline and highlight the tokens with the most relevant information. For example, it could identify a word that is particularly funny in the headline, which would be helpful for sarcasm detection.

After the global maxpooling layer, we add two fully connected layers, the first one with 40 units and a dropout probability of 0.5, and the second one with 20 units and a dropout probability of 0.5. The addition of fully connected layers in deep learning models has shown to improve the performance of the text classification task (Kim, 2014). ReLU, a nonlinear activation function capable to capture complex relationships, is used as the activation function. As it is sparsely activated (it provides zero for all negative inputs, and thereby, units often do not activate at all), it's more likely that neurons are actually processing meaningful aspects of the problem.

For the output layer, one single unit with a sigmoid function has been used that allows us to obtain the probability of an instance (text) being sarcasm. Therefore, one single probability is returned. For $p > 0{,}5$, it would be considered that the instance (text) belongs to the positive class (sarcastic), whereas if $p < 0{,}5$ then it would be considered as the negative class (non-sarcastic).

3.2.2 Bidirectional Encoder Representations from Transformers (BERT)

BERT has been repeatedly showing state-of-the-art results in a wide range of tasks (Lee et al., 2020; Zheng and Yang, 2019; Hakala and Pyysalo, 2019), however, it has hardly been used for sarcasm detection (Khatri and P, 2020). Thus, one of the main contributions of our study is the use of BERT (Devlin et al., 2019) to address the task of sarcasm detection from news headlines.

BERT relies on a transformer to learn the contextual relationships between the words in a text. The purpose of BERT is to generate a language representation model. Therefore, an encoder is needed in which an input tokenizer is used. For the implementation, the official tokenization script provided by BERT was used, which is progressively being updated with the latest improvements.

After the encoding process, the tokens, as well as the masks and segments are obtained. Each of these will correspond to an input layer of the network. There are different versions of the BERT model (Devlin et al., 2019): BERT-Base and BERT-Large. The last one is an improved and computationally more intensive version of the first model. This model has the following parameteres: L=24, H=1024, A=16, where L is the number of stacked encoders, H is the hidden size and A is the number of heads in the MultiHead Attention layers. Therefore, we use the BERT-Large model (*bert_en_uncased_L24_H1024_A-16*), which was pre-trained for English on Wikipedia and Books Corpus. Inputs are "un-

cased", which means the text is converted to lower-case before the WordPiece tokenization (e.g., 'John Doe' becomes 'john doe'). Additionally, all the accent markers are stripped. For the training process, random input masking is applied independently to word pieces, as described in (Devlin et al., 2019). The tokens, as well as the masks and segments obtained after the encoding process, correspond to the inputs of the BERT layer.

The output of the BERT layer is then processed by the *tf_op_layer_strided_slice* layer, which performs the extraction of a straded slice of a tensor. Then, a Sigmoid layer with one single unit receives the output of this layer and obtains a probability of an instance being sarcasm.

It often occurs in the field of machine learning, that an algorithm performs incredibly well on the training dataset, but poorly on the test set. This common phenomenon is called overfitting. That is, the model has a high variance, which makes it difficult to generalize well on new data. To avoid overfitting, we apply different strategies such as dropout and early stopping. Dropout (Srivastava et al., 2014) is the standard regularizer for deep neural networks in NLP. This regularization technique involves setting a probability of keeping certain nodes or not. A value between 0 and 1 is specified, which is the fraction of the input units to drop. It has been shown, that a dropout rate of 0.5 is effective in most scenarios (Kim, 2014). Therefore, there is a probability of 50 % that a node will be removed from the network. This, ultimately, results in a much simpler network that helps to prevent overfitting. Early stopping was also used to prevent the model from overfitting. Early stopping is a method in which an arbitrarily large number of training epochs are specified, and the training process is stopped once the model performance stops improving on the validation dataset. Therefore, loss in the validation dataset was monitored. A patience of 3 was used, which means that the network is allowed to continue training for up to an additional 3 epochs, after the point that validation loss stopped improving. This allows us to get across flat spots or find some additional improvement during the training process. Then, the last best model is the one that is stored for posterior predictions.

We used the well-known API written in Python, Keras (2.3.1), for building and training deep learning models. Keras runs on top of the machine learning platform TensorFlow. We also use a TensorFlow 2 (TF2) SavedModel, which is the recommended way to share pre-trained models and model pieces on TensorFlow Hub. These models can be integrated with Keras by making use of TensorFlow's high-level API.

The chosen optimizer is Adam, which is an adaptive learning rate optimizer introduced by Kingma and Ba (2015). The authors proposed the optimizer as "a method for efficient stochastic optimization that only requires first-order gradients with little memory requirement. The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients". For training the Bi-LSTM model, we used the default parameters in Keras for Adam.However, for our BERT model, we use the default parameters, except for the learning rate, whose value was modified to 2e-6. The selected loss for both models is binary cross entropy, which is the standard cross-entropy loss for binary classification tasks.

For the LSTM model, the number of epochs is 25 (early stopping at the 5th) and the batch size is 100. For the BERT model, The number of epochs is 10 (early stopping at the 5th) and the batch size is 20.

As the environment to train and test the models, Google Colab was used with GPU activated. Google Colab is a Google Research product that enables running Python code on the browser for free with computational resources, such as GPU. The dataset as well as the code to replicate the experiments can be found in the GitHub repository.[3]

4 Results and Discussion

Table 1 shows the results obtained with the BiLSTM and BERT models for sarcasm detection, displaying different metrics. This also includes results of an SVM classifier and a CNN model(Kim, 2014). They have been considered as the baseline systems, as they have been widely and successfully used in sarcasm detection (Khatri and P, 2020; Ptáček, Habernal, and Hong, 2014; Amir et al., 2016). As in the BiLSTM model, the CNN model was also initialized with pre-trained word

[3]https://github.com/ElsaScola/Sarcasm-Detection-with-Natural-Language-Processing-and-Deep-Learning

embeddings from Glove(Pennington, Socher, and Manning, 2014) and used the adam optimizer for the training. The convolutional layer has 128 filters of size 5. After this layer, a maxpooling layer is added to select the most important features.

All the deep learning models provide better results than the baseline system based on SVM. CNN and BiLSTM resulted in very similar results, being BiLSTM slightly better. It can be seen that BERT performed in general better than the BiLSTM model. In particular, BERT provides an improvement of 4.65 % in F1 score over LSTM. More specifically, BERT has surpassed the LSTM model in both precision (6.83 % improvement) and recall (2.51 % improvement), which indicates the effectiveness of this approach for sarcasm detection. Despite using BiLSTM to capture better the context of the sentence, the attention mechanism of BERT surpasses it in this task. To the best of our knowledge, this is the first study that applies BERT to detect sarcasm in texts that are not social media messages.

Moreover, we compare our results to those presented in (Misra and Arora, 2019) since both studies use the same dataset. Our BERT model shows an improvement of $\sim$ 1,7 % in the results, compared to the hybrid network architecture proposed by (Misra and Arora, 2019). It also outperforms the systems described in Section 2, although our results are not comparable to what have been reported in systems which focused on social media texts. BERT was also used in (Xu and Xu, 2019; Khatri and P, 2020), but their results were much worse than those obtained by our BERT model. Like BERT, our CNN and BiLSTM models also have significant better performance than those previous deep learning systems trained and tested on social media texts (Amir et al., 2016; Ghosh, Richard Fabbri, and Muresan, 2017; Xu and Xu, 2019; Khatri and P, 2020). This agrees with the fact of social media texts are characterized by a lack of context, which leads to high ambiguity and makes the task of detecting sarcasm even more difficult.

We have studied a small sample of false positives and negatives produced by the models to identify their major weak points in which these models fail. Table 2 shows some headlines that were wrongly classified as sarcastic by both models. Having a look into these false positives, it can be hypothesized that both models seem to consider those instances containing humorous or very surrealistic sentences as sarcasm. For example, both models agree that the sentence *"Man apparently opens beer with butt, inspires bartenders everywhere"* is sarcastic. This could be caused by the absurdity of the sentence, which might result in the models considering it as a joke. The same situation happens for sentences like *"Farting teen sparks fight."*, which can be interpreted as jokes. This is due to the lack of knowledge of the model on the context. Paying attention to another sentence, which both models considered sarcastic: *"Passport robot tells man of Asian descent his eyes are too closed."*, it can be seen that the models might learn to see the humor in situations that are not necessarily humorous or that might oppress certain collectives, as the Asian community in this case. This is an issue that goes beyond the scope of this study, however, it is interesting to observe and analyze this kind of phenomenon. The data that is given to a model to learn humor can eventually lead the model to reproduce the same racist stereotypes that we see in society. Thus, special care should be put in curating the training data and reach a consensus of what type of humor is funny and which is harmful or offensive.

In the current study, BERT was able to classify correctly various instances that BiLSTM could not. While BiLSTM failed to classify the headline *"Obama is like that really great neighbour who's moving out."* as a sarcastic one, BERT was able to recognise that there was humor in the sentence but was not meant to be sarcastic. That differentiation is key in order to obtain more accurate results in the task. On the other hand, the headline *"Jailed for being too poor"* was correctly classified as non-sarcastic by BiLSTM, but wrongly as sarcastic by BERT.

We now review some of the False Negatives. For example, for the headline *"Angelina Jolie coming for your baby"*, both models agree that this is not sarcastic, even if it is indeed sarcastic. This is probably the lack of context handling in the two models on who Angelina Jolie is and what is known for. The same situation happens with the sarcastic headline *"Police repeatedly shoot Tim Cook after mistaking Iphone for gun"*, which was wrongly classified as non-sarcastic by both

Model	Loss	Acc.	P	R	F1
SVM		0.79	0.81	0.75	0.78
CNN	0.4434	0.8654	0.8547	0.8639	0.8593
LSTM	0.4391	0.8680	0.8561	0.8687	0.8623
BERT	2.9443	0.9147	0.9244	0.8938	0.9088

Tabla 1: Results over the test dataset.

Trump suggests Iran brought deadly terrorist attacks upon itself.
Farting teen sparks fight.
Man apparently opens beer with butt, inspires bartenders everywhere.
Passport robot tells man of Asian descent his eyes are too closed.
Dog gives priceless reaction when owner pretends to faint.

Tabla 2: Examples of false positives for BiLSTM and BERT models.

models. It is needed to know that Tim Cook is the CEO of Apple to understand the sentence. It could be helpful in identifying important subjects in sentences and getting some context from them, in order to help the models make more accurate predictions. While BiLSTM was able to detect sarcasm in the headline *"Jeff Bezos named Amazon employee of the month"*, BERT classified it as non-sarcastic. Again, this may be due to the lack of context in the model.

There are some headlines particularly hard to identify as they could be real facts, independently of the context the model is given. For example, the headline *"Visit to Google Earth reveals house is on fire"*, even if the headline was given to a human that is aware of what Google Earth is and had the context to understand the headline, the person could think is a real sentence as it is a possibility, and therefore, the context by itself, would not play a big role for this type of sentences.

If the models were able to know the "absolute truth" around the proposed headlines, they could easily classify them as sarcastic/non-sarcastic. However, providing context to the model is a challenging task for which some approaches were proposed in social network data (see Section 2). Nevertheless, this is a challenging task, which we plan to address in future work.

5 Conclusions

To the best of our knowledge, this is the first study that applies BERT to detect sarcasm in texts that are not social media texts. Our experiments show that BERT achieves better performance than BiLSTM. Our BERT-based approach also overcomes the hybrid neural architecture (based on Bi-LSTM) described in (Misra and Arora, 2019), which was also evaluated using the news headlines dataset.

As future work, we plan to extend the evaluation with other sarcasm datasets to measure the results of our models on different types of texts. We also plan to study how to encode knowledge of the world in our deep learning models, which will help us to obtain a correct interpretation of any text.

Furthermore, we will explore the results of multimodal sarcasm detection, by accompanying texts with audio to also contribute to sarcasm recognition in the evolving field of virtual assistants. Intonation in the speech of the user could be indicative of sarcasm. This type of research is still in the early stages, however, a few datasets have been presented in this direction (Castro et al., 2019).

Acknowledgments

This work has been supported by the Madrid Government (Comunidad de Madrid) under the Multiannual Agreement with UC3M in the line of "Fostering Young Doctors Research"(NLP4RARE-CM-UC3M), as well as in the line of "Excellence of University Professors"(EPUC3M17), and in the context of the V PRICIT (Regional Programme of Research and Technological Innovation).

Bibliografía

Amir, S., B. C. Wallace, H. Lyu, P. Carvalho, and M. J. Silva. 2016. Modelling con-

text with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany.

Apidianaki, M., S. M. Mohammad, J. May, E. Shutova, S. Bethard, and M. Carpuat, editors. 2018. *Proceedings of The 12th International Workshop on Semantic Evaluation*, New Orleans, Louisiana.

Bamman, D. and N. A. Smith. 2015. Contextualized sarcasm detection on twitter. In *Proceedins of the 9TH International AAAI Conference On Web And Social Media*, Oxford, UK.

Cai, Y., H. Cai, and X. Wan. 2019. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515, Florence, Italy.

Capelli, C. A., N. Nakagawa, and C. M. Madden. 1990. How children understand sarcasm: The role of context and intonation. *Child Development*, 61(6):1824–1841.

Castro, S., D. Hazarika, V. Pérez-Rosas, R. Zimmermann, R. Mihalcea, and S. Poria. 2019. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy.

Cortes, C. and V. Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186, Minneapolis, USA.

Eke, C. I., A. A. Norman, L. Shuib, and H. F. Nweke. 2020. Sarcasm identification in textual data: systematic review, research challenges and open directions. *Artificial Intelligence Review*, 53(6):4215–4258.

Garain, A. 2019. Humor analysis based on human annotation(haha)-2019: Humor analysis at tweet level using deep learning. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF-SEPLN 2019*, volume 2421, pages 191–196, Bilbao, Spain.

Garain, A. and S. K. Mahata. 2019. Sentiment analysis at SEPLN (TASS)-2019: sentiment analysis at tweet level using deep learning. In *Proceedings of the Iberian Languages Evaluation Forum co-located with 35th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2019*, volume 2421, pages 611–617, Bilbao, Spain.

Ghosh, A. and T. Veale. 2016. Fracking sarcasm using neural network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 161–169, San Diego, California.

Ghosh, D., A. Richard Fabbri, and S. Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 186–196, Saarbrücken, Germany.

Hakala, K. and S. Pyysalo. 2019. Biomedical named entity recognition with multilingual BERT. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 56–61, Hong Kong, China.

Hernandez Farias, D., V. Patti, and P. Rosso. 2016. Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology*, 16:1–24.

Hochreiter, S. and J. Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9:1735–80, 12.

Joshi, A., P. Bhattacharyya, M. Carman, J. Saraswati, and R. Shukla. 2016. How do cultural differences impact the quality of sarcasm annotation?: A case study of Indian annotators and American text. In *Proceedings of the 10th SIGHUM Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 95–99, Berlin, Germany.

Joulin, A., E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. 2017. Fasttext. zip: Compressing text classification models. In *5th International Conference on Learning Representations (ICLR)*.

Khatri, A. and P. P. 2020. Sarcasm detection in tweets with bert and glove embeddings. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 56–60.

Khodak, M., N. Saunshi, and K. Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Kingma, D. P. and J. Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, San Diego, CA, USA.

Lee, J., W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Liebrecht, C., F. Kunneman, and A. van den Bosch. 2013. The perfect solution for detecting sarcasm in tweets #not. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 29–37, Atlanta, Georgia.

Littlestone, N. 1988. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318.

Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Misra, R. and P. Arora. 2019. Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*.

Nigam, K. 1999. Using maximum entropy for text classification. In *Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67.

Oprea, S. and W. Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online.

Oraby, S., V. Harrison, L. Reed, E. Hernandez, E. Riloff, and M. Walker. 2016a. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles, USA.

Oraby, S., V. Harrison, L. Reed, E. Hernandez, E. Riloff, and M. A. Walker. 2016b. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, 13-15 September 2016, Los Angeles, CA, USA*, pages 31–41. The Association for Computer Linguistics.

Ortega-Bueno, R., F. Rangel, D. Hernández Farıas, P. Rosso, M. Montes-y Gómez, and J. E. Medina Pagola. 2019. Overview of the task on irony detection in spanish variants. In *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*, Bilbao, Spain.

Pennington, J., R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar.

Ptáček, T., I. Habernal, and J. Hong. 2014. Sarcasm detection on Czech and English Twitter. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 213–223, Dublin, Ireland.

Riloff, E., A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang. 2013. Sarcasm as contrast between a positive sen-

timent and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 704–714, Seattle, Washington.

Rockwell, P. and E. M. Theriot. 2001. Culture, gender, and gender mix in encoders of sarcasm: A self-assessment analysis. *Communication Research Reports*, 18(1):44–52.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 06.

Wang, J.-H., T.-W. Liu, X. Luo, and L. Wang. 2018. An lstm approach to short text sentiment classification with word embeddings. In *Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018)*, pages 214–223, Hsinchu, Taiwan.

Xu, L. and V. Xu. 2019. Project report: Sarcasm detection. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/project.html. Online; accessed 8 July 2021.

Zheng, S. and M. Yang. 2019. A new method of improving bert for text classification. In *Proceedings of International Conference on Intelligent Science and Big Data Engineering*, pages 442–452, Nanjing, China.

Zhou, P., Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu. 2016. Text classification improved by integrating bidirectional LSTM with two-dimensional max pooling. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3485–3495, Osaka, Japan.