Humor detection in short English phrases using NLP

Leyla Rocío Becerra Barajas
Department of Industrial and Computer Engineering
Universidad Nacional de Colombia
Bogotá. Colombia.
Email: lrbecerrab@unal.edu.co

Joan Gabriel Bofill Barrera
Department of Industrial and Computer Engineering
Universidad Nacional de Colombia
Bogotá. Colombia.
Email: jgbofillb@unal.edu.co

Camilo Alfonso Mosquera Benavides
Department of Industrial and Computer Engineering
Universidad Nacional de Colombia
Bogotá. Colombia.
Email: camosquerab@unal.edu.co

***Abstract*—The present document describes a project of a Subject Data Science in the masters degree of computer Science Engineering. In this paper, a research about humor detection is carried out, the methodological steps are described and a case of study and a dataset are presented. The objective is to develop an strategy for humor detection and assess its potential in a database of short texts in English Language. For this purpose, several machine learning techniques and methods from the literature like transformers, decision trees and neural networks are considered and some will be applied in the actual data. A discussion about the advantages and problems and possible improvements of our strategy for humor detection will be developed throughout the semester and the principal conclusions and found difficulties will be pointed out in further submits of this document.**

1. Introduction

Humor is a daily concept for a lot of people, it provokes laughter and happiness, and yet it remains a complex matter, some researchers found that humor leads to enhance the human's health and mood [1]. Humor is an illustrative example of how humans use creative language devices in social communication. Humor not only serves to interchange information or share implicit meaning, but also engages a relationship between those exposed to the funny message. Besides, humor is also a distinctive quality of humans, which is considered a figurative language; where the person can express his feelings through social media once entertainment media have existed or if he faces a funny event. It is important to mention that there is no objective consensus of what humor is and what defines a funny person, thing or even sentence. Nevertheless, the fact of analysing the aspects of the language that evokes humor is essential for understanding the conditions and thoughts of people. Automatic humor detection in texts has interesting use cases in building human-centered artificial intelligence systems such as chatbots and virtual assistants. There are a lot of possible applications of this kind of development, for instance to identify whether an input text should be taken seriously or not, which is a critical step to understand real motive of users' queries, return appropriate answers, and enhance the overall experience of user with the system. There are also interesting applications can be addressed in a short and middle time future like emulates humor and sarcasm in IA devices.

1.1. Main goal

Further discover and learning in Natural language processing techniques and creating a methodology for applying machine learning models capable of detecting whether a sentence is humorous or not and apply it to our chosen dataset.

1.2. Scope

We want to take into account the methodologies in the literature for detecting humor and then be able to run and test a model with good properties. The chosen/designed model will be trained with short texts in English language and tested in a fraction (*test set*) of the dataset. We expect to be able to apply the trained model to other data sets and see how its performance varies by setting different conditions and compared with different criteria.

2. Literature review

2.1. Methodology

This topic was planned, conducted and documented using the Systematic Literature Review (SLR) process, that was proposed by [2]. The SLR includes some stages: research questions, search process, inclusion and exclusion criteria, quality assessment, data collection and data analysis.

In the next paragraphs each stage will be explained:

2.1.1. Research questions. In this stage the research questions that addressed this project were defined:

1) What are the main characteristics in texts that let support the humor detection?

2) Which natural language processing techniques have been used to detect humor in texts?
3) Which case study of humor detection would use as reference in this project?

2.1.2. Search process. The answers to the research questions must be investigated in the scientific literature through a search process. In order to focus the search, in this stage is necessary to define a search equation and specific bibliographic sources.

Search equation: The terms "*natural language processing*", "*humor*", "*feelings*", "*detection*", "*computation*" were selected to build the equation. In order to limit the search was selected the language *English* and the publication year *after 2015*. So the search equation was defined as:

(TITLE-ABS-KEY (natural AND language AND processing AND (humor OR feelings) AND (detection OR computation)) AND LANGUAGE (English)) AND PUBYEAR ¿ 2015

Source of information: The primary searches where based upon the *Scopus-Elsevier*, the *IEEE Xplore* and the *Google Scholar* bibliographic resources.

2.1.3. Inclusion and exclusion criteria. Using the last search equation in the selected databases resulted in 110 candidates for relevant papers. Considering that many papers do not provide useful information to address the research questions, it's paramount to select the most convenient papers. The following inclusion and exclusion criteria which refined the selection was implemented:

Inclusion criteria:

- The publication must have a complete literature review or detailed explanation about techniques of humor detection or a comparative study with tables/figures.
- The most complete publication will be included in case of similar or redundant studies.

Exclusion criteria:

- Informal papers will be excluded.

After this state, there are 33 studies selected.

2.1.4. Quality assessment. The strategy for quality assessment is leading by the following questions:

1) The literature review is consistent and is based in a good bibliography?
2) If the document has more than one year: Has been cited?

2.1.5. Data collection. Finally, the papers were collected and selected according to the before aspects and in this stage there are 16 studies selected.

2.1.6. Data analysis. Some figures are elaborated about the data that were collected using SLR. In the first term, in the Fig. 1 from *Scopus-Elsevier* a progressive increase in scientific interest is shown from the initial year of the search to the present.

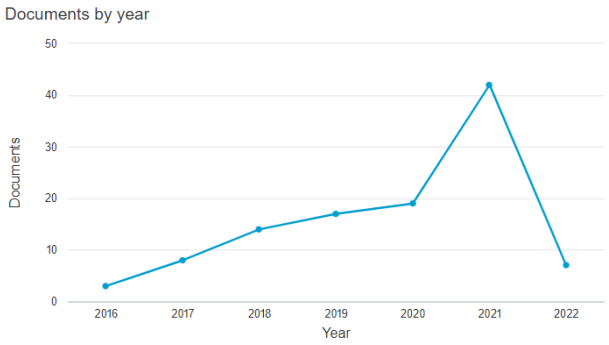

Figure 1. Scientific interest: Papers by year

A keyword analysis performed using VOSViewer shows in the Figure 2 draw the co-occurrence of words in the documents found, It's possible identify two clusters (in different color: red and green), the first one is close to natural language, semantic and ontology, and the second one, more recent that refers to artificial intelligence: machine learning and deep learning.

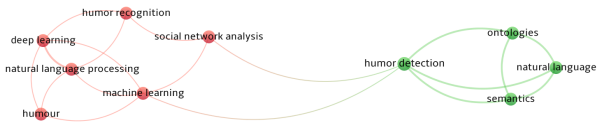

Figure 2. Co-occurrences of keywords in papers

2.2. Basic concepts

Humor is a essential element of all verbal communications and it may be described as "a cognitive experience to provoke laughter and provide amusement" [3]. Understanding humorous language is a easy task for humans, however, classification of humor is a challenge in the real life, because the sense of humor varies from person to person.

Classification of Humor detection is a task in which we distinguish between humorous and non-humorous instances. However it's important to note that the researchers have identified many different theories of humour and also, many different types of humour such as [4] :

- **Sarcasm**, that may be described as an ironic or ambivalent way to convey mockery.
- **Pun**, humour is created using wordplay and ambiguity.

- **Irony**, is a rhetoric literary artifact that expresses the opposite in meaning of what is conveyed via language to create humour.
- **Exaggeration**, It is a way of expressing something in proportions ridiculously exceeding the actual range of that entity affecting amusement .

Researching in humor detection is an important task to improve the human- machine interaction. One of the main researches on it was developed by a branch of artificial Intelligence known as *Natural Language Processing* (NLP): *"is an area of research and application that explores how computers can be used to understand and manipulate natural language text or speech to do useful things*" [5].

2.3. Background

One of the first humor detection models was proposed by [6], they used support vector machines [1] to separate one-liners from other types of texts . Other implementation was developed by [7], one of the best characteristics was the building of a dataset using laughter markers from TED talks as cues to determine which parts lead up to a punchline, and which do not, and then trained multimodal neural netwoks improving the humor detection accuracy. Actually, one of most impressive models in terms of out-performance are the BERT-like (which stands for Bidirectional Encoder Representations from Transformers) models proposed by [9] that had an accuracy close to 98 percent.

3. Case of Study

Having consulted diverse sources, we find a challenge associated to a dataset about humor detection that we will describe in detail in the next section. The quest can be found in the following site: https://paperswithcode.com/sota/humor-detection-on-200k-short-texts-for-humor-1 . Muiltiple papers propose different methods to solve this problem. So one of the approaches taken in the literature is our case of study of the present document [9], the winner of that challenge, or at least the paper whose model's score metric (F1)[2] was higher (over 0.982). In employs an approach uses BERT model [8] to encode text into a few sentence embeddings which enter into an eight-layered neural network. On the other hand, the final layers of the network combine the output of all previous lines of hidden layers in order to predict the final output. In theory, these final layers should determine the congruity of sentences and detect the transformation of reader's viewpoint after reading the punchline. There are similar papers that use BERT model to humor detection task as we can see in [10] but in this case Reddit Data is used.

1. are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories by Vladimir Vapnik with colleagues in the 90's.

2. The F1 score is a machine learning metric that can be used in classification models, it can be defined as the harmonic mean of precision and recall.

4. Data Set Collection

Our principal dataset for this work is the same that is proposed in the aforementioned case of Study paper and it is available on: https://www.kaggle.com/datasets/deepcontractor/200k-short-texts-for-humor-detection. It is a single file called "dataset.csv" of about 15MB, it has two columns: "text" and "humor". The former one is where the phrases potentially containing humour are, so its type is "string", on the contrary the latter is a boolean Column (where the possible values are "True" or "False"), indicating for each row if the sentence present in the **text** variable is humorous. In total there are 200.000 rows without repeated values. A mayor detail is that exactly 50% of the entries have True and the other half have False. A figure with an example of some entries of the dataset will be presented next:

Text	Humor
Why do native americans hate it when it rains in April? because it brings mayflowers	True
Obama's climate change legacy is impressive imperfect and vulnerable	False
My family tree is a cactus we're all pricks	True
Donald trump has found something mysterious for rudy giuliani to do	False

TABLE 1. EXAMPLE ENTRIES OF THE DATASET

We intend to use a similar BERT model and test it on this dataset and probably it will exhibit reasonably good performance. Besides, one could consider also other Machine Learning architectures or mixed methods as in [11]. In addition, we could try our model in another dataset with similar characteristics, such as the ones available here: https://github.com/orionw/RedditHumorDetection and determine how much our model can generalize its results to a completely new data.

5. Data Set Organization

5.1. Relational database model

The data has been modeled using a Relational Database Model as is shown in the figure 3 as already mentioned in the description of the data, the main dataset is a file with two columns: Text and Mood, which were proposed as the "Phrase" entity with Phrase and Label attributes respectively. Other entities such as Repository, Source and File are proposed to store metadata.

5.1.1. Relational database implementation. The deployment was done in AWS using the RDS service. It was created on MySQL,the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

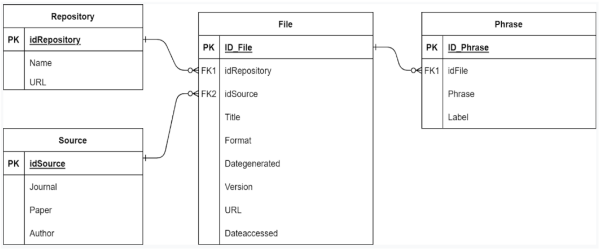

Figure 3. SQL model

The SQL part of "MySQL" stands for "Structured Query Language". SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly, embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

The deployment has the features shown in Figure 4:

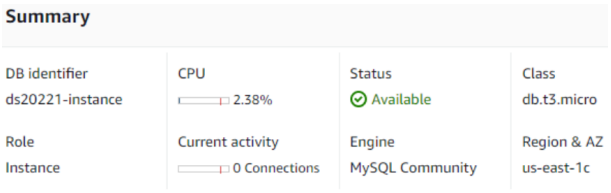

Figure 4. database relational implementation features

The connectivity and security parameters in the implementation are shown in Figure 5

Figure 5. database connectivity features

5.2. NonRelational database model

This model store the information using collections in specific structure as is shown in the next Figure 6.

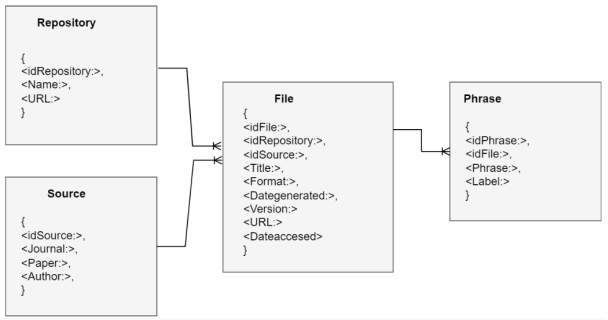

Figure 6. Non Relational model

And the document structure looks like is shown in the Figure 7

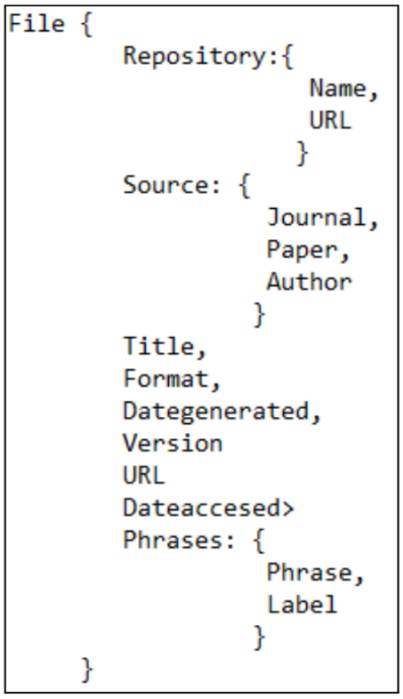

Figure 7. NoSQL structure

5.2.1. Non Relational database implementation. NoSQL implementation: The deployment was done in MongoDB. MongoDB is a schema-free, document-oriented database written in C++. The choice of encoded format in MongoDB is JSON. This means that even if the data is nested inside JSON documents, it will still be queryable and indexable. As a document store based, it stores values (referred to as documents) in the form of encoded data MongoDB has a flexible storage system, which means stored objects are not necessarily required to have the same structure or fields. MongoDB also has some optimization features, which distributes the data collections across, being overall a more balanced and performance focused system.

5.3. comparison of the relational and NoSQL

Given the nature of our problem, we do not have a huge dataset that cannot be accessed efficiently in a relational way, actually it is not a Big Data problem nowadays. We have 200 thousand entries and our feature is really the text that is in a single column named “humor”. Additionally, in terms of computational complexity our target is simple, being our target a boolean value that says if the given phrase in “humor” is funny or not. The complexity of our problem lies in the language analysis that could understand such a complex matter in the human relationship as humor is. On the other hand, considering that sql databases are more mainstream and easier to understand, the project can benefit from that fact in order to have better data management to whatever process may appear in the development of the project.

6. Data Cleaning

In order to have a good dataframe for the analysis involved in this project, several cleaning procedures must take place. First, missing values must be treated in some way (deleted or imputed in some way). The database was loaded in colab as a pandas dataframe, and then the isnull().sum() method allow to detect the possible missing values, as one can observe in the Figure 8:

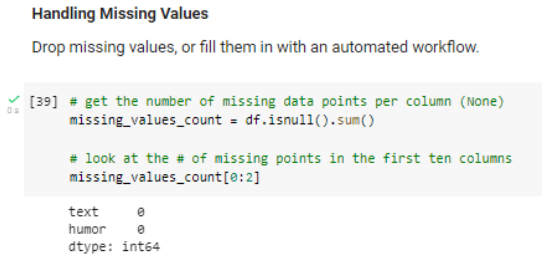

Figure 8. Missing values in data

After checking that there are no missing values, other important aspect to take into account is the duplicated values. It only applies to the ”text” column, considering that the ”humor” column is boolean and for more than two items, duplicates will always appear. To verify duplicates, the method .duplicated() also from pandas’ library was used, as in the Figure 9:

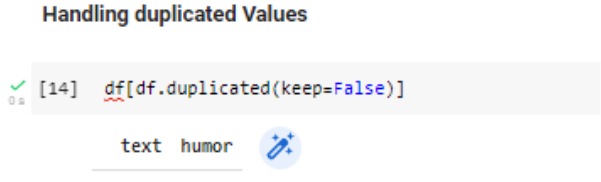

Figure 9. Duplicated values in data

In consequence, there are no duplicates in the dataset.

Additionally, the boolean column ”humor” was explored to verify that there is no value different to ”True” or ”False”. It can be evident given its boolean nature, but sometimes typos or some other error in the database management and creation. So, in order to do this, the method value_counts() from pandas was employed, see Figure 10:

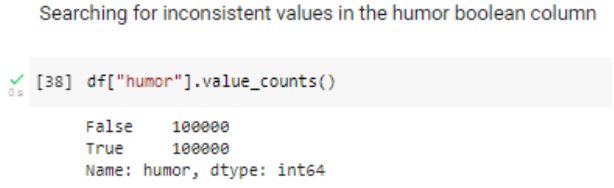

Figure 10. Values that appear in ”humor” column

In conclusion, as it was stated in the kaggle page, the database is perfectly balanced and there are no strange values that could provoke some sort of problem in the future analyses.

Next step, to facilitate the analysis and optimize the performance of Natural Language Processing methods applied to the dataset, further cleaning of the text is suggests. So, a new column similar to ”text” is created in which special characters such as exclamation and punctuation signs, symbols and so on will be omitted. It is important to mention that Capital letters are transformed as well into lowercase. To do this task, the following function is developed (see Figure 11).

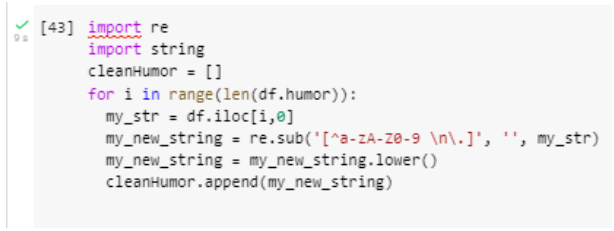

Figure 11. Function employed for text cleaning

And, as an example of its results can be seen in the figure 12. For further testing the original column was conserved so multiple experiments could be accomplished.Thus, the dataset now has three columns.

Further transformation of the dataset will not be included in the cleaning section as it depends strongly of the techniques and models to predict the ”humor” target. The script containing the procedures showed in the previous images is available in the github repository of the project that is rteferenced at the end of this paper.

	textCleaned	text	humor
0	joe biden rules out 2020 bid guys im not running	Joe biden rules out 2020 bid: 'guys, i'm not r...	False
1	watch darvish gave hitter whiplash with slow p...	Watch: darvish gave hitter whiplash with slow ...	False
2	what do you call a turtle without its shell dead.	What do you call a turtle without its shell? d...	True
3	5 reasons the 2016 election feels so personal	5 reasons the 2016 election feels so personal	False
4	pasco police shot mexican migrant from behind ...	Pasco police shot mexican migrant from behind,...	False
...	...	...	...
199995	conor maynard seamlessly fits oldschool rb hit...	Conor maynard seamlessly fits old-school r&b h...	False
199996	how to you make holy water you boil the hell o...	How to you make holy water? you boil the hell ...	True
199997	how many optometrists does it take to screw in...	How many optometrists does it take to screw in...	True
199998	mcdonalds will officially kick off allday brea...	Mcdonald's will officially kick off all-day br...	False
199999	an irish man walks on the street and ignores a...	An irish man walks on the street and ignores a...	True

200000 rows × 3 columns

Figure 12. Dataset after cleaning function

7. Exploratory Data Analysis (EDA)

Exploratory data analysis is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods [18].

Before modeling and predicting, data should first be explored to uncover the patterns and structures that exist. Exploratory data analysis involves both numerical and visual techniques designed to reveal interesting information that may be hidden in the data.

The particular graphical techniques employed in EDA are often quite simple, consisting of various techniques of:

- Plotting the raw data (such as data traces, histograms, bihistograms, probability plots, lag plots, block plots, and Youden plots).
- Plotting simple statistics such as mean plots, standard deviation plots, box plots, and main effects plots of the raw data.
- Positioning such plots so as to maximize our natural pattern-recognition abilities, such as using multiple plots per page.

According with the selected dataset previously described, we have gotten the following data and visualisations:

7.1. Dataset

A simple review of the dataset identifying columns, format, number of records, will allow us to become familiar with the data. So, It's necessary display the main characteristicas of raw data (see Figure 13):

7.2. Dataset head

A very simple way to identify the format and columns in a dataset is display its header in the Figure 14:

```
3 dsfile
```

	text	humor
0	Joe biden rules out 2020 bid: 'guys, i'm not r...	False
1	Watch: darvish gave hitter whiplash with slow ...	False
2	What do you call a turtle without its shell? d...	True
3	5 reasons the 2016 election feels so personal	False
4	Pasco police shot mexican migrant from behind,...	False
...	...	...
199995	Conor maynard seamlessly fits old-school r&b h...	False
199996	How to you make holy water? you boil the hell ...	True
199997	How many optometrists does it take to screw in...	True
199998	Mcdonald's will officially kick off all-day br...	False
199999	An irish man walks on the street and ignores a...	True

200000 rows × 2 columns

Figure 13. Humor Dataset: dataset.csv

```
1 dsfile.head()
```

	text	humor
0	Joe biden rules out 2020 bid: 'guys, i'm not r...	False
1	Watch: darvish gave hitter whiplash with slow ...	False
2	What do you call a turtle without its shell? d...	True
3	5 reasons the 2016 election feels so personal	False
4	Pasco police shot mexican migrant from behind,...	False

Figure 14. Humor Dataset: head

7.3. Dataset describe

The text statistics will help to explore the main characteristics of the text data. So We can see them using the describe command in a notebook (see Figure 15):

```
1 dsfile.describe()
```

	text	humor
count	200000	200000
unique	200000	2
top	Joe biden rules out 2020 bid: 'guys, i'm not r...	False
freq	1	100000

Figure 15. Humor Dataset: describe

7.4. Histograms

The text statistics visualisation are great tools in order to understand the data in a simple view. Mostly we use

histograms like:

7.4.1. Characters in each sentence histogram. This can give us a rough idea about the phrases length. The Characters in each sentence histogram for the dataset is shown in Figure 16

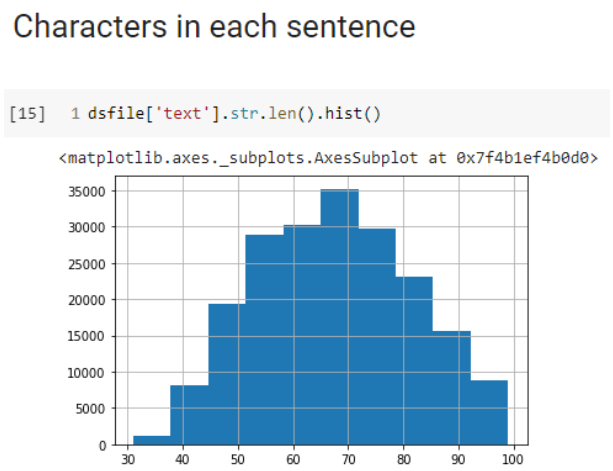

Figure 16. Humor Dataset: Characters in each sentence

7.4.2. Words in each sentence histogram. On to data exploration at a word-level this histogram will plot the number of words appearing in each humor phrase. See the Figure 17

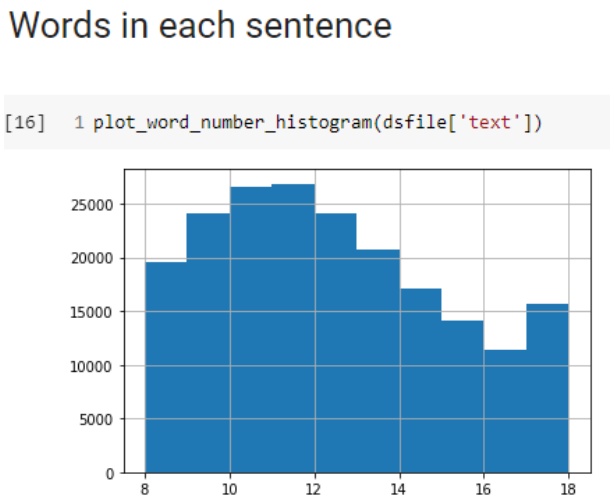

Figure 17. Humor Dataset: Words in each sentence

7.4.3. Stopwords histogram. Stopwords are the words that are most commonly used in any language such as "the"," a"," an" etc. As these words are probably small in length these words may have caused the above graph to be left-skewed. Analyzing the amount and the types of stopwords can give us some good insights into the data. See the Figure 18.

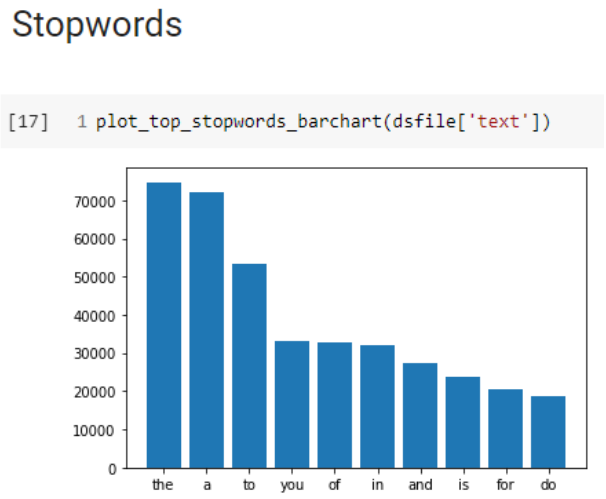

Figure 18. Humor Dataset: Stopwords

7.5. Top non stopwords

Top non stopwords bar chart allows to idenfity the main words (without stopwords) in the humor phrases. The most used are question words (See Figure 19), this let identify an important relation between questions and humour.

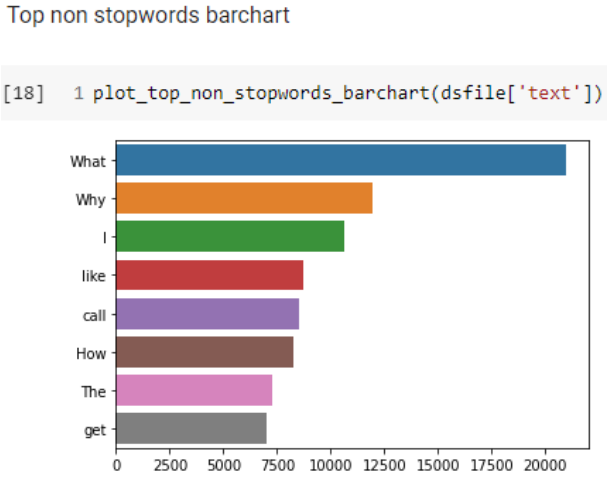

Figure 19. Humor Dataset: Top of nonstopwords

7.6. N-grams

Ngrams are simply contiguous sequences of n words. We prepare 2-grams, 3-grams and 4-grams:

7.6.1. Top bigrams. Sequences of 2 words, see Figure 20. In this chart is possible to identify the top word pairs most used. Most of them are stopwords with correspond to the two words that begin a question. It is possible to identify a correlation between questions and humor.

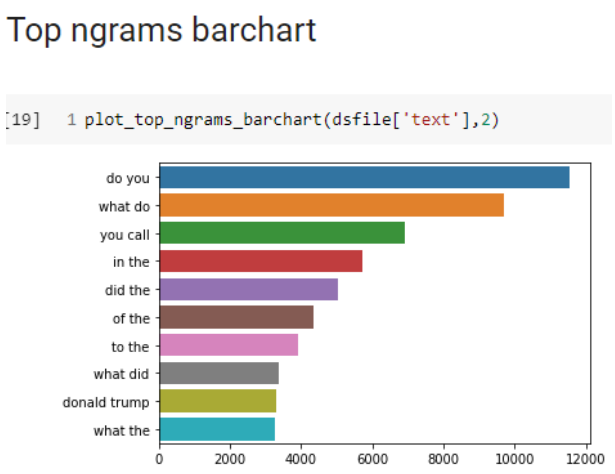

Figure 20. Humor Dataset: Top of bigrams

7.6.2. Top 3grams. Sequences of 3 words, see Figure 21. In this chart is possible to identify the top three word sentences most used. Most of them are the most common three words used to elaborate a question. This reinforces the possible correlation between questions and humor.

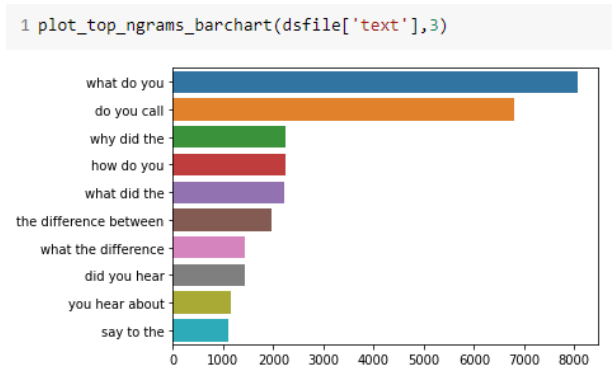

Figure 21. Humor Dataset: Top of 3grams

7.6.3. Top 4grams. Sequences of 4 words, see Figure 22. In this chart is possible to identify the top four word sentences most used. Most of them are the most common sentences the people use to begin a joke using a question.

7.7. Wordcount

Finally, we could identify in a same picture all words using Wordcloud that is a great way to represent text data. The size and color of each word that appears in the wordcloud indicate it's frequency or importance, , see Figure 23.

8. Data Analysis

This section describes the methodology and approaches that have been used for the preparation of this work. The

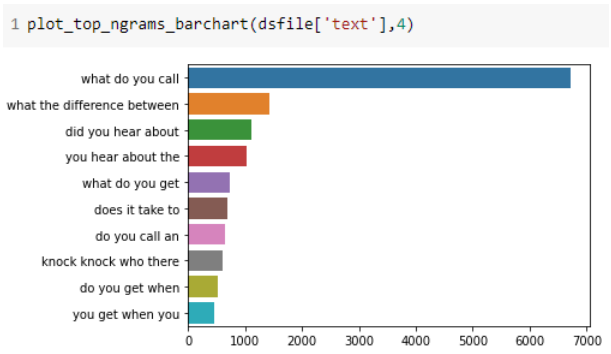

Figure 22. Humor Dataset: Top of 4grams

Figure 23. Humor Dataset: Wordcount

main objective of this analysis is to compare some artificial intelligence algorithms that allow to identify if a sentence is funny or not.

8.1. Methodology

The methodology used follows three steps, as can be seen in Figure 24 :

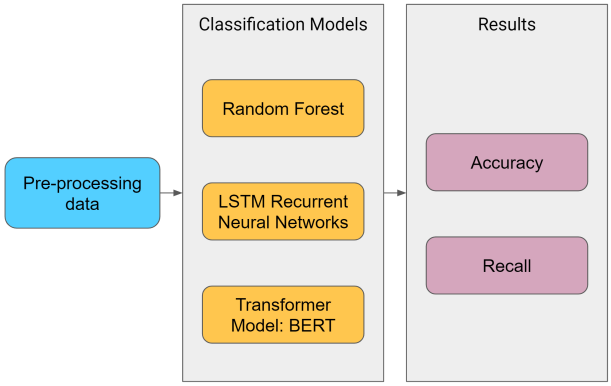

Figure 24. Data analysis: Methodology

8.1.1. Pre-processing data. In this stage, the dataset is prepared for processing. It's necessary to:

1) Validate that there are no null samples
2) Remove special characters
3) Remove stopwords
4) Replace uppercase words with their corresponding lowercase

8.1.2. Classification models. For this work some artificial intelligence applications were selected to take advantage of their ability to modeling natural language processing problems and use them for the classification of humorous phrases. They are Random Forest, LSTM recurrent neural networks and an applications of transformer model known as Bert.

8.1.3. Results comparation. The last stage seeks to compare the results obtained from the application of the selected models in terms of accuracy and recall.

9. Models

9.1. Random Forest

9.1.1. Model Description. Random forest is a solution proposed by Leo Breiman in 1996 with the intention of finding a set of predictors with a set of decision trees growing on randomly selected data subspaces.

Random forest is based on tree-based models because they form the basic components of the random forest algorithm. A tree-based model involves the recursive partitioning of the given dataset into two groups based on a given criteria until a predetermined stopping condition is met, as the figure25 shows. At the bottom of decision trees are so-called leaf or leaf nodes.

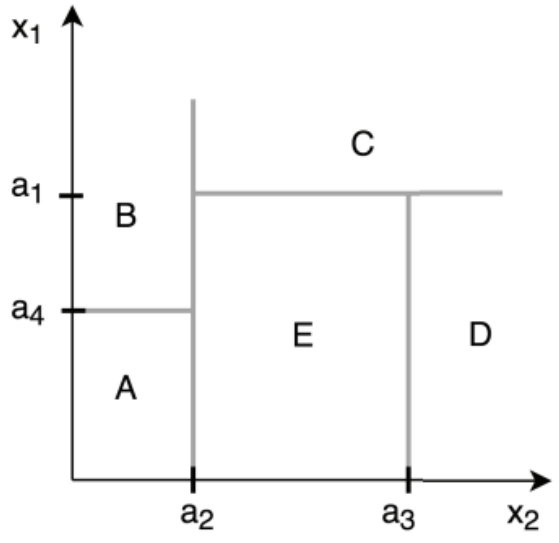

Figure 25. Recursive binary partition of a two-dimensional subspaces

Depending on how the partition and stopping criteria are set, decision trees can be designed for both classification tasks (categorical outcome, for example, logistic regression) and regression tasks (continuous outcome). For both classification and regression problems, the subset of predictor variables selected to split an internal node depends on predetermined splitting criteria that are formulated as an optimization problem. The decision tree for the previous example of decision trees is shown in the figure 26, this example is taken from the paper [21]

https://www.overleaf.com/project/625b5ffff6325807ea850fee

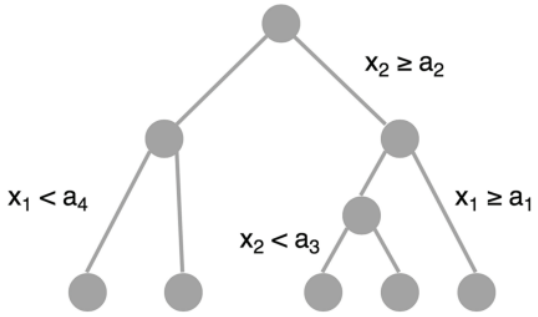

Figure 26. graphical representation of the decision tree

The random forest model is an ensemble tree-based learning algorithm; that is, the algorithm averages predictions over many individual trees. The individual trees are built on bootstrap samples rather than on the original sample.

Individual decision trees are easily interpretable, but this interpretability is lost in random forests because many decision trees are aggregated. However, in exchange, random forests often perform much better on prediction tasks, The figure 27 is a graphical representation of random forest.

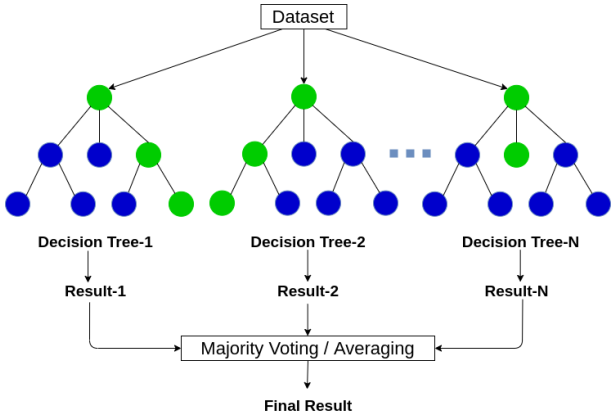

Figure 27. diagram of random forest

The random forest algorithm more accurately estimates the error rate compared with decision trees. More specifically, the error rate has been mathematically proven to always converge as the number of trees increases.

9.1.2. Implementation. For the implementation, a google colab tool was used and it was developed in python and with the use of the specialized package sklearn for the creation of trees and automatic learning. The solution was taken from https://www.kaggle.com/code/vishnu0399/humour-detection-using-nlp.

Regarding the preprocessing, the texts were transformed to lowercase and the data was cleaned (removing: special characters, shorthands, stopwords, links, accents, spaces)

9.1.3. Results. According with the proposed evaluation metrics, the results were: accuracy 0.8968, recall: 0.8996. We can conclude that this model has a great performance

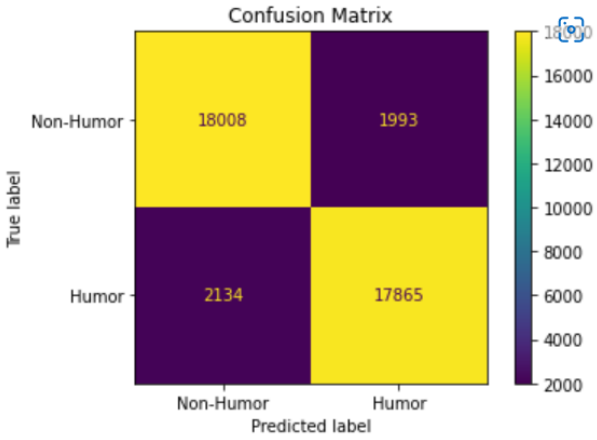

Figure 28. Confusion Matrix for the proposed Random forest

9.2. LSTM Recurrent Neural Networks

9.2.1. Model Description.

Recurrent neural network (RNN): is an extension of a conventional feedforward neural network, originally proposed in 1986 [14] which uses sequential data or time series data and is able to handle variable size inputs and outputs. These deep learning algorithms are commonly used for ordinal or temporal problems, such as time series forecasting, translation, natural language processing (NLP), speech recognition, and image captioning; they are incorporated into popular applications such as Alexa, voice search, and Google Translate. Similar to feedforward and convolutional neural networks (CNNs), recurrent neural networks utilize training data to learn. They use several variations of the backpropagation algorithm to be trained, such as Backpropagation to train. They are distinguished by their "memory" as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions so there exist also bidirectional versions of these architectures to solve this problem.

Long short Term memory LSTM RNN: Invented in the paper [15] in 1987, LSTM is an advanced RNN, a sequential network, that allows information to persist. This model is the most employed among the recurrent models. It is capable of handling the vanishing gradient problem faced frequently by RNN.

When a person interacts with a sequence of data it is important to remember the previous information, for example, while watching a video there is a need to remember the previous scene or while reading a book the previous chapters are paramount to the consistency of the book. Similarly RNNs work, they are able to "remember" the previous information and use it for processing the current input. The shortcoming of the aforementioned RNNs is, they can not remember Long term dependencies due to vanishing gradient. LSTMs are explicitly designed to avoid long-term dependency problems. In other words, the net has a way to relate distant inputs in time, for example words that are far in the same paragraph. The most popular architecture of LSTM is called Vanilla, and below some important aspects of these models are described. [16].

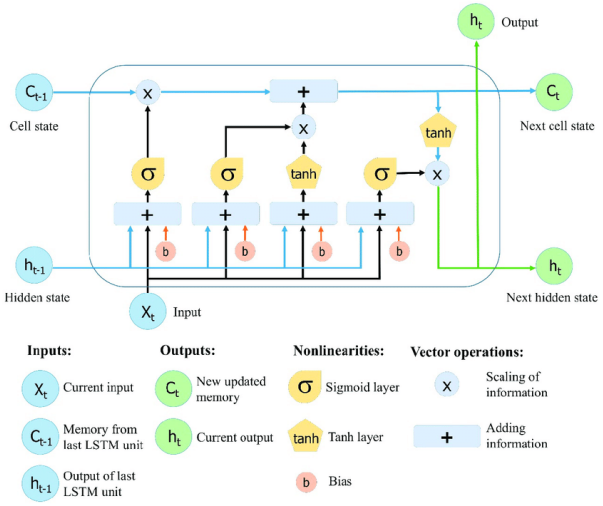

Figure 29. Model LSTM architecture (towardsdatascience.com)

- Forget Gate The LSTM architecture introduced the forget gate , enabling the LSTM to reset its own state. This allowed learning of continual tasks such as embedded Reber grammar.
- Peephole Connections In order to learn precise timings, the cell needs to control the gates. So far this was only possible through an open output gate. Peephole connections (connections from the cell to the gates) were added to the architecture in order to make precise timings easier to learn. Additionally, the output activation function was omitted, as there was no evidence that it was essential for solving the problems that LSTM had been tested on so far.

- Full Gradient The final modification towards the vanilla LSTM was the full backpropagation through time (BPTT) training for LSTM networks with the architecture described. Using full BPTT had the added advantage that LSTM gradients could be checked using finite differences, making practical implementations more reliable.
- Other Variants: Since its introduction the vanilla LSTM has been the most commonly used architecture, but other variants have been suggested too. Before the introduction of full BPTT training, Cho et al [17] proposed a simplified variant of the LSTM architecture called Gated Recurrent Unit (GRU) which has less parameters. They used neither output activation functions nor peephole connections, and coupled the input and the forget gate into an update gate. In general GRU exhibits similar results compared with LSTM for the most part of applications.

.

9.2.2. Implementation. For the implementation, it was used a google colab tool and was developed in python and with use of keras specialised package for machine learning. The architecture was inspired in https://www.kaggle.com/code/saaranshpandey/humor-detection-model-accuracy-0-95 with a single Bidirectional LSTM layer of 256 neurons plus a dense layer of one neuron with sigmoid activation function, also a Dropout of 0.2 was included to prevent overifitting,. As to the representation of the words, Glove embedding was used. GloVe is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

Concerning to the preprocessing, the texts was transformed into lowercase and the contractions were removed by a dictionary of contractions available in a homonym package (contractions.fix(val[0])), the ponctuation signs were kept. Additionally the boolean targets describing the humor of the phrase were transformed into 1's for *True* and 0's values for False. After, as is usual in the Machine learning context, the database was split into training and validation sets with a proportion of 70% for training set and 30% for the validation set which allows to adjust the hyperparameters of the model. It is important to note that we used a bidirectional LSTM to take into account words that are after each processed word.

9.2.3. Results. The model was trained online in google colab for five epochs, bith a batch size of 64 and approximately during 15 minutes. The model exhibits a reasonably good ability to distinguish or interpreting humor (Accuracy and recall over 95%). Other architectures with more LSTM layers or even other similar layers as GRU, and extra dense layers were trained but the results were similar among these architectures so we opt for the simpler one according to the parsimony's principle..Next we can see the confusion matrix of the model that describe its performance.

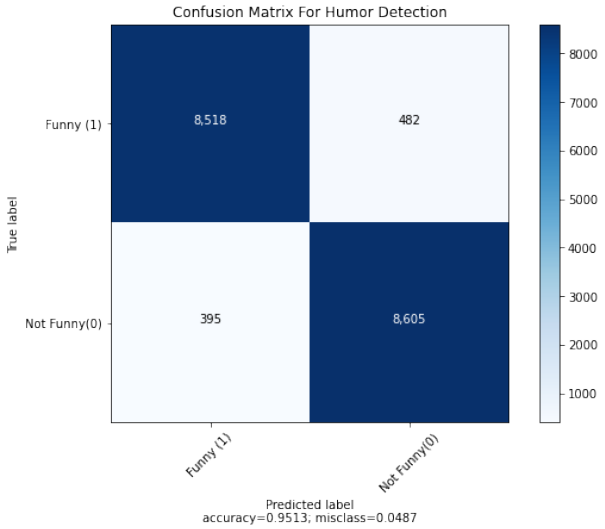

Figure 30. Confusion Matrix for the proposed Model LSTM

9.3. BERT

9.3.1. Model Description. The Transformer Neural Networks are an special architecture proposed by Ashish Vaswani in 2017 in the paper "Attention Is All You Need" [19]. A transformer model is a neural network that can apply mathematical techniques to learns context and find relationships between sequential data. The architecture of BERT is shown in Fig.25, there is possible to identify some structures: Input and output embeddings, positional encodings, Encoder, Decoder, a linear layer and a last classifier layer (Softmax).

The transformers represented an important precedent due to their ability to process data in any order, which facilitated the implementation of pre-training models such as BERT (Bidirectional Encoder Representations from Transformers). Bert has been trained with millions of data from the internet, this is why it offers great benefits especially if fine tuning is used for special applications. Bert uses just the encoder structure of transformers as is shown in Figure 31. There are two implementations of Bert: Base which uses twelve transformer encoders and Large which uses twenty four transformer encode (see Figure 32).

9.3.2. Implementation. We implement a BERT Base Classifier Model inspired in the laboratory 5.3 of [20] of sentiment analysis. The architecture and parameters proposed by [20] was modified and used to humor detection. So, the implemented architecture is shown in Figure 33. The last layers represent the fine tuning for BERT and are: one LSTM layer using a Hyperbolic Tangent Activation Function, one feed forward neural network with Relu Activation

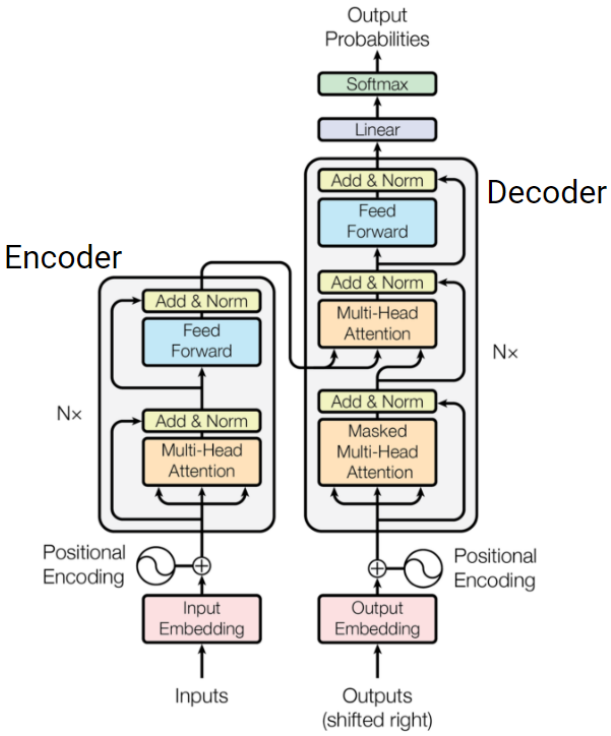

Figure 31. Transformers: Network's Architecture. Image from [19]

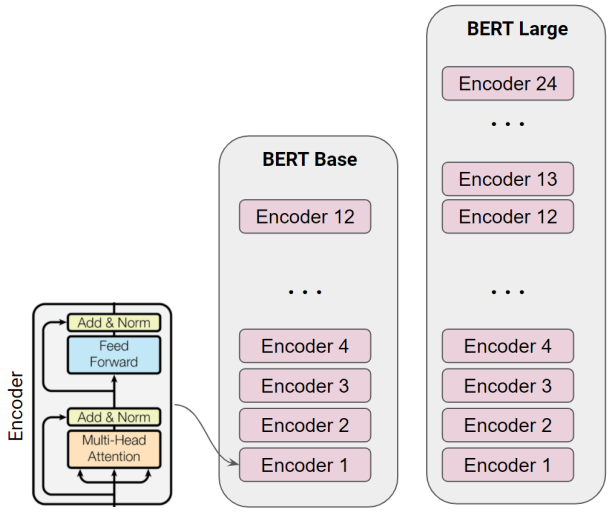

Figure 32. BERT: Network's Architecture

Function and an output Layer using an Sigmoid Activation Function. Were used batches of 32 samples and 5 epochs. We took approximately 34 minutes to train the model.

9.3.3. Results. According with the proposed evaluation metrics, the results were: accuracy 0.9567, recall: 0.9548. We can conclude that this model has a great performance. The accuracy is very high and it only took 35 minutes to train it, which was thanks to the use of Bert's pre-trained module. As it can see in the Fig. precision-by-epoch , it

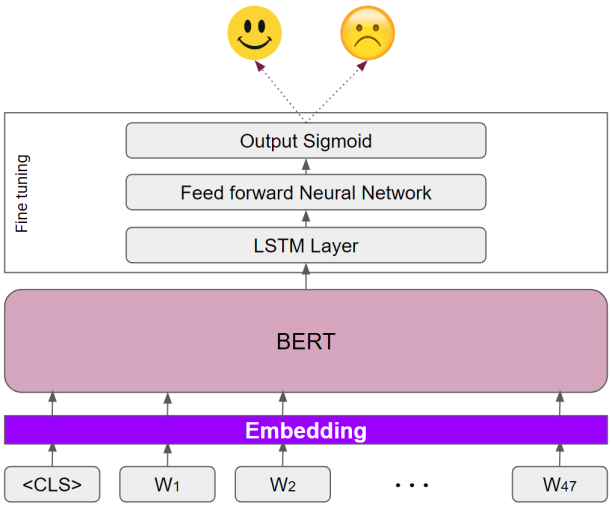

Figure 33. BERT: Implementation Model Implementation

didn't take many steps to achieve this. Even in the training stage, a slight deterioration in the result is seen when the epochs are increased.

, that allow us to conclude that the model performance is very good, specially having in account that the pre-trained BERT module . The accuracy chart by epoch is shown in the Figure 34 and the confusion matrix is shown in Fig. 35.

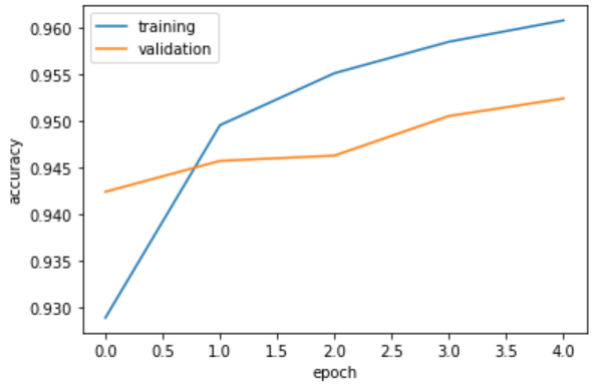

Figure 34. BERT: Accuracy vs Epochs

10. Results

The summarized results are shown in Table 2. The best accuracy was obtained by Bert model, although it was only slightly better than the recurrent LSTM model. In terms of execution time the fastest model was the random forest even if its accuracy was 5% less.

11. Contribution of the Team Members

The contribution of each member of team has been tabulated in the Table 2:

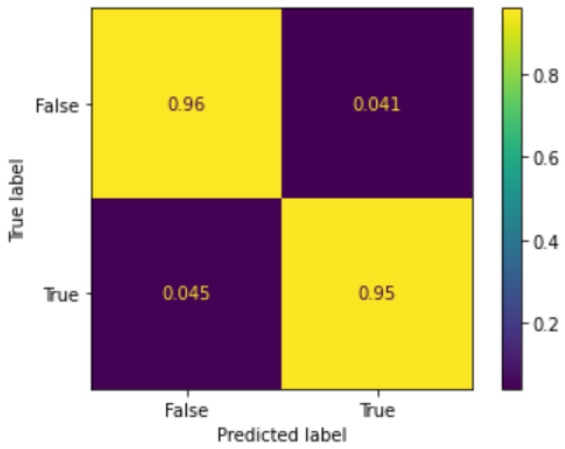

Figure 35. BERT: Confusion matrix

Model	Accuracy	Recall	Approx.Execution time (min)
Random Forest	0.8968	0.8996	10
LSTM	0.9548	0.9567	15
BERT	0.9567	0.9548	35

TABLE 2. RESULTS OF DATA ANALYSIS

Name	Activities
Joan Bofill	Approach with LSTM and report preparation
Camilo Mosquera	Approach with Random Forest and report preparation
Leyla Becerra	Approach with BERT, updating of github repository and report preparation

TABLE 3. CONTRIBUTION OF THE TEAM MEMBERS

12. Future works and conclusions

Further comparison can be developed in the future among new models perhaps with very innovative characteristics or classic models such as Support Vectoir machines combined with some representation technique as Word2Vec. Besides, it would be interesting to find other databases for testing the humor detection models apply on this document.

The results achieved with the selected Random Forest, LSTM Recurrent Neural Networks and BERT models show good performance compared to other models used for humor identification [9], as is shown in Figure 36 . However, it is important to highlight that a balance between performance metrics and the required computational infrastructure is necessary. Those models whose computational cost is very high are unlikely to be implemented, especially in mobile devices that are widely used in applications but whose computational power is limited.

Method	Configuration	Accuracy	Recall
Decision Tree		0.786	0.821
SVM	sigmoid, gamma=1.0	0.872	0.880
Multinomial NB	alpha=0.2	0.876	0.902
XGBoost		0.720	0.777
XLNet	XLNet-Large-Cased	0.916	0.973
Colbert		0.982	0.974

Figure 36. Comparison of different methods, Taken from [9]

There is a new model that represents a great opportunity to test and compare: ELECTRA

Due to the growing use of applications in which humans interact with computers, as well as numerous chat bots recognize the human request and automatically answer them, it is necessary to continue investigating mechanisms that facilitate this interaction, specially with the best performance in terms of pattern recognition and also computational requirements.

13. Project's Github

We have prepared a github repository in to handle the data, software and documentation of our project. The access data are:

Link:https://github.com/lrbecerrab/ids2022iteam3.git
Access via an invitation.

References

[1] Xinru Yan and Ted Pedersen. 2017. Who's to say what's funny? a computer using language models and deep learning, that's who! arXiv preprint arXiv:1705.10272.

[2] B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," 2008, doi: 10.1016/j.infsof.2008.09.009.

[3] "What makes us laugh? Investigations into Automatic Humor Classification - ACL Anthology." https://aclanthology.org/W18-1101/ (accessed Apr. 17, 2022).

[4] A. Jaiswal, M. Monika, A. Mathur, P. Prachi, and S. Mattu, "Automatic Humour Detection in Tweets using Soft Computing Paradigms," Proc. Int. Conf. Mach. Learn. Big Data, Cloud Parallel Comput. Trends, Prespectives Prospect. Com. 2019, pp. 172–176, Feb. 2019, doi: 10.1109/COMITCON.2019.8862259.

[5] G. G. Chowdhury, "Natural language processing," Annu. Rev. Inf. Sci. Technol., vol. 37, pp. 51–89, 2003, Accessed: Apr. 17, 2022. [Online]. Available: http://eprints.cdlr.strath.ac.uk/2611/.

[6] R. Mihalcea and C. Strapparava, "Making Computers Laugh: Investigations in Automatic Humor Recognition," Accessed: Apr. 17, 2022. [Online]. Available: http://www.mutedfaith.com/funny/life.htm.

[7] M. K. Hasan et al., "UR-FUNNY: A Multimodal Language Dataset for Understanding Humor," EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., pp. 2046–2056, 2019, doi: 10.18653/V1/D19-1211.

[8] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[9] I. Annamoradnejad and G. Zoghi, "ColBERT: Using BERT Sentence Embedding for Humor Detection," Apr. 2020, Accessed: Apr. 17, 2022. [Online]. Available: http://arxiv.org/abs/2004.12765.

[10] F. Shatnawi, M. Abdullah, and M. Hammad, "MLEngineer at SemEval-2020 Task 7: BERT-Flair based Humor Detection Model (BFHumor)," 14th Int. Work. Semant. Eval. SemEval 2020 - co-located 28th Int. Conf. Comput. Linguist. COLING 2020, Proc., pp. 1041–1048, 2020, doi: 10.18653/V1/2020.SEMEVAL-1.136.

[11] Weller, O., Seppi, K. (2019). Humor detection: A transformer gets the last laugh. arXiv preprint arXiv:1909.00252.

[12] Komorowski, M., Marshall, D. C., Salciccioli, J. D., Crutain, Y. (2016). Exploratory data analysis. Secondary Analysis of Electronic Health Records, 185–203.

[13] A Basic Guide to Initial and Exploratory Data Analysis — by Kovid Rathee — Towards Data Science. (n.d.). Retrieved May 20, 2022, from https://towardsdatascience.com/a-basic-guide-to-initial-and-exploratory-data-analysis-6d2577dfc242

[14] Michael I. Jordan. Serial order: A parallel distributed processing approach. Technical Report 8604, Institute for Cognitive Science, University of California, San Diego, 1986.

[15] Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation,9(8), 1735-1780.

[16] Greff, Klaus, et al. "LSTM: A search space odyssey." IEEE transactions on neural networks and learning systems 28.10 (2016): 2222-2232.

[17] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN EncoderDecoder for Statistical Machine Translation. arXiv preprint arXiv:1406.1078, 2014. URL http://arxiv.org/abs/1406.1078.

[18] Seltman, H. J. (2018). Experimental Design and Analysis. Chapter 4: Exploratory Data Analysis

[19] Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł.,Polosukhin, I. (n.d.). Attention Is All You Need.

[20] Ramos, R., Arias, J., Fundamentos de Deep Learning, Universidad de Antioquia. https://rramosp.github.io/2021.deeplearning/intro.html. Accessed on 22/06/2022

[21] Schonlau M, Zou RY., The random forest algorithm for statistical learning. The Stata Journal.