Universidad Autónoma de Madrid
Escuela politécnica superior

MSc in Bioinformatics and Computational Biology

MSc Project

AUTOMATIC INFORMATION SEARCH FOR COUNTERING COVID-19 MISINFORMATION THROUGH SEMANTIC SIMILARITY

Author: Álvaro Huertas García
Directors: Manuel Sánchez-Montañés Isla,
Alejandro Martín García

February 2021

AUTOMATIC INFORMATION SEARCH FOR COUNTERING COVID-19 MISINFORMATION THROUGH SEMANTIC SIMILARITY

Author: Álvaro Huertas García
Directors: Manuel Sánchez-Montañés Isla,
Alejandro Martín García

Department of Computer Engineering
Escuela Politécnica Superior
Universidad Autónoma de Madrid
February 2021

Abstract

Information quality in social media is an increasingly important issue and misinformation problem has become even more critical in the current COVID-19 pandemic, leading people exposed to false and potentially harmful claims and rumours. Civil society organizations, such as the World Health Organization, have demanded a global call for action to promote access to health information and mitigate harm from health misinformation. Consequently, this project pursues countering the spread of COVID-19 infodemic and its potential health hazards.

In this work, we give an overall view of models and methods that have been employed in the NLP field from its foundations to the latest state-of-the-art approaches. Focusing on deep learning methods, we propose applying multilingual Transformer models based on siamese networks, also called bi-encoders, combined with ensemble and PCA dimensionality reduction techniques. The goal is to counter COVID-19 misinformation by analyzing the semantic similarity between a claim and tweets from a collection gathered from official fact-checkers verified by the International Fact-Checking Network of the Poynter Institute.

It is factual that the number of Internet users increases every year and the language spoken determines access to information online. For this reason, we give a special effort in the application of multilingual models to tackle misinformation across the globe. Regarding semantic similarity, we firstly evaluate these multilingual ensemble models and improve the result in the STS-Benchmark compared to monolingual and single models. Secondly, we enhance the interpretability of the models' performance through the SentEval toolkit. Lastly, we compare these models' performance against biomedical models in TREC-COVID task round 1 using the BM25 Okapi ranking method as the baseline. Moreover, we are interested in understanding the ins and outs of misinformation. For that purpose, we extend interpretability using machine learning and deep learning approaches for sentiment analysis and topic modelling. Finally, we developed a dashboard to ease visualization of the results.

In our view, the results obtained in this project constitute an excellent initial step toward incorporating multilingualism and will assist researchers and people in countering COVID-19 misinformation.

Keywords

Natural Language Processing, Machine Learning, Deep Learning, Transformers, Topic Modeling. Sentiment Analysis, Semantic search, COVID-19, Infodemic, Misinformation, Twitter, Fact-checking

Acknowledgment

I would like to thank the following people for their support, without whose help this work would never have been possible. My friends Samuel Gómez Sánchez and José Antonio Marín Rodríguez for brainstorming ideas and giving great advice. My directors and colleagues who have guided me through the immersive world of Natural Language Processing. Lastly, my family who has supported me in every step of this project.

Contents

List of Figures ix

List of Tables xi

1 Introduction 1

1.1 Motivation . . . 1

1.2 Goals and task definition . . . 2

2 Background and State-of-the-art in NLP 3

2.1 Natural Language Processing (NLP) . . . 3

2.1.1 Brief History . . . 3

2.1.2 Principles, Applications and Tasks . . . 4

2.2 Sub-symbolic and Feature extraction . . . 6

2.3 State-of-the-art: Deep Learning based methods . . . 7

2.3.1 Recurrent Neural Networks . . . 8

2.3.2 Transformers . . . 9

2.3.3 Bidirectional Encoder Representations from Transformers (BERT) . . . 10

2.3.4 Robustly optimized BERT approach (RoBERTa) . . . 12

2.3.5 Sentence embeddings using Siamese architecture . . . 12

2.3.6 Multilingual models . . . 12

2.3.7 Knowledge Distillation . . . 14

2.4 Benchmark Datasets . . . 15

2.5 NLP, Misinformation and COVID-19 . . . 16

3 Methodology 19

3.1 Semantic similarity: Sentence Transformers . . . 20

3.1.1 Ensemble method . . . 21

3.1.2 Semantic Textual Similarity metric: Cosine similarity . . . 21

3.1.3 Dimensionality Reduction . . . 22

3.2 Data used for Semantic Search . . . 22

3.2.1 Multilingual Data for Dimensionality Reduction . . . 22

3.2.2 STS Benchmark for PCA model selection . . . 23
3.2.2.1 STS Benchmark expanded to new languages: Google Translator 23
3.2.2.2 PCA model selection criteria . . . 23
3.2.3 Evaluation Benchmarks . . . 24
3.3 Sentiment Analysis . . . 25
3.3.1 Datasets . . . 25
3.3.2 Building and Evaluating Machine Learning and Deep Learning models . . 26
3.4 Topic Modeling . . . 27
3.5 Dashboard . . . 27
3.5.1 Tweets collection . . . 28

4 Experiments and Discussion **29**
4.1 Semantic Similarity . . . 29
4.1.1 Insight into embeddings . . . 31
4.1.2 TREC-COVID: evaluation on scientific biomedical documents field . . . 32
4.2 Sentiment Analysis . . . 33
4.2.1 Binary sentiment classification for English texts . . . 34
4.2.2 Multi-class sentiment classification for Spanish texts . . . 36
4.3 Topic Modeling . . . 39
4.4 Dashboard . . . 41

5 Conclusion and Future Work **43**
5.1 Conclusions . . . 43
5.2 Future work . . . 44

Glossary **45**
5.3 General acronyms . . . 45
5.4 Languages acronyms . . . 46

References **47**

A Semantic Search Appendix **53**

B Sentiment Analysis Appendix **57**
B.1 Sentiment Analysis . . . 57

C Topic Analysis Appendix **61**
C.1 Parameters used for Topic modeling based on BERTopic . . . 61
C.2 Topics and most representative words . . . 61

List of Figures

2.1 Analysis of phrase structure through symbolic approach 4
2.2 Relationship between NLP and NLU and their associated tasks 6
2.3 Comparison between classic and deep learning NLP pipelines 7
2.4 Recurrent Neural Network (RNN) representation 8
2.5 Transformers-based model and Encoder architectures 9
2.6 BERT base architecture and workflow visualization 11
2.7 Comparison between Bi-encoders and Cross-encoders for extracting sentence embeddings similarity 13
2.8 Conceptual view of the Natural Language Processing resource hierarchy 13
2.9 Translation Language Modeling (TLM) task visualization 14
2.10 Knowledge Distillation for creating multilingual models 15

3.1 Structure and the methodology applied 19
3.2 Ensemble and dimensionality reduction approach proposed 21

4.1 STS Benchmark for PCA model selection 30
4.2 SST2 PCA and T-SNE sentence embeddings visualization 35
4.3 ROC curve, Precision-Recall curve and Confusion Matrix for DistilRoBERTa on SST2 36
4.4 TASS PCA and T-SNE sentence embeddings visualization from FastText trained on Spanish Billion Word Corpus 37
4.5 ROC curve, Precision-Recall curve and Confusion Matrix for Distilbert-multilingual-nli-stsb-quora-ranking on TASS 38
4.6 Topic modeling visualization using multilingual ensemble models 39
4.7 Topic prediction probabilities 40

A.1 PCA cumulative variance plot in Train data 53

List of Tables

3.1 Semantic Textual Similarity Benchmark breakdown 23

3.2 SST2 and TASS datasets breakdown according to sentiments and train-dev-test splits 26

4.1 Spearman ρ and Pearson r correlation coefficient in STS Benchmark test set using the sentence representation from single multilingual models 29

4.2 Spearman ρ and Pearson r correlation coefficient in STS Benchmark test set using the sentence representation from single and ensemble models (a) without applying and (b) applying PCA 30

4.3 Evaluation of multilingual sentence embeddings using the SentEval toolkit 32

4.4 Evaluation of multilingual sentence embeddings using TREC-COVID round 1. . . 33

4.5 Test Metrics on SST2 34

4.6 Test Metrics on TASS 37

The limits of my language mean
the limits of my world.

Ludwig Wittgenstein

1 Introduction

1.1 Motivation

The Coronavirus disease (COVID-19) is the first pandemic in history in which technology and social media are being used on a massive scale to keep people safe, informed, productive and connected. At the same time, the technology we rely on to stay connected and informed is enabling and amplifying an infodemic that continues to undermine the global response and jeopardizes measures to control the pandemic. An infodemic is an overabundance of information, both online and offline. It includes deliberate attempts to disseminate wrong information to undermine the public health response and advance alternative agendas of groups or individuals [57]. These currents of thought are not harmless to health; on the contrary, they cost lives [28]. Two illuminating examples are Ebola and SARS outbreaks. During the Ebola outbreak in the Democratic Republic of Congo in 2019, misinformation included violence, mistrust, social disturbances, and targeted attacks on healthcare providers [35]. During the SARS outbreak in China in 2002–2003, fear and anxiety about contracting the disease caused social stigma against Asian people [35].

The UN system and civil society organizations are using their collective expertise and knowledge to respond to the infodemic. At the same time, as the pandemic continues to create uncertainty and anxiety, there is an urgent need for stronger action to manage the infodemic, and for a coordinated approach among states, multi-lateral organizations, civil society and all other actors who have a clear role and responsibility in combatting mis- and disinformation [57].

Social media platforms have been identified as the best sources for monitoring misinformation and dispelling rumors, stigma, and conspiracy theories among the general people. The detection, assessment, and response to them and their impact on public health are a challenge [35]. The use of Natural Language Processing (NLP) can help address these limitations. Natural Language Processing or NLP is a field of Artificial Intelligence that use computational techniques for the automatic analysis and representation of human language [85]. In his review, Young et al. [85] shows that NLP enables computers to perform a wide range of natural language related tasks at all levels, ranging from parsing to machine translation and dialogue systems, which can be applied in areas such as health care, media and finance, among others.

Lewis et al. [44] estimated that in the world, there are around 7.000 spoken languages. Regardless of the exact amount, most of the inhabitants of the globe communicate in a small

number of them. According to the 2020 report of "Instituto Cervantes" [13], some languages have a substantial native population, such as Chinese, Spanish, Hindi and English, where Spanish is the second most spoken native language after Chinese with 950 million speakers. Others do not have such robust demography, but have a broad international diffusion, like French, Arabic or Portuguese. Moreover, according to [39], nowadays there are 4.66 billion of Internet users all over the world. In October 2020, the number of Internet users increased by 321 million contrasted with October 2019. As well as spoken languages, languages do not spray uniformly on the Internet. English is the far more used, since 56.8% of all the websites use it [39]. The second and third languages most used by websites are Russian and Spanish with 7.6% and 4.6%, respectively [39]. Consequently, it is factual that the language spoken determines access to information online. So, there is no doubt that the use of NLP multilingual approaches could mitigate these problems and tackle misinformation across the globe.

For all the above reasons, combating misinformation during the COVID-19 health emergency making use of NLP tools with multilingualism is a very challenging and inspiring project, whose results can help to diminish the infodemic originated and to encourage future work on this issue.

1.2 Goals and task definition

This project's primary goal is to develop a multilingual automatic tool that helps users to contrast information related to COVID-19 in order to address the infodemic issue. For this purpose, it is required to extract the semantic meaning of the texts, aiming to provide users with the most appropriate information and evaluate the semantic similarity between pairs of texts. This approach has the potential features to be applied to journalism and media. Before this primary goal is addressed, it is needed to establish a series of partial goals:

- To explore the application of Transformer-based multilingual models combined with ensemble and PCA dimensionality reduction methods in semantic search and compare them with state-of-the-art monolingual and single models. This goal is breakdown in:
  - To study the effects and improve the performance of multilingual models through an ensemble architecture.
  - To study the effects of dimensionality reduction on the performance of the models.
  - To evaluate these multilingual models on semantic search using Semantic Textual Similarity Benchmark (STS)
  - To enhance the performance understanding from the models through the SentEval toolkit and TREC-COVID dataset.
- To develop a collection of tweets from fact-checkers Twitter accounts verified by the International Fact-Checking Network of the Poynter Institute.
- To enhance interpretability and explainability of the results apply sentiment analysis and topic modeling based on machine and deep learning approaches.
- To develop a dashboard to visualize the results and facilitate their interpretation and readability.

You shall know a word by the company it keeps.

John Rupert Firth

2

Background and State-of-the-art in NLP

In this chapter, Natural Language Processing (NLP) and its advantageous use to tackle COVID-19 infodemic will be covered. This chapter is divided into five sections. The first section gives a brief overall overview of the past-present history of NLP field. The second and third sections examine the State-of-the-art approaches used in light of the current literature in NLP. In the fourth section, several well-known benchmark datasets used to develop NLP models are described. Finally, the last section outlines the present status of NLP, misinformation and COVID-19 infodemic.

2.1 Natural Language Processing (NLP)

2.1.1 Brief History

NLP emerged in the 1950s as the intersection of Artificial Intelligence (AI) and linguistics from the need for Machine Translation (MT) during World War II [45, 54]. Two achievements in the field that should be pointed out are *Syntantic Structures* published by Chomsky in 1957 [43] and the development of ELIZA. Chomsky established NLP symbolic approach and the basis of the *regular expressions* [54]. ELIZA was developed from 1964 to 1966 at the Artificial Intelligence Laboratory of MIT and is considered the first chatbot program. It was able to replicate the conversation between a psychologist and a patient [82].

NLP *symbolic* approach used human-readable symbols to represent real-world entities (e.g., words and punctuation) as well as logic in order to create "rules" for the manipulation of those symbols (e.g., grammar rules) to teach the NLP system to understand a language [54, 85]. Advantages from this approach are that the system is a "transparent box"; there is full control on how to fine-tune the system; and high explainability [85]. Unfortunately, the rules may become uncontrollable numerous, often interacting unpredictably [54].

It was not until the 1980s when computational language comprehension became an active field of research [45, 54]. Machine Learning (ML) methods that used probabilities and statistical inference to automatically learn such rules were introduced. This orientation resulted in the birth of *statistical* NLP [54]. In the last ten years of the millennium, the field was proliferating. The reasons can be attributed to 1) increased availability of large amounts of electronic text; 2)

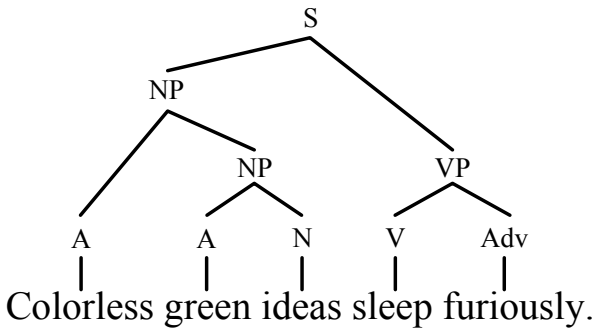


Figure 2.1: Simple example of the analysis of phrase structure through symbolic approach [30]. Where NP means Noun Phrase; VP Verb Phrase; A Adjective; Adv Adverb: and S Sentence.

availability of computers with increased speed and memory, and 3) the emerging of the Internet [45].

NLP growth continued during the last two decades, where the first significant advancement came in 2013 with the breakthrough of Word2Vec [52]. At this point, NLP systems behave as "black boxes" because real-world entities are represented as vectors. This type of representation implies a reduction in explainability, but a better understanding of natural language. Thanks to Word2Vec, the term *word embedding* and the usage of neural networks were introduced to the research field [52]. This turning point, simultaneously with the advances in deep learning and increasing computational capabilities, set the seeds for incorporating Recurrent Neural Networks (RNN) in 2014 [37, 40]. Attention-based networks became popular around the years 2015–2016 [78]. A specific type of attention-based network introduced in 2017, the Transformer model, has been incredibly dominant in modern NLP architecture [88]. The apparition of Bidirectional Encoder Representations from Transformers (BERT) introduced in 2018 by Google [23] was just the tip of the iceberg for attention-based models. Since then, more models have shown up (e.g., XML, RoBERTa, XML-RoBERTa).

In short, NLP is an ever-changing developing field that has evolved since its origins in the 1950s to our current days thanks to the contribution of the advances in increasing computational capabilities. NLP approaches can be found during history, where NLP research has changed from a symbolic and statistical approach towards a state-of-the-art deep learning era.

In the literature, authors refers to two different methods for analyzing human texts, Natural Language Processing and Text Mining. Natural Language Processing (NLP) combines linguistics and AI to enable computers to understand human natural language input [85]. On the other hand, Text Mining is based on the extraction of useful information hidden inside the redundancy of the natural language. Both approaches share that the data used comes in an unstructured way (e.g., clinical history, papers, social media comments) and it is organized into structured data to accomplish a determined task [54]. Therefore, the difference resides in the methodology of the text analysis. Text mining techniques use the words themselves as a unit of analysis (e.g., frequencies, the presence or absence of specific words of interest) and do not consider the text structure [26]. However, NLP methods usually involve the text structure because capturing context and meaning from the text matters.

2.1.2 Principles, Applications and Tasks

NLP researchers aim to create a system capable of processing and interpreting natural language, just like humans use language as a communication and reasoning tool [15, 77]. Natural language differs from formal languages (e.g., a programming language) in the absence of formalism. Indeed, that is the reason to make use of the NLP learning approach. If a formal language

were to be studied the construction and logic issues bound to the language's formalism would already be known, and predefined [77].

The foundations of NLP lie in several disciplines such as computer and information science, linguistics, mathematics, artificial intelligence and robotics, and psychology [15]. The most explanatory method for explaining how an NLP system works is through the *"levels of language"* [15, 45]:

- Phonology: deals with pronunciation within and across words
- Morphology: deals with morphemes, the smallest parts of words that carry meaning (e.g., root, suffixes, and prefixes of a word).
- Lexical: deals with the interpretation of the meaning of individual words.
- Syntactic: analyzes the words in a sentence to uncover the grammatical structure of the sentence.
- Semantic: determines the possible meanings of a sentence by focusing on the interactions among word-level meanings in the sentence.
- Discourse: as syntax and semantic but works with text unit longer than a sentence.
- Pragmatic: deals with the purposeful use of language in situations taking into account information outside the content of the document (e.g., intentions, plans).

Performing the main NLP task of a project implies that some levels are more relevant than others, and they should be combined in an NLP pipeline [86]. Thus, an NLP system may begin at the word level – to determine the morphological structure, nature (such as part-of-speech, meaning) of the words. Then may move on to the sentence level – to determine the word order, grammar, meaning of the entire sentence. Finally, to the context and the overall environment or domain [15].

At the core of any NLP task, there is the critical issue of Natural Language Understanding (NLU) [15]. The NLP subfield of Natural Language Understanding (NLU) usually results in transforming natural languages from one representation into another by understanding a text as humans do [86]. NLU deals with the problem of determining whether a natural language hypothesis h can be reasonably inferred from natural language premise p, a task named Natural Language Inference (NLI) [48]. Other tasks can be addressed to NLU: semantic parsing, question answering (Q&A), summarization, sentiment analysis, relation extraction and dialogue agents. (see Figure 2.2).

The number of NLP applications is broad, so it is the different tasks that NLP can face up [59].NLP can be applied such us for social media monitoring, sentiment analysis, survey analytic, autocompletion, spell checking, duplicate detection, text classification, text generation, machine translation, speech recognition and conversational chatbot development.

This study aimed to elucidate the use of state-of-the-art deep learning approach based on semantic level and NLU techniques to infer and understand information from texts to tackle infodemic and misinformation.

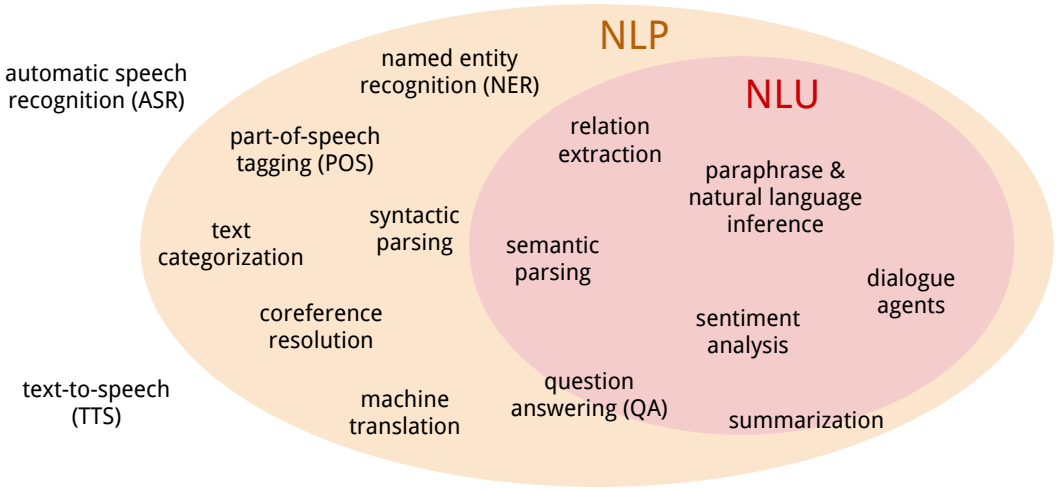


Figure 2.2: Relationship between NLP and NLU and their associated tasks [49]

2.2 Sub-symbolic and Feature extraction

As stated at the beginning of this chapter, the real task behind NLP is to transform the raw unstructured data (e.g., text) into a structured representation which allows computers to learn from the text [54]. High-level NLP uses a sub-symbolic approach to achieve this. The sub-symbolic approach represents real-world entities (e.g., words, sentences) as vectors throughout the process of feature extraction [40, 85].

The purpose of feature extraction is to derive features from the raw unstructured data and map each one into a specific number [75]. A feature is defined as a piece of information or measurable property that is useful for building an NLP system with a specific task. The resulting sequence of numbers inferred from the text is called a vector. Thus, NLP systems are not any longer "transparent box" that transform text to human-readable symbols, instead, they are "black boxes" [40].

In classical NLP systems, the feature extraction would be achieved using weighted techniques, where each word is mapped to a number matching the occurrences of that word in the data, called Bag-of-Words (BoW) or Term Frequency (TF). A notable improvement of this technique is Term Frequency-Inverse Document Frequency (TF-IDF), where the word frequency is penalized by the number of data instances (e.g., documents) that contain the word.

These classical techniques suffer from several disadvantages, such as the word order is not considered, and most importantly, semantic meaning is not incorporated because each word is counted independently. As previously mentioned, semantic deals with the meaning of a sentence by focusing on the interactions among word-level [45]. To solve this problem, researches use word embedding.

Word embedding is a feature learning technique where each word from the vocabulary is mapped to a N-dimension vector of real numbers [40]. Words with similar meanings should have similar representations [38]. The difference of word embedding from classical vectorization methods is that word embeddings are learned from the data. For example, for the word "pandemic" all the contexts where this word is used would be represented in its embedding. As a result, the vectorization of the word "pandemic" is richer in capturing the meaning of the word.

It is an undeniable fact that the use of neural networks to generate word embedding was introduced by Word2Vec technique. Word2Vec [52] is an unsupervised methodology that uses a simple neural network to create high dimension vector for each word [38, 40].

Word2Vec opened the door to the use of more complex neural networks architectures and its extension to represent sentences and documents as embeddings.

2.3 State-of-the-art: Deep Learning based methods

Deep Learning (DL) models have achieved state-of-the-art results across many domains, including a wide variety of NLP applications [40]. Figure 2.3 depicts a comparison between classical and deep learning-based NLP methods and shows a standard representation of a fully connected Deep Neuronal Network (DNN).

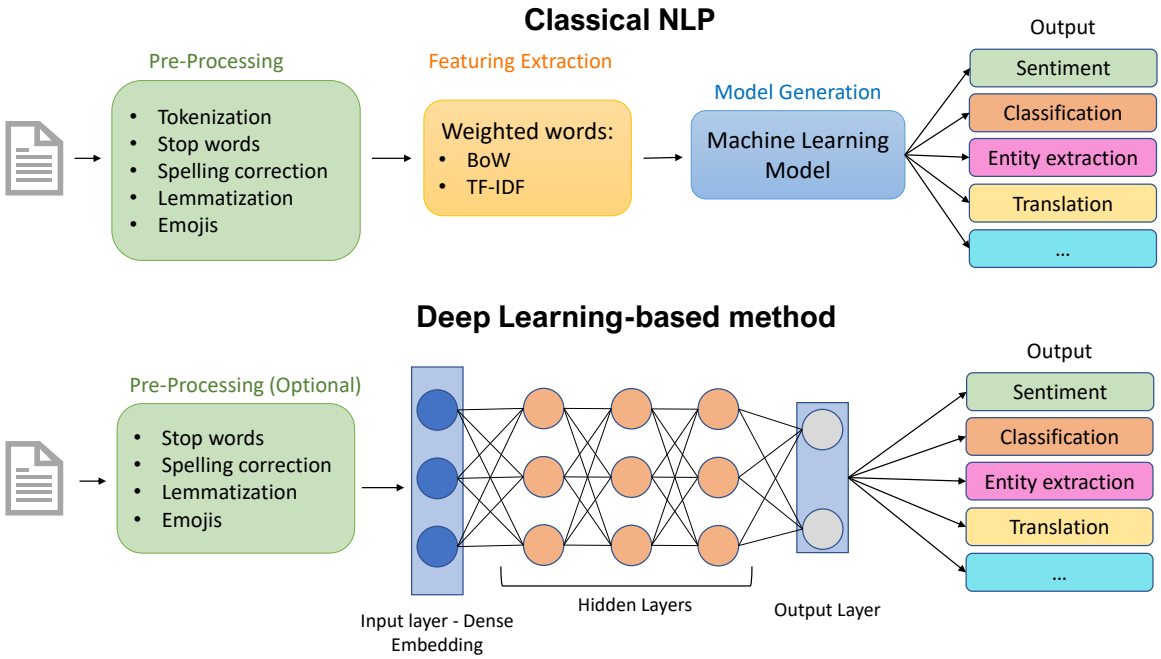


Figure 2.3: Comparison between classic and deep learning NLP pipelines. Standard fully connected deep neural network (DNN) is represented in Deep Learning-based method. Adapted from [75].

Deep learning (DL) is a subfield of machine learning that deals with deep artificial neural networks (ANNs) [1]. DL models are composed of multiple layers and multiple units within layers to represent highly complex functions. The units of a layer are also known as nodes or neurons, and every layer use the knowledge extracted from the previous layer to learn deeply [53]. Three types of layers can be distinguished in a DNN: input layer, hidden layer and output layer.

According to [40], the input layer takes care of the input data (text) vectorization. The input layer is set up via feature extraction techniques as TF-IDF or embeddings. The embeddings do not need to be learned jointly with the main task (e.g., text classification), embeddings can be pre-computed in a DL model and loaded into other models. A case in point is Gensim's pre-computed embeddings [61], which allow simple models to access embeddings resulting from larger-scale models' training. This pre-trained embeddings are an illuminating example of Transfer Learning.

Secondly, the hidden layers are the intermediate layers between the input and output layer with the tasks of learning the relationship between the input and target spaces. The number of hidden layers and the number of nodes in each layer are hyperparameters that need to be tuned

[38]. Finally, the output layer provides the result for given inputs [40]. It collects the results from the hidden layers and puts it across. The number of nodes in the output layer depends on the task [38].

2.3.1 Recurrent Neural Networks

Sentences are essentially sequences of words, and the contextual meaning of a particular word in a sentence may not be derived solely from the immediately surrounding words. It might actually be a result of some words far away in the sentence as well [38].

A neural network architecture that is used are Recurrent Neural Networks (RNNs). RNNs help us capture context and temporal relationships in sequences by assigning more weights to the previous data points of a sequence [40].

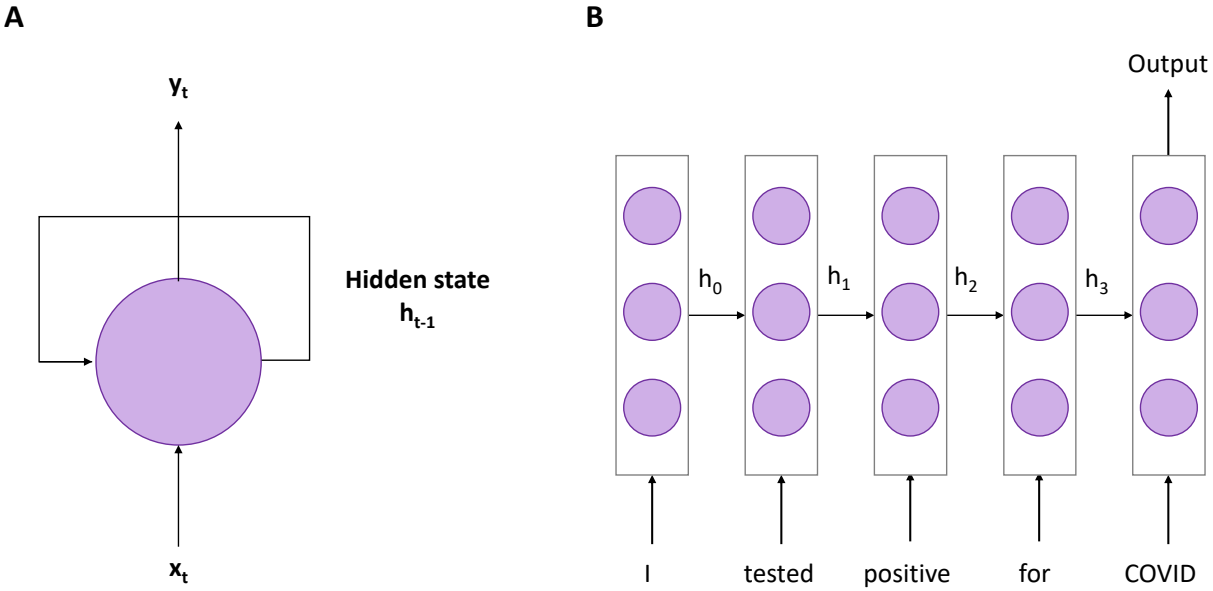


Figure 2.4: A) Representation of the feedback loop in a Recurrent Neural Network (RNN) unit, which shows how the hidden state from previous time step h_{t-1} is also provided as input in addition to the input x_t. B) Unrolled version of a many-to-one RNN wherein each rectangular block containing the circles is the neural network. At every step, the embedding for the input word is also affected by the hidden state information captured from the previous words. The final output is a sentence embedding which preserves the interaction among words. Adapted from [38].

As Figure 2.4A depicts, every recurrent neuron takes two intputs – one is the current or external input and the output from the previous state, called hidden state [38]. The hidden state acts as the neural network's internal memory that accumulates information from data seen in previous time steps [85]. In Figure 2.4A it can be seen that the output from a time step t (y_t) depends on the input at time step t (x_t) and the hidden state from step *t-1* (h_{t-1}) [38]. It is necessary to point out that, as it was established before, first the text is transformed into machine-readable vectors in the input layer, and then the RNN process the sequence of vectors.

In order to increase the memory, two variants of memory-based RNN were create: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Both of them use an internal mechanism called "gates". Gating is a technique that helps the network learn which data in a sequence is important to remember from the past (keep in the hidden state) and which should be forgotten [85]. The difference between them lies in how the hidden state is computed and the number of gates used. By and large, LSTMs are huge networks and they have a lot of

parameters. Consequently, to speed up the training and reduce the computational cost GRUs were introduced [38].

2.3.2 Transformers

The reason why Transformers are discussed in this work is because they have some advantages, such as reducing the time for training and the ability of being parallelizable [38, 78]. Moreover, as it will be explain below, Transformer-based models are state-of-the-art in NLP and they will be used in this work.

An increasing number of authors in the field think of Transformers [78] as a substitute for LSTMs due to the results on different NLP tasks [2]. This was compounded by the fact that Transformers deal with long-term dependancies better than LSTMs [2].

As explained in [38], Transformer modeling is based on converting a set of input sequences (source language) into a bunch of hidden states, which are then further decoded into a set of output sequences (target language) (see Figure 2.5A). Transformers are composed of six encoders stacked on top of each other, and has six identical decoders stacked on top of each other.

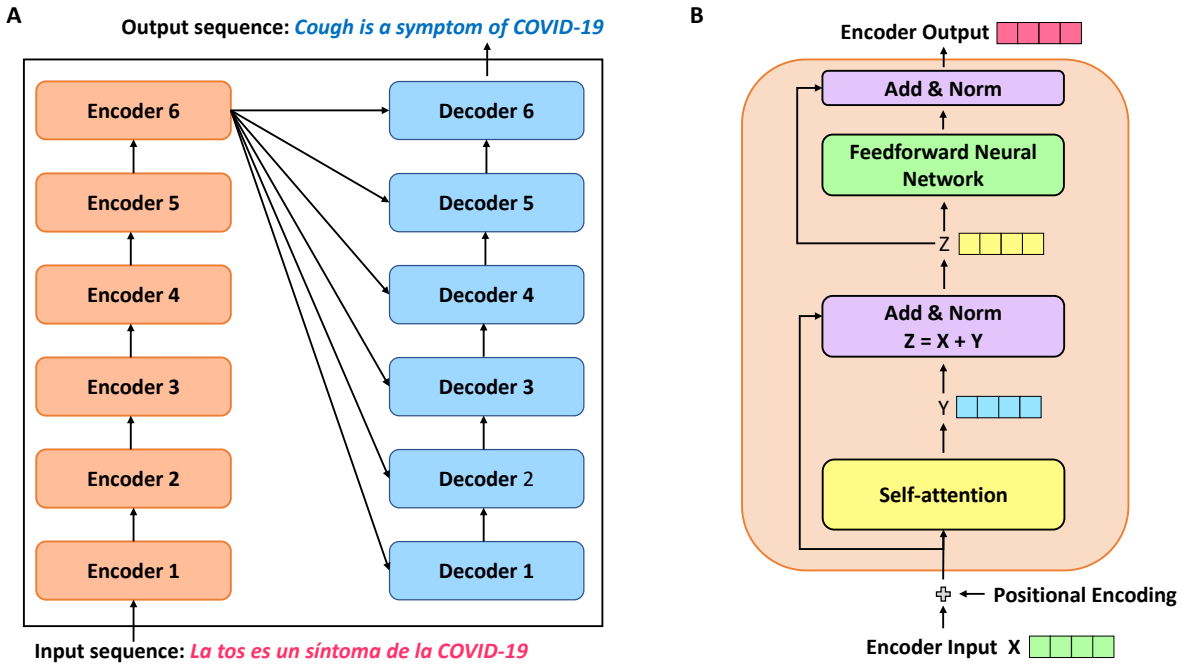


Figure 2.5: A) Transformers-based model architecture. 6 encoders stacked on top of each other connected with 6 stacked decoders. B) Encoder architecture visualization. Adapted from [38, 78].

Encoders take care of building the machine-readable representation, which captures the meaning of the input data. Decoders take this representation and build the homologous representation in the target language. Decoders are specific architectures from machine translation task. This study is focused on inferring and understanding information from texts to tackle infodemic and misinformation, by comparing sentence embeddings that condense the semantic level of language. Therefore, we will get only focused on encoders architecture.

As it can be seen in Figure 2.5B, each encoder is composed of two main components: a self-attention layer – a layer that helps the encoder look at other words in the input sequence as it encodes a specific word –, and a two-layer feedforward neural network (FFNN) that maps embedding from input space to another space. Furthermore, between these main components

there are a residual and normalization layer. All the encoders have the same structure, but they do not share weights [3].

According to [3], as the model processes each word (each position in the input sequence), self-attention allows the model to look at other positions in the input sequence that are useful for a better encoding of the input word. Consequently, self-attention mechanism is the method that Transformers use to "understand" the interaction among words, as RNN did with hidden states. The best way to understand self-attention is by an example.

Imagine the sentence "*My sister tested positive for COVID and she has to quarantine herself for 15 days*". The embedding belonging to the word *she* has to be highly influenced by the word *sister*. Self-attention mechanism reflects this association in the embeddings representation. Each encoder uses multiple attention heads, which allows the model to focus on multiple different positions, instead of just one. The outputs from the multiple attention heads are concatenated and projected to provide the final values [38].

As explained in [38], the input flowing into the first encoder is an embedding for the input sequence. The embeddings can be as simple as one-hot vectors, or other forms such as Word2Vec embeddings, and so on. The input to the other encoders is the output of the previous encoder. Along with the input for each encoders, a denominated positional embedding is added to the input. Positional embeddings try to capture the sequential order of tokens and extract features such as the relative distance between tokens [3, 38].

To sum up, the encoder receives an input embedding along with a position embedding, which are summed together and passed to a series of stacked encoders composed of self-attention and feedforward neural networks.

2.3.3 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) model [24] is one of the state-of-the-art Transformer-based model used in NLP. According to [38], BERT was built using the encoder component of the Transformer architecture. BERT architecture is nothing but a set of stacked encoders. So what are the advantages of using BERT?

It is worth regarding that, the embeddings created by Word2vec (see subsection 2.2) are static in the sense that the word embedding created for a particular word is the same whatever the context. Once a word embedding is computed during training, it will be used for the word whatever the context is [38]. On the other hand, BERT performs word embeddings that are context-sensitive – the word embedding computed changes depending on the context [38]. Moreover, according to [2, 38], Transformer-based models can be pre-trained on a broader task and later be fine-tuned to a specific task. This is referred to as Transfer Learning. As mentioned above, BERT architecture is composed of a set of stacked encoders. BERT-base and BERT-large are two variants released by the authors with a different number of stacked encoders [24]. The critical aspect of BERT remains in the model input and output, and the pre-training.

As Figure 2.6 shows, BERT takes a sequence of words as input which keep flowing up the stacked encoders. Each encoder applies self-attention, and passes its results through a feed-forward network, and then hands it off to the next encoder [2]. What is significant is that BERT model expects input data in a specific format. Two unique tokens, [CLS] and [SEP], are added at the beginning and the end of the input sequence, respectively [24, 38]. These unique tokens allow BERT to handle single-sentence and two-sentences as input data. As with Transformers, a positional embedding is added to the tokens [38]. Additionally, a segment embedding which indicates to which sentence belongs each token is added. Consequently, the input is a sum of token embedding, positional embedding and segment embedding.

The focus should be on how BERT computes the token embeddings. The BERT model was built with a vocabulary of 30,000 words and used the WordPiece tokenizer for tokenization [24, 38]. In order to build up a vocabulary, the first thing to do is break the data (e.g., document, sentence) into chunks called tokens [38]. This segmentation process is called tokenization. As explained in [50], Wordpiece tokenizer is a tokenizer that greedily creates a fixed-size vocabulary of individual characters, subwords, and words that best fits our language data. Since the vocabulary limit size of BERT tokenizer model is 30,000, the WordPiece model generates a vocabulary that contains all English characters plus the 30,000 most common words and subwords found in the English language corpus the model is trained on. Consequently, we can always represent a word as, at the very least, the collection of its individual characters [50].

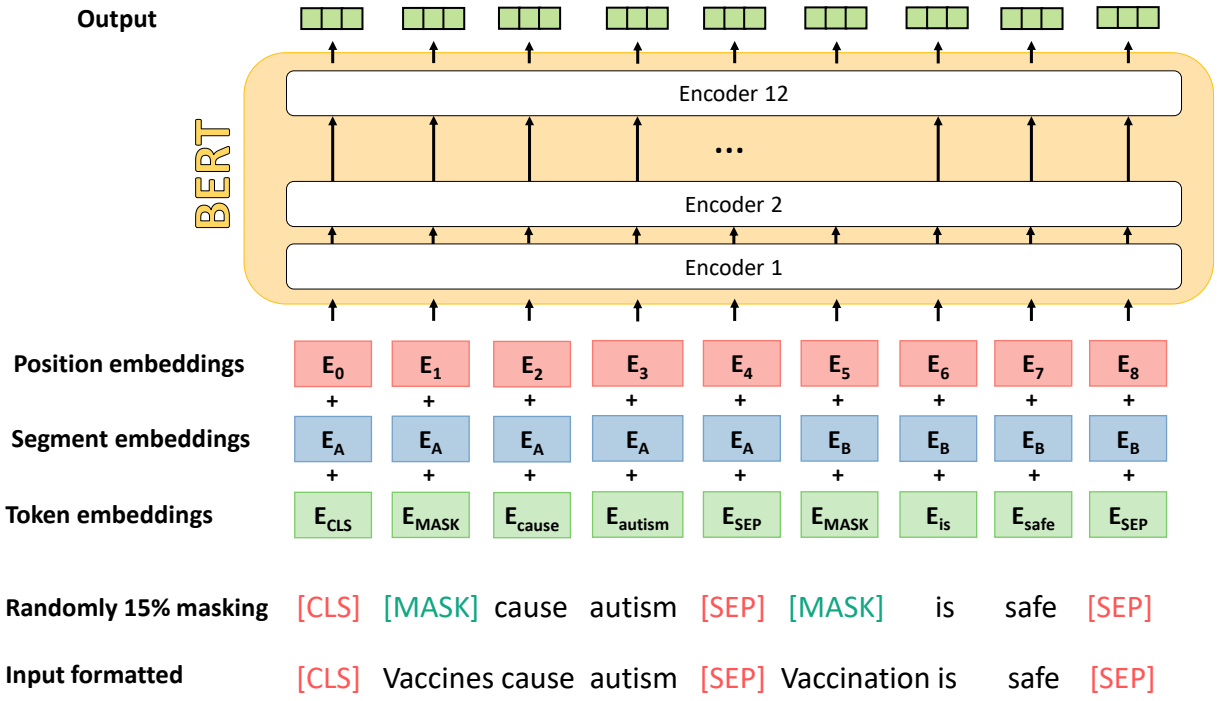


Figure 2.6: BERT base architecture and workflow visualization. Adapted from [38, 2, 24].

BERT output for each token in the input sequence is a vector of size 768 or 1024 depending on the variant [24]. These output vectors are context-sensitive embeddings can be used differently depending on the fine-tuning task to be solved, such as calculating semantic similarity in our case of interest.

The last crucial concept to understand the potential of BERT is the pre-training. BERT was originally pre-trained on the whole English Wikipedia and Brown Corpus [24]. BERT is pre-trained using two unsupervised tasks, namely the Masked Language Model (MLM) and Next-Sentence Prediction (NSP) [24, 38].

Language modelling consists of predicting a word given a set of previous words. To improve the effects of self-attention on this task, a model whose language model looks both forward and backwards the input sentence is required [24]. Looking forward and backwards refer to acknowledge the interaction among words at left and at the right of a word while encoding that word. This is called bidirectional conditioning [38]. However, in a Transformer-based model, such as BERT, bidirectional conditioning would allow each word to indirectly see itself [2, 24, 38]. To solve this, BERT picks 15% of the tokens at random and masks them. Next, it tries to predict these masked tokens [38]. Consequently, the task is named Masked Language Model (MLM). Beyond masking 15% of the input, BERT also sometimes it randomly replaces a word

with another word and asks the model to predict the correct word in that position [2].

Next-Sentence Prediction (NSP) is included in BERT pre-training to make it better at handling the relationship between multiple sentences. This task consists of calculating the probability of a sentence to be the following sentence to another sentence. That means, given two sentences (A and B), the probability of A to be the sentence that follows B, and vice versa.

In short, BERT is a Transformer-based model with high powerful at understanding natural language. It has been pre-trained to infer context-sensitive embeddings of words and the relationship between multiple sentences. Furthermore, these skills of BERT can be fine-tuned over a specific task, to get better performances.

2.3.4 Robustly optimized BERT approach (RoBERTa)

There is no doubt that BERT was a landmark in the NLP field. Since it open source releases in 2018 [23], new approaches have been proposed. A case in point is Robustly optimized BERT approach (RoBERTa) [46]. According to these researchers, BERT was significantly undertrained according to the number of key hyper-parameters and training data size.

According to [46, 47], RoBERTa is developed based on BERT but introduced several modifications. Firstly, it uses Byte-Pair Encoding (BPE) as tokenizer and increases the vocabulary size (from 30k to 50k). Moreover, RoBERTa is trained on bigger data and on longer sequences. Secondly, as well as BERT did, RoBERTa masks training data but instead doing it once, replicates the process several times masking the training data differently. This is named dynamic masking, instead of BERT static masking. Finally, the Next-Sentence Prediction (NSP) task used to train BERT is modified for inputs which represent multiple sentences from the same document or across documents [47, 46].

2.3.5 Sentence embeddings using Siamese architecture

It is well-known that BERT and RoBERTa are state-of-the-art Transformer-based models on sentence-pair regression tasks like semantic textual similarity (STS) [64] (see Figure 2.7B). However, it requires that both sentences are fed into the models, which causes a massive computational overhead [64]. To overcome this difficulties, deriving semantically meaningful sentence embeddings can be achieved by using siamese architectures for training. [64]. Siamese architectures consists of two pre-trained Transformer-based models with tied weights that can be fine-tuned on a specific task like compute similarity scores (see Figure 2.7A). This approach is also called dual-encoders o bi-encoders [34].

Bi-encoders appply self-attention and maps the sentences separately to a common features space. The main advantage of Bi-encoders is the inference speed because of the precomputation of the embeddings of all possible sentences of the system [34]. The pooling layer applied in the siamese architecture of Bi-encoders merges all the outputs from the encoder into one representation. Several pooling strategies can be followed: choose the first output of the transformer (e.g., [CLS] special token in BERT), averaging all the outputs, or computing a max-over-time of the output vectors [64, 34].

2.3.6 Multilingual models

In current machine learning, the amount of available training data is the main factor that influences an algorithm's performance. As Figure 2.8 depicts, languages are classified in high, medium or low resources depending on the computational data resources available [16, 68]. Taking advantage of the powerful Transformer-based models to solve NLP tasks is a great strategy. However, the models seen so far are monolinguals, usually only for English which is a

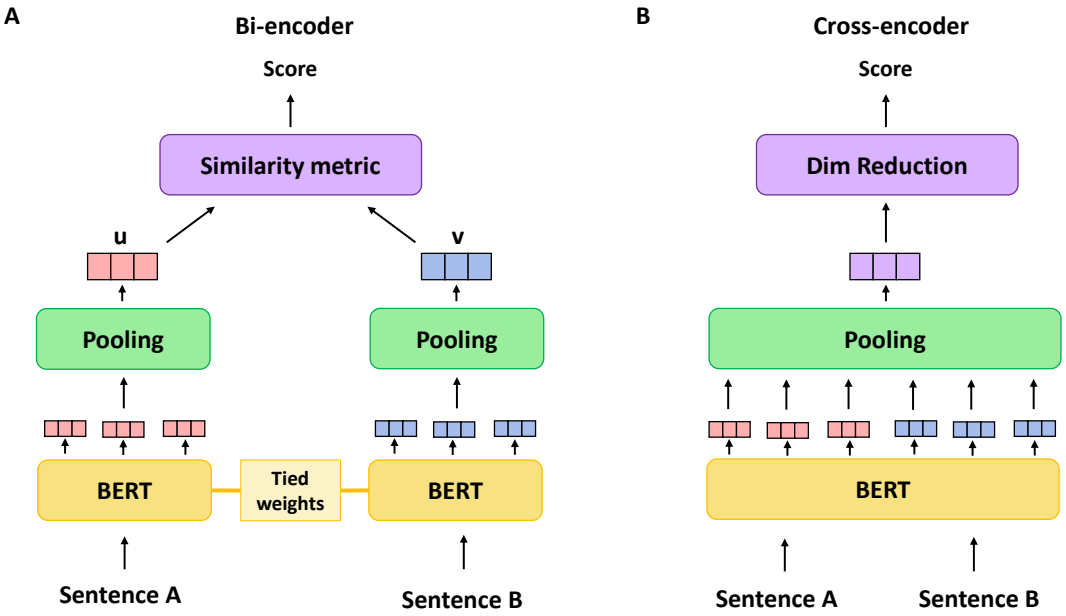


Figure 2.7: Comparison between Bi-encoders and Cross-encoders for extracting sentence embeddings similarity. BERT is the chosen encoder model just for illustration purpose. A) The Bi-encoder encodes the sentences separately. B) The Cross-encoder jointly encodes the sentences in a single transformer, achieving richer interactions between sentences at the cost of slower computation. Adapted from [34].

high-resource language [63]. Extend existing embeddings models to new languages with lower resource level is a new challenge in NLP. To cope with the challenge, researchers use cross-lingual transfer learning. The central idea underlying the cross-lingual transfer learning approach is to transfer resources and models from high-resource language to low-resource target languages [73].

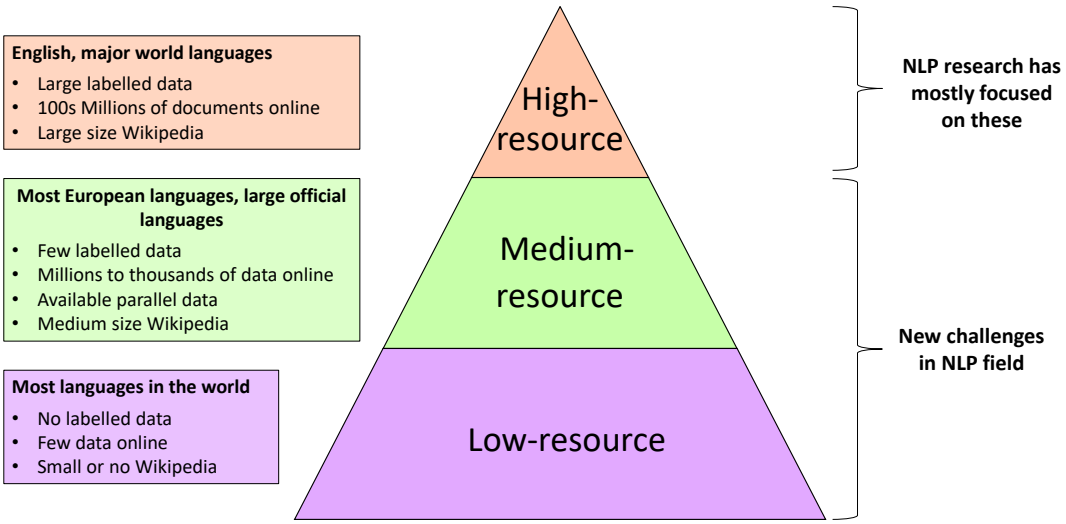


Figure 2.8: Conceptual view of the Natural Language Processing resource hierarchy. Note that many languages cannot be assigned clearly to a single level of the hierarchy. Adapted from: [68].

According to [73], multilingual embedding models map text from multiple languages to a shared embedding space (or cross-lingual space). As a result, in this embedding space, related or similar words will lie closer to each other, and unrelated words will be distant, independently of the language. In this subsection, the following models will be covered: XLM, LaBSE and mUSE.

XLM [41] was developed by Facebook AI in 2019. It is a Transformer-based model similar to

BERT, but with several modifications. Firstly, the data used to pre-train the model comes from 15 different languages. Secondly, BPE tokenizer is applied instead of WordPiece tokenizer used by BERT. Moreover, the segments embeddings are substituted for language embeddigns (see Figure 2.9). Finally, new tasks are incorporated into the pre-training. XLM is trained with the Casual Language Modeling (CLM, next token prediction), Masked Language Modeling (MLM, as BERT did 2.3.3) and Translation Language Modeling (TLM). The TLM task force the model to learn representations for different languages by using as input two parallel sentences (same sentence in two languages) and predicting mask words using words from both languages.

A derived model from XLM is XML-R [19] where the R stands for RoBERTa. It is necessary to point out that XML-R is not the combination of XLM with RoBERTa. XML-R is just RoBERTa model trained on a vast multilingual dataset (2.5 TB) with the training task of MLM. Another remarkable difference of XML-R from RoBERTa is the vocabulary size (250k), which is five times bigger than RoBERTa's vocabulary. This model introduced also by Facebook AI handles 100 languages and remains competitive with monolingual counterparts [41, 19].

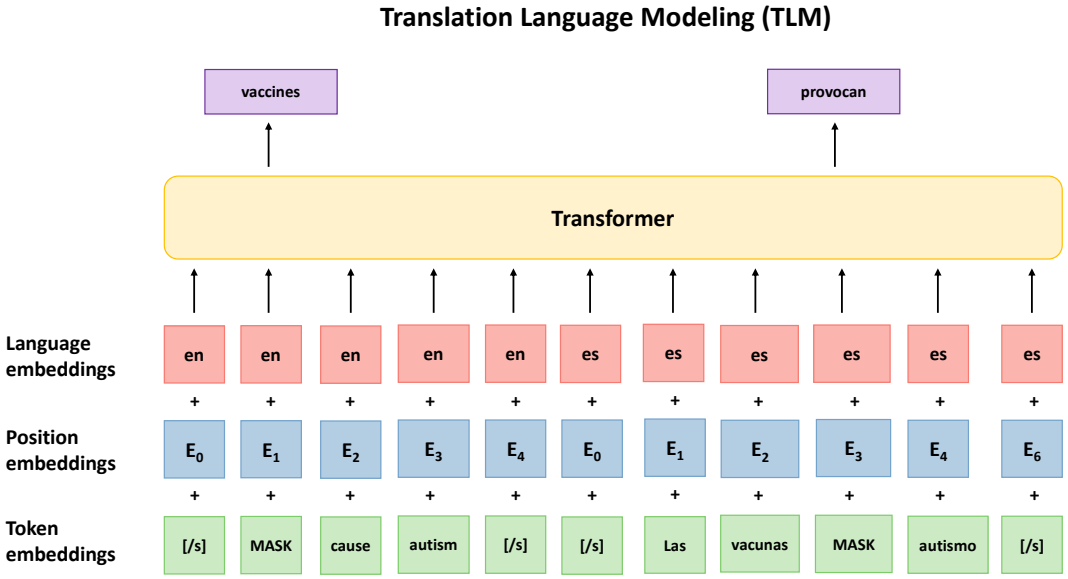


Figure 2.9: Translation Language Modeling (TLM) task visualization from XML pre-training. Position embeddings of the target sentence are reset to facilitate the alignment. Adapted from [41].

Multilingual Universal Sentence Encoder (mUSE) [14] is a dual-encoder transformer architecture. It was trained in a multi-task on Stanford Natural Language Inference (SNLI) corpus [9] and on over billion question-answer pairs from popular forums and QA websites (e.g., Reddit, StackOverflow). As explained in [63], in order to align the cross-lingual vector spaces, mUSE used a translation ranking task. Given a translation pair (s_i, t_i) and various incorrect alternatives translations, identify the correct translation. To improve the performance, the alternatives translations are *hard negatives*, i.e incorrect translations that have a high similarity to the correct translation [63].

Language-agnostic BERT Sentence Embedding (LaBSE) [27] was trained similar to mUSE with a dual-encoder transformer architecture based on BERT with 6 Billion translation pairs for 109 languages.

2.3.7 Knowledge Distillation

It is an undeniable fact that transferring the knowledge from monolingual models trained in high-resource languages and fine-tuned on encoding meaningful sentence embeddings to medium

or low-resources languages is an appealing strategy. For this purpose, Knowledge Distillation can be used.

Knowledge Distillation is a compression technique employed to obtain smaller, faster and lighter variants of large-scale state-of-the-art models, which keep getting larger and larger. This compression technique was introduced by Bucila et al. [10] and generalized by Hinton et al. [32]. The original idea behind this technique is to train a small model (called the student) in reproducing the behaviour of a larger model (called the teacher).

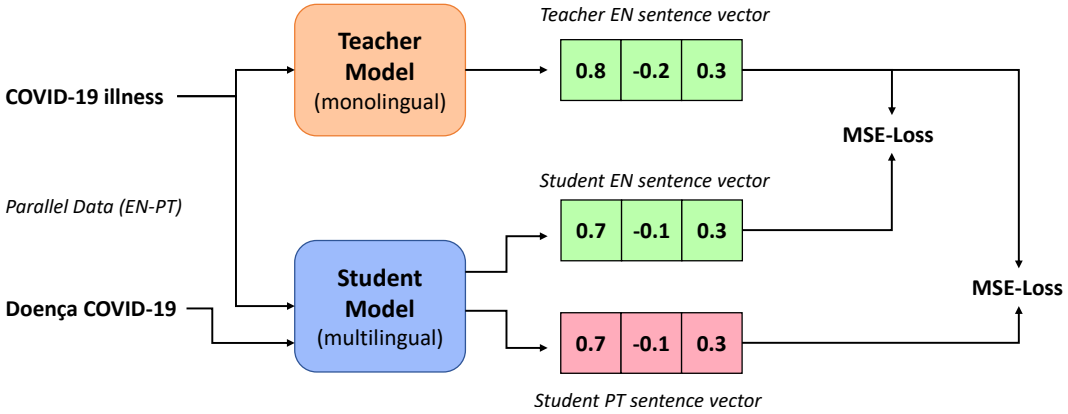


Figure 2.10: Given parallel data (e.g. English and Portuguese), train the student model such that the produced vectors for the English and German sentences are close to the teacher English sentence vector. The multilingual student model imitates the teacher model and achieves by this a high performance. Adapted from [63].

Knowledge Distillation is an easy and efficient method to extend existing sentence embedding models to new languages [63]. Reimers and Gurevych [63], used it to create multilingal models from monolingual models. According to their research, the training is based on the idea that a translated sentence should be mapped to the same location in the vector space as the original sentence. They use the original model (monolingual teacher model) to generate sentence embeddings for the source language and then train a new system (multilingual student model) on translated sentences to mimic the original model. Figure 2.10 depicts the use of Knowledge Distillation technique to obtain multilingual sentence embeddings.

2.4 Benchmark Datasets

Gain an independent perspective about how well a model performs compared to other models is necessary. For that purpose, Benchmark Datasets are used. Cases in point are the STS Benchmark [12], SentEval [17] and TREC-COVID [66].

In Semantic Textual Similarity (STS) tasks the systems need to compute how similar two sentences are, returning a similarity score between 0 and 5. STS Benchmark[1] comprises a selection of the English datasets used in the STS tasks between 2012 and 2017.

SentEval[2] is a library for evaluating the quality of sentence embeddings. SentEval assess the sentence embeddings generalization power by using them as features on a broad and diverse set of downstream tasks, including binary and multi-class classification, entailment, semantic relatedness, and paraphrase detection [18]. STS Benchmark is included in these tasks. SentEval

[1]Semantic Textual Similarity Benchmark: http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark
[2]SentEval github: https://github.com/facebookresearch/SentEval

also includes a series of "probing tasks" to evaluate what linguistic properties are encoded in your sentence embeddings such as verb tense prediction or word content analysis.

TREC-COVID[3] is an information retrieval (IR) shared task initiated to support clinicians and clinical research during the COVID-19 pandemic and combat misinformation. The goal of this dataset is to evaulate how a model ranks a set of biomedical documents according to the relevance with a certain topic. The document dataset to rank is the COVID-19 Open Research Dataset (CORD-19) [81]. This is a collection of biomedical literature articles related to COVID-19 that is updated regularly.

2.5 NLP, Misinformation and COVID-19

In recent years there has been growing interest in fighting misinformation, and fake news spread with NLP techniques, especially since the US election in 2016 [4]. The misinformation problem has become even more critical during the COVID-19 pandemic. Since COVID-19 emerged in Wuhan, China, in December 2019 the public have been bombarded with vast quantities of information, much of which is not checked [55]. There is no doubt that public health emergencies are stressful times for people and communities. The abundance of information on social media frequently without any check on its authenticity makes it difficult for an individual to distinguish between what are facts, and what are opinions, propaganda or biases [55].

The first time a model based on BERT was used in the fake news detection field was in [36]. The authors conclude that fine-tuning the model in the specific task leads to better results than traditional approaches, such as using a simple classifier model based on TF-IDF and cosine similarity to classify fake news [65]. Furthermore, some authors [51] use propagation features to detect fake news on Twitter, and their results reveal that real news is significantly bigger, is spread by users with more followers and fewer followings, and is actively spread on Twitter for a more extended time than fake news. Finally, several tools and data resources for fighting this infodemic have already been developed, such as Fact-Check Explorer from Google[4], or the new COVID-19 explorer from AI2's Semantic Scholar [5]. Research has focused on automatically fact-checking rather than using NLP techniques to help manual fact-checking counter misinformation. There is still considerable uncertainty with regard to using a fully automatic data-driven decision-making algorithm to establish what is fake and true [5]. The additional problem is that fake news and rumours are highly changeable and unpredictable, so fact-checking approaches have to be dynamic [42].

Vijjali et al. [79], proposed a different approach using a two stage automated pipeline, where the first step consists of retrieving the most relevant facts from a COVID-19 claim database, and analysing the entailment between the claim and the true facts as second steps. In our opinion, this method could be used in countering COVID-19 infodemic avoiding the drawbacks of fully automatic classification approaches. However, one of the main issues in our knowledge of Vijjali et al. approach is a lack of multilingualism and dynamic data. Therefore, our research aimed to extend current techniques into different languages.

One of the recommendations from the expert to fight the infomedic and identify fake news is to consult fact-checking sites. A case in point is The International Fact-Checking Network (IFCN) at the Poynter Institute[6]. IFCN unites more than 100 fact-checkers around the world in publishing, sharing and translating facts surrounding the new coronavirus. Our belief is that

[3]TREC-COVID guidelines: https://ir.nist.gov/covidSubmit/index.html
[4]Google Fact-Check: https://toolbox.google.com/factcheck/explorer
[5]AI2's Semantic Scholar COVID-19 literature explorer: https://cord-19.apps.allenai.org/
[6]Poynter official site: https://www.poynter.org/coronavirusfactsalliance/

the first step to counter misinformation should be accessing to checked information. As it was mentioned above, this information should be accessible across all the globe and languages spoken should not be a hurdle. Consequently, in this study we aimed to elucidate the combination of multilingual state-of-the-art deep learning models based on semantic level and NLU techniques with ensemble and dimensionality reduction methods to extract fact-checked text semantically similar to a query and expand interpretability.

Language is the source of misunderstandings.

Antoine de Saint-Exupéry

3 Methodology

This section describes all the steps and methods used for the experiments carried out in this project. For the sake of simplicity, Figure 3.1 depicts how the project is structured.

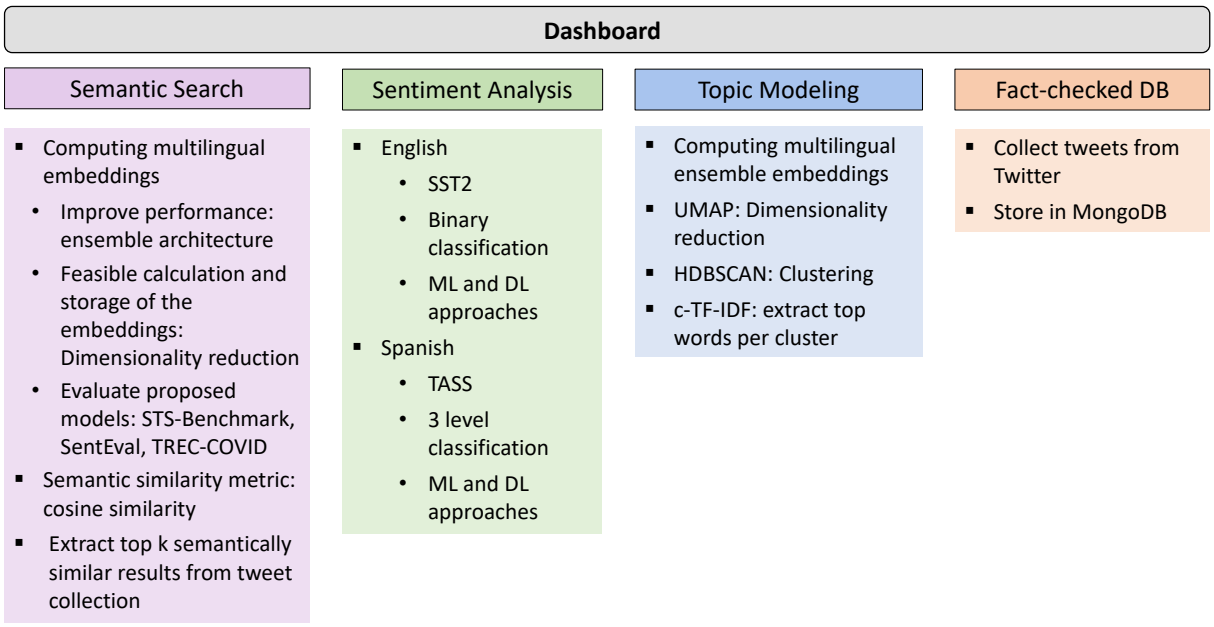


Figure 3.1: Structure and methodology applied in the experiments carried out. This project focuses on countering COVID-19 misinformation through semantic similarity using a collection of fact-check tweets and interpretability using sentiment and topic analysis.

Combining models, also known as ensemble methods, are among the most well-known and widely used techniques in Machine Learning field [6]. Two illuminating examples are Random Forest and Adaboost. The ensemble methods aim to improve performance by combining multiple models to enhance robustness over a single estimator [6].

Much work on the potential of ensemble methods has been carried out for word embeddings [74, 84]. These studies have found that the ensemble approach has some advantages compared with the single approaches. Firstly, ensemble embeddings enhance the embeddings' represen-

tation, leading to more robust embeddings and better performance than single embeddings. Secondly, ensemble methods have the added advantage of increasing vocabulary coverage. However, as shown in [84], it does not mean the more models, the better. Whether including a model to the set helps depends on the complementarity among the models. Consequently, exploring different combinations is needed.

To the best of our knowledge, no other authors have analyzed the effects in semantic similarity of developing an ensemble method based on multilingual sentence transformer models trained with a siamese architecture. Following this approach, this project focuses on countering COVID-19 misinformation applying semantic similarity between two texts: a user claim and a tweet from an official fact-checker verified by the International Fact-Checking Network of the Poynter Institute. We also analyze the effects of dimensionality reduction to improve the similarity calculation and storage of the embeddings.

Furthermore, to countering misinformation the project also pursues interpretability. To fulfill this goal, sentiment and topic analysis are included in the project. It is an undeniable fact that an important part of our behavior is influenced by what other people think [58]. Sentiment analysis can be employed to seek out and understand others' opinions and be aware of the polarity of the information people receive. Finally, topic modeling gives us an overall picture of the thematic organization of our fact-checked tweets collection, gaining insights into how our database is structured.

3.1 Semantic similarity: Sentence Transformers

In this project, to wage COVID-19 misinformation the multilingual SentenceTransformers[1] models [63] are used. SentenceTransformers is a Python framework or set of models for state-of-the-art sentence and text embeddings. These models are transformer-based models (BERT, RoBERTa) which are fine-tuned specifically for various task such as Semantic textual similarity. To derive semantically meaningful sentence embeddings the models consist of a modification of the BERT network using siamese networks, also called bi-encoders. As explained in 2.3.5, bi-encoders, unlike cross-encoders, make feasible large-scale semantic similarity tasks.

The multilingual SentenceTransformers models used are:

- ***distiluse-base-multilingual-cased***: Multilingual knowledge distilled version of multilingual Universal Sentence Encoder (mUSE).
- ***xlm-r-distilroberta-base-paraphrase-v1***: Multilingual knowledge distilled version of RoBERTa trained on large scale paraphrase data.
- ***xlm-r-bert-base-nli-stsb-mean-tokens***: Multilingual version of knowledge distilled BERT version trained in Natural Language Inference (NLI) and Semantic Textual Similarity benchmark (STSb).
- ***LaBSE***: Language-agnostic BERT Sentence Embedding.
- ***distilbert-multilingual-nli-stsb-quora-ranking***: Multilingual version of knowledge distilled BERT version first tuned on NLI and STSb data, then fine-tune for Quora Duplicate Questions detection retrieval.

[1] SentenceTransformwer web page: https://www.sbert.net/index.html

3.1.1 Ensemble method

Ensemble methods are techniques that combine several models to produce improved results. In this work, we analyse the effects of concatenating embeddings from all the combinations of the multilingual models mentioned above. For the sake of explainability, Figure 3.2a depicts the approach proposed in this project. An additional future step could be developing a lighter model that performs as the ensemble methods using the teacher-student strategy 3.2b.

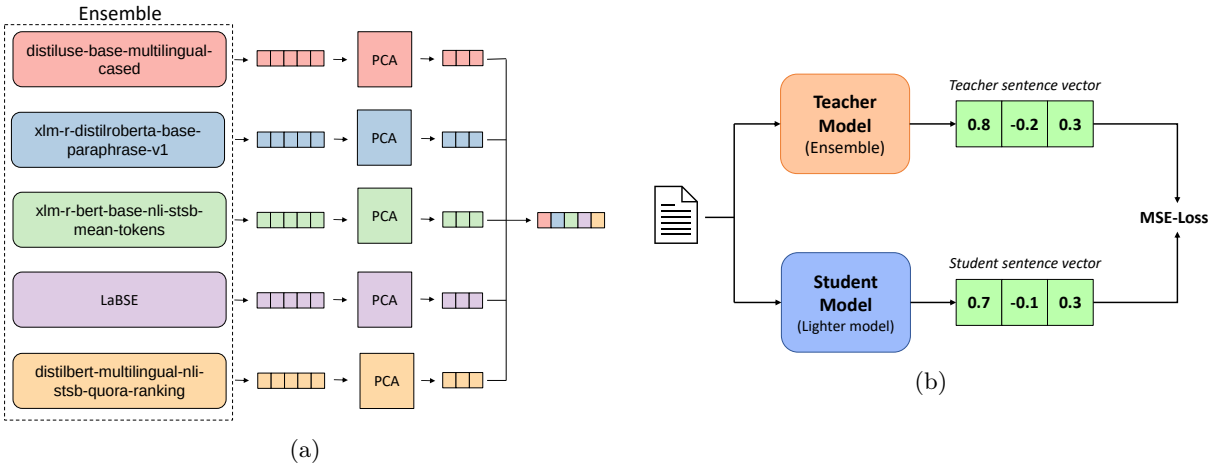


Figure 3.2: Ensemble and dimensionality reduction approach proposed. (a) Concatenation of embeddings from five multilingual sentence-transformers models applying PCA dimensionality reduction. (b) Future work of developing a distilled version of the ensemble models using the teacher-student strategy.

3.1.2 Semantic Textual Similarity metric: Cosine similarity

In order to apply multilingual models to fight against COVID-19 infodemic, we propose using semantic similarity between two texts: a user query and a tweet from a collection gathered by an official fact-checker verified by the International Fact-Checking Network of the Poynter Institute. To measure the semantic similarity between the texts, cosine similarity function is used. Cosine similarity metric takes advantage of the text representation as a vector in high-dimensional space to compute the concurrence between texts, which depict their semantic similarity. The use of this metric is granted by its wide-broad use across the field [87], and its use as similarity metric during the training of the bi-encoder models (Figure 2.7).

The cosine similarity between the two sentence embeddings u and v is a variant of the inner product of the vectors normalised by the vectors' L2 norms, as shown in equation 3.1. Where N represents the number of dimension of the sentence embeddings u and v, $\langle u, v \rangle$ is the inner product between the two vectors, and $\|.\|$ is the L2 norm.

$$CosSim(u, v) = \frac{\sum_{i=1}^{N} u_i v_i}{\sqrt{\sum_{i=1}^{N} u_i^2}\sqrt{\sum_{i=1}^{N} v_i^2}} = \frac{\langle u, v \rangle}{\|u\| \, \|v\|} \tag{3.1}$$

Cosine similarity is symmetric and bounds the inner product between -1 and 1. It has an interpretation as the cosine of the angle between u and v. A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match. The closer the cosine value to 1, the smaller the angle and the greater the match between vectors. In this way, we can retrieve a tweet semantically meaningful for a user query.

3.1.3 Dimensionality Reduction

As explained in [71], cosine similarity applied to a pair of N-dimensional vectors has both time and memory complexity $O(N)$. That is, time and memory grow linearly with the number of dimensions of the vectors compared. The pre-trained multilingual models output embeddings with size 768, except for *distiluse-base-multilingual-cased* with 512 dimensions. Consequently, the ensemble method proposed above compromises the feasibility of using it on semantic search. The reason for this is that the concatenation of embeddings leads to a vectorial representation of higher dimensions.

To solve this problem, the Principle Component Analysis (PCA) is computed and applied for each multilingual model. PCA is a data transformation and dimensionality reduction method. Data transformation aims to transform the original feature space of data into another space with better properties. It is typically combined with dimensionality reduction, so the dimensionality of the transformed space is smaller. PCA finds a subspace that explains most of the data variance. There are both advantages and disadvantages to the use of PCA.

On the one hand, the new PCA feature space has attractive properties, such as removing linear correlation between dimensions and those dimensions with low variance can be considered irrelevant. On the other hand, PCA is an unsupervised method that does not guarantee that the new feature space will be the most appropriate for a supervised task. To cope with this disadvantage, a wide range of number of principal components are analysed, and the best PCA space is selected according to the STS Benchmark performance.

It cannot be ignored the fact that the application of PCA as a dimensionality reduction technique neither improves the runtime, nor the memory requirement for running the models. It only diminishes the needed space to store embeddings and increases the speed to compute the cosine similarity between them.

3.2 Data used for Semantic Search

3.2.1 Multilingual Data for Dimensionality Reduction

In order to fit a PCA for each multilingual model, we use parallel data for 15 languages[2]. The parallel data has been collected from:

- **TED2020**: This dataset contains a crawl of nearly 4000 TED[3] and TED-X transcripts from July 2020. The transcripts have been translated by a global community of volunteers[4] to more than 100 languages.
- **WikiMatrix**: Mined sentences from Wikipedia in different languages [70].
- **OPUS-NewsCommentary**: A parallel corpus of News Commentaries provided by Workshop on Statistical Machine Translation (WMT) [76].

The training data for PCA is composed of 1000 sentences per language. It is worth to point out that OPUS-NewsCommentary is the type of data most related to fact-checked news we hope to face up during COVID-19 infodemic. Thus, half of the train sentences belong to this dataset, and the rest are equally represented by TED2020 and WikiMatrix. Nevertheless, the absence of

[2] ar, cs, de, en, es, fr, hi, it, ja, nl, pl, pt, ru, tr, zh. See Glossary 5.4 for more information
[3] TED2020: https://www.ted.com/
[4] TED Translators https://www.ted.com/participate/translate

data in OPUS-NewsCommentary for Polish, Turkish and Hindi is supplied with TED2020 and WikiMatrix data. The selection of the 15 languages is geared towards the languages available for the OPUS-NewsCommentary data. Nevertheless, the multilingual models support more languages [63]. The composition of training data is depicted in the Figures webpage. Moreover, the code used for this purpose is publicly available[5].

3.2.2 STS Benchmark for PCA model selection

The process of computing and selecting a PCA for each model has three steps. Firstly, PCA is fitted through the embeddings computed by the multilingual models for multilingual data. Secondly, a wide range of number of principal components are analysed, and the best PCA is selected for each model according to the performance of STS Benchmark development set. Finally, STS Benchmark test set is used to evaluate the best PCA selected for each models. The development and test set composition can be checked in Table 3.1.

Genre	***Train***	***Dev***	***Test***	***Total***
news	3299	500	500	4299
caption	2000	625	525	3250
forum	450	375	254	1079
Total	5749	1500	1379	8628

Table 3.1: Semantic Textual Similarity Benchmark breakdown according to genres and train-dev-test splits. Only development and test sets are translated and used in this project.

3.2.2.1 STS Benchmark expanded to new languages: Google Translator

As well as we were concerned about fitting a PCA through multilingual sentences representations, to select a representative PCA among the 15 languages, we translate the STS Benchmark development and test sets from English to the 14 remaining languages. Google Translator python package[6] is used for this purpose. Translated sentence pairs with a confidence value below 0.7 were dropped. The final number of sentence pairs from STS Benchmark development and test set can be interactively visualized in the Figures webpage. As a result, Dutch is the language with the lowest amount of sentence pairs in development (1483 sentence pairs) and test (1358 sentence pairs) sets. As a matter of fact, Google Translator distinguishes two variants from Chinese: simplified and using Mainland Chinese terms (*zh-CN*), and, traditional and using Taiwanese terms (*zh-TW*).

3.2.2.2 PCA model selection criteria

In the development set, the selection criteria of the PCA from each model is based on the Spearman correlation coefficient and the number of principal components.

According to [20], a correlation coefficient is a symmetric, scale-invariant measure of association between two variables. It ranges from -1 to +1, where extremes indicate a perfect negative or positive correlation, respectively, and 0 means no correlation. Pearson (r) and Spearman (ρ) are two of the coefficients used for measuring the correlation between the computed similarity score and the gold score. In its foundations, Semantic Textual Similarity used Person correlation as an evaluation measure. Nevertheless, the scientific community has questioned its usage [62].

[5] Multilingual data for dimensionality reduction repository: https://github.com/Huertas97/Get_Multilingual_Data
[6] Google Translator python package: https://pypi.org/project/google-trans-new/

Pearson correlation is a parametric coefficient rooted in the two-dimensional normal distribution [20]. It is noteworthy that the Pearson correlation coefficient is sensitive to outliers, so it can get severely affected by non-linearities, and the two variables to compare need to be approximately normally distributed [62]. To overcome these limitations, the Spearman's rank correlation coefficient is recommended. The Spearman's rank correlation coefficient does not use the actual values to compute a correlation. Instead, it replaces the observations by their rank and calculates the correlation [20]. It is therefore not sensitive to outliers, non-linear relationships, or non-normally distributed data [62].

For these reasons, the PCA is selected according to the Spearman's rank correlation performance value in STS Benchmark development set. Nevertheless, the Pearson correlation coefficient is also computed to ease comparison with previous models.

The selection criteria are based on selecting the number of components that preserves at least the 99% of the maximum Spearman correlation coefficient scored in STS Benchmark development set and reducing the embeddings below 250 dimensions. In cases where several numbers of components ensure the 99% maximum score, the lowest number of principal components is selected.

3.2.3 Evaluation Benchmarks

Once the PCA number of components is selected for each multilingual model, the performance of the models and the different ensemble combination proposed are evaluated. The datasets employed to assess the models are STS Benchmark, SentEval and TREC-COVID.

For the STS Benchmark test set Spearman correlation coefficient is the metric used to calculate the similarity between the computed scores and the gold scores. STS Benchmark represents how our models capture the semantic similarity between a pair of sentences or short texts (e.g., tweets, abstracts). This is the main characteristic we focus on to countering COVID-19 infodemic.

Not only a good semantic similarity performance is pursued in this project, but also understand as far as possible the embeddings from the multilingual models and the effect of the dimensional reduction. As a consequence, SentEval is used as an evaluation toolkit. This benchmark includes a suite of probing tasks that evaluate what linguistic properties are encoded in sentence embeddings. SentEval is also useful for evaluating sentence embeddings' quality and generalization power by using them as features on a broad and diverse set of transfer tasks, named downstream tasks. As explained in [64], SentEval trains a logistic regression classifier with 10-fold cross-validation setup with the sentence embeddings generated by a model, and use it in the test fold to compute the accuracy for a task. The following probing and downstream task are used in the project:

- TREC: multi-class classification downstream task in information retrieval (IR) based on classify a question in one of the 8 classes according to the type of its answer.
- MRPC: binary classification downstream task where given a pair of sentences, classify them as paraphrases or not paraphrases
- SUBJ: binary classification downstream task for detecting subjectivity in a text.
- SICK-R: downstream task based on predicting the degree of relatedness between two sentences.
- SentLen: this is a classification probing task where the goal is to predict the sentence length which has been binned in 6 possible categories.

- Tense: binary classification probing task, based on whether the main verb of the sentence is in the present or past tense.
- SubjNum: binary classification probing task, focusing on the number of the subject of the main clause.
- ObjNum: binary classification task analogous to the one above, but this time focusing on the direct object of the main clause.

The forked GitHub repository of SentEval with the data and code used in this project is publicly available[7].

Even though the semantic similarity is the primary purpose in the project, its application in COVID-19 misinformation is also a goal to fulfil. Therefore, TREC-COVID dataset is used. Although we cannot ignore the slight differences between semantic similarity and recommendation systems, TREC-COVID offers us the opportunity to compare our multilingual models with other strategies that introduce biomedical knowledge to the models. For TREC-COVID evaluation the official metrics reported are normalised discounted cumulative gain at a cut-off rank of 10 (NDCG@10), precision metric computed at a cut-off rank of 5 (P@5), binary preference (Bpref), and mean average precision (MAP).The GitHub repository for multilingual models TREC-COVID evaluation with explanations, the data and code used in this project is publicly available[8].

3.3 Sentiment Analysis

There has been, and there is a great deal of heated debate about how to manage misinformation. However, it is common knowledge that opinion mining allows to evaluate opinions, analyze the feedback of decisions made and help to make future decisions. In this project, we are interested in understanding the ins and outs of misinformation. Consequently, we apply sentiment analysis to evaluate how a text leans towards a sentiment understanding others' opinions and be aware of the polarity of the information people receive

3.3.1 Datasets

The Sentiment Analysis is applied for English and Spanish languages. The datasets used for these purpose are the Stanford Sentiment Treebank with binary labels (SST2) [72] and *Taller de Análisis Semántico en la SEPLN* (TASS) [69] datasets, respectively.

The SST2 dataset contains a total of 70k sentences divided into train, development and test sets. The sentences are tagged as Positive or Negative, so it consists of a binary classification task. On the other hand, the TASS dataset used is a compilation of tweets from TASS competitions celebrated from 2012 to 2019 with a total of 53k tweets. TASS includes tweets from various topics (TV, politics, sports) from different Spanish speaking countries (Spain, Costa Rica, Uruguay, Mexico and Peru), where tweets are labelled as Positive, Negative or Neutral. Therefore, it is a multi-class classification task. TASS tweets are split into train, development and test sets in a stratified way, maintaining the same distribution of labels in all the sets. The Table 3.2 shows the breakdown of SST2 and TASS datasets according to train-dev-test split and sentiment labels.

[7]SentEval forked GitHub repository: https://github.com/Huertas97/SentEval
[8]TREC-COVID evaluation repositroy: https://github.com/Huertas97/TREC_COVID_sentence_transformers

Dataset	*Split*	*Negative*	*Positive*	*Neutral*	*Total*
SST2	Train	29780	35769	-	67349
	Dev	428	444	-	872
	Test	912	909	-	1821
TASS	Train	20672	26032	3673	50377
	Dev	422	532	75	1029
	Test	1111	1398	197	2076

Table 3.2: SST2 and TASS datasets breakdown according to sentiments and train-dev-test splits used for Sentiment Analysis

3.3.2 Building and Evaluating Machine Learning and Deep Learning models

To begin with Sentiment Analysis, the data is preprocessed and cleaned. Firstly, emojis not related to emotions or feelings are deleted from the text, but emojis related to emotions are converted into text. Secondly, URLs and tweet mentions are removed. Finally, only those tweets from TASS datasets with a level of agreement for the sentiment label are selected.

Several models from both Machine Learning and Deep Learning are developed. In both datasets, the Machine Learning classifiers explored are Naïve Bayes (NB), K-Nearest Neighbors (KNN), Logistic Regression (LR) and Random Forest (RF). A further preprocessing step is added for ML algorithms, the sentences are tokenized and lemmatized to facilitate the word embedding computation. The terms "token" and "lemma" are generally understood as smaller units present in a text string (i.e, words) and the base form of a word, respectively.

For SST2, the English text data is vectorized following two strategies. In the first one, we train the Word2Vec algorithm from Gensim library [61] combined with the TF-IDF over the SST2 corpus for computing word embeddings. On the other hand, the second strategy uses pre-computed word embeddings from Gensim library alongside TF-IDF. These pre-computed Gensim word embeddings come from a Word2Vec model trained on part of the Google News dataset, covering approximately 3 million words and phrases. In both cases, to represent a sentence into a vector, we take the average of all the word vectors in a sentence. Thus, the average vector represent the sentence embedding.

In the matter of TASS, Spanish preprocessed text data is vectorized using word embeddings pre-computed with FastText method. FastText [8] is an improved version of Word2Vec technique where word representations includes character-level and n-grams information. The Spanish word embeddings were computed by Jorge Pérez from Universidad de Chile using Spanish Billion Word Corpus (SBWC) [11] as text data. SBWC corpus compiles text from resources such as Spanish Wikipedia, Wikisource and Wikibooks, Spanish portion of the Europarl (European Parliament), Spanish portion of the Ancora Corpus and Spanish portion of several OPUS Projects.

Regarding the Deep Learning models, all the models used are Transformer-based models from Hugging Face Transformers library [83]. For the English SST2 binary classification task, the Transformer-based model explored are XLM-RoBERTa base size, DistilBert multilingual cased base size, DistilRoBERTa base size, and DistilBert base size fine-tuned for NLI and STS Benchmark tasks. In relation to the Spanish TASS multi-class classification task, the models used are XLM-RoBERTa base size, DistilBert multilingual cased base size, DistilBert multilingual base size fine-tuned for NLI, STS Benchmark and Quora Duplicate Questions detection, and the Spanish version of BERT (BETO) uncased.

Throughout developing Transformer-based classification models, hyperparameters such as learning rate, batch size, number of epochs, weight decay, optimizer's scheduler or gradient

accumulation steps are optimized. To effectively tune hyperparameters, grid search and Bayesian hyperparameter tuning search methods are used. The best hyperparameters values for each model are picked according to the loss of the development set. Regarding the Machine Learning algorithms, the hyperparameters such as the number of neighbours for KNN or the number of Tree Classifiers and their depth for RF are optimized using grid search alongside cross-validation. Finally, the best model for each dataset is selected according to classification metrics.

The loss function is Binary Cross Entropy with logits in SST2, and Categorical Cross Entropy with logits in TASS. According to the official leader board, the metric used for SST2 is accuracy. For TASS dataset, the Matthews correlation coefficient (MCC), macro-averaged F1-score and Cohen's kappa coefficient (κ) are the classification metrics used.

3.4 Topic Modeling

Topic modeling is a textual analysis technique used for discovering the hidden thematic structure in a collection of documents [25]. Topic models gather documents into a set of interpretable topics, where each topic embodies a group of words associated under a single theme.

In an effort to apply semantic meaningful sentence embeddings for topic modeling in COVID-19 field, the procedure used in this project is based on BERTopic [31]. BERTopic is a topic modeling technique with three main steps, (1) extract document embeddings, (2) cluster them to create groups of similar documents with UMAP and HDBSCAN, and (3) extract topics by getting the most important words per cluster with class-based TF-IDF (c-TF-IDF). All the parameters used for these steps are reported in Appendix B.2. Taking advantage of this procedure, we apply our ensemble architecture and dimensionality reduced multilingual models in the first step to compute the fact-checked tweet embeddings for topic modeling.

Due to the fact that the tweets are highly unstructured textual data, we include a preprocessing step before computing the ensemble embeddings and c-TF-IDF. Before computing the ensemble embeddings mentions, URLs, emojis and emoticons are removed; hashtags and accents are not modified; and English contractions are expanded. On the other hand, the preprocessing for c-TF-IDF also removes numbers and multilingual stopwords, and applies lemmatization.

Unlike simply and widely used topic modelling techniques, such as LDA [7], the approach suggested in this work includes semantic (sentence structure matters), is dynamic (topics can be updated), the number of topics needn't be selected ahead and manages short texts (e.g.., tweets). Moreover, one important aspect is that this topic modeling technique is unsupervised. Thus, ensemble multilingual models might reveal topics that manual or supervised modeling might not otherwise detect.

3.5 Dashboard

Dashboards are user interfaces (UIs) that visualize data in an organized manner. Building a dashboard allows deploying the models developed in the sections above to counter COVID-19 misinformation. The dashboard has been carried out using Dash [33].

The dashboard for this project is shown in this video. The dashboard helps users contrast information about the COVID-19 extracting a selected number of the most semantically related fact-checked news to an introduced claim. Furthermore, information about the sentiment polarity and the topic of the request are reported.

3.5.1 Tweets collection

The database of fact-checked news used in the dashboard for semantic search and topic analysis is extracted from Twitter using Tweepy. Tweets are extracted since October 1, 2020 and updated daily until January 20, 2021. The dehydrated tweets collected and the code used is available in GitHub[9].

To ensure the quality of the tweets extracted, we only use tweets from Fact-Checkers recognised by The International Fact-Checking Network (IFCN) at the Poynter Institute. The Fact-Checker Twitter accounts used are depicted in Appendix A. As a matter of fact, during the tweets extraction a language filter is applied by extracting only tweets from Fact-Checkers within the 15 languages used for the PCA. After the extraction, tweets that contain one of the following keywords are selected. For the sake of simplicity, keywords are only shown in English: *coronavirus, virus, covid, sars, disease, ncov, immunity, corona, pneumonia, wuhan, health, isolating, mythm antibody, antigen, pcr, remedy, curfew, infection, lockdown, quarentine, outbreak, distancing, mask, vaccine, fake*. Finally, the database of fact-checked tweets is composed of 65k COVID-19 related tweets.

[9]Tweets collection repository: https://github.com/Huertas97/tweets_collection

I know all those words, but that sentence makes no sense to me.

Matt Groening

4 Experiments and Discussion

This chapter describes the experiments carried out and their discussion. The first section belongs to the experiments related to applying ensemble method and dimensionality reduction to semantic similarity. The second examines the Machine Learning, and Deep Learning approaches to develop a polarity classifier for English and Spanish languages. In the third section, the topic modelling of the fact-checked tweets database is depicted.

4.1 Semantic Similarity

The aim of our work is to evaluate the effects of concatenating the embeddings from all the combinations of the 5 multilingual models mentioned in 3.1 in COVID-19 field. Firstly, we evaluate the single multilingual models performance. The results showed in Table 4.1 establish the performance baseline that we aim to improve using the power of ensemble methodology.

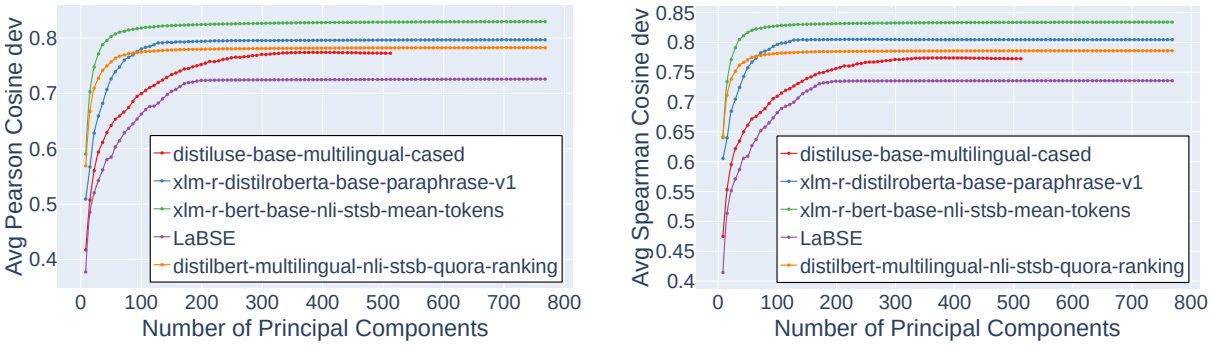
Model	Dimensions	EN-EN		EN-ES		ES-ES		Avg	
		r	ρ	r	ρ	r	ρ	r	ρ
distiluse-base-multilingual-cased	512	81.68	80.75	77.61	75.89	77.49	75.95	75.67	74.19
xlm-r-distilroberta-base-paraphrase-v1	768	83.55	83.50	80.36	80.72	80.65	79.65	79.44	78.91
xlm-r-bert-base-nli-stsb-mean-tokens	768	**83.79**	**85.04**	**81.57**	**82.99**	**81.70**	**82.36**	**80.01**	**80.77**
LaBSE	768	72.69	72.25	71.03	72.15	72.56	70.50	71.37	70.80
distilbert-multilingual-nli-stsb-quora-ranking	768	74.85	78.66	70.66	74.57	71.16	73.01	69.41	72.14

Table 4.1: Spearman ρ and Pearson r correlation coefficient between the sentence representation from single multilingual models and the gold labels for STS Benchmark test set.

The main pitfall of applying ensemble models on semantic search is that it compromises its feasibility due to the high dimensionality of the embeddings. To solve this problem, the Principle Component Analysis (PCA) is computed and applied for each multilingual model. The cumulative explained variance percentage as a function of the number of components from the parallel multilingual train data is available at Appendix A (Figure A.1).

Figure 4.1a and 4.1b show the average Pearson Correlation Coefficient (r) and average Spearman Correlation Coefficient (ρ) using cosine similarity for the 15 languages as a function of the

number of components from the STS-Benchmark development set. It is worth noting that not only STS Benchmark development and test sets are translated into 15 languages, as described in 3.2.2.1 (using Google Translator), but also the languages are combined into monolingual and cross-lingual tasks, giving a total of 31 tasks. Monolingual tasks have both sentences from the same language source (e.g., ar-ar, es-es), while cross-lingual tasks have two sentences, each in a different language being one of them English (e.g., en-ar, en-es).



(a) (b)

Figure 4.1: Average Pearson Correlation Coefficient (a) and average Spearman Correlation Coefficient (b) using cosine similarity for the 15 languages as a function of the number of components from the STS-Benchmark development set. Link to interact with the data.

Model	Dimensions	EN-EN		EN-ES		ES-ES		Avg	
		r	ρ	r	ρ	r	ρ	r	ρ
Best ensemble of 2 models	1536	84.94	86.14	82.69	84.07	82.79	83.45	81.25	81.70
Best ensemble of 3 models	2034	85.17	86.63	82.88	84.50	82.82	83.99	81.29	82.53
Best ensemble of 4 models	3072	85.19	86.65	82.90	84.51	82.77	84.01	81.48	82.56
Ensemble of 5 models	3584	**85.21**	**86.66**	**82.92**	**84.53**	**82.78**	**84.02**	**81.50**	**82.57**

(a)

Model	Dimensions	EN-EN		EN-ES		ES-ES		Avg	
		r	ρ	r	ρ	r	ρ	r	ρ
distiluse-base-multilingual-cased + PCA	249	80.64	80.32	76.65	75.51	76.62	75.42	74.80	73.81
xlm-r-distilroberta-base-paraphrase-v1 + PCA	107	81.71	82.29	78.14	78.08	78.79	79.37	77.35	77.36
xlm-r-bert-base-nli-stsb-mean-tokens with PCA	86	82.84	84.43	80.69	81.84	80.85	82.37	79.35	80.33
LaBSE + PCA	171	71.52	71.22	71.33	70.66	73.27	72.51	71.52	70.79
distilbert-multilingual-nli-stsb-quora-ranking + PCA	72	78.20	79.77	74.25	74.33	74.88	75.61	72.97	73.34
Best ensemble of 2 models with PCA	193	84.06	85.46	81.87	82.89	82.01	83.40	80.60	81.42
Best ensemble of 3 models with PCA	265	84.70	86.02	82.45	83.37	82.58	83.88	81.16	81.95
Best ensemble of 4 models with PCA	436	84.72	86.04	82.48	83.39	82.61	83.89	81.19	81.96
Ensemble of 5 models with PCA	685	**84.74**	**86.05**	**82.50**	**83.41**	**82.63**	**83.91**	**81.21**	**81.99**

(b)

Table 4.2: Spearman ρ and Pearson r correlation coefficient between the sentence representation from single and ensemble models (a) without applying and (b) applying PCA, and the gold labels for STS Benchmark test set. Performance is reported by convention as $\rho \times 100$ and r. Best combination of 2 models = xlm-r-distilroberta-base-paraphrase-v1 & xlm-r-bert-base-nli-stsb-mean-tokens. Best combination of 3 models = xlm-r-distilroberta-base-paraphrase-v1 & xlm-r-bert-base-nli-stsb-mean-tokens & distilbert-multilingual-nli-stsb-quora-ranking. Best combination of 4 models = xlm-r-distilroberta-base-paraphrase-v1 & xlm-r-bert-base-nli-stsb-mean-tokens & distilbert-multilingual-nli-stsb-quora-ranking & LaBSE.

Based on the results from Figure 4.1, all the models, except for *distiluse-base-multilingual-*

cased, achieve approximately the best result within 200 components. Following the criteria explained in 3.2.2.2, the dimensionality reduction implemented for each model is reported in Table 4.2. We achieve minimum dimensionality reduction with of 51% and a maximum of 90% for the single models. The results for each language in the STS-Benchmark development set, either cross-lingual (i.e, en-ar) or monolingual (i.e, ar-ar), are available at the Figures webpage.

Conclusively, the evaluation of the PCA selection and the different ensemble combinations on STS Benchmark test set are reported in Table 4.2. For reasons of space, we only report English, Spanish and the average results across all languages. All the results broken down by monolingual and cross-lingual tasks are available at Figures repository. The most striking result to emerge from the data is that PCA transformation and dimensionality reduction can be applied to the models reducing significantly the embedding dimension and slightly decreasing the performance. Remarkably, ensemble architecture contributes to improving the performance of the models on the STS Benchmark. Combining PCA transformation and ensemble architecture clearly has an advantage over the single multilingual models and can be applied to mono and cross-lingual tasks. Furthermore, the score obtained on the official dataset (en-en task) combining both techniques outperforms the official scores from STS Benchmark[1] and place us among the top 40 from the updated leaderboard[2].

4.1.1 Insight into embeddings

In an attempt to understand the logic behind the embeddings generated by the multilingual models, we evaluate the different models showed above on different tasks from SentEval, as explained in 3.2.3. We are aware that SentEval is restricted to the English language. This limitation is evidence of the difficulty of collecting data for multilingual models. Despite this constraint, SentEval can still give an impression of the quality and insight into our models' performance. Moreover, to better understand the impact of ensemble and PCA techniques on models' performance, we have included classical not fine-tuned models as BERT and RoBERTa. Models fine-tuned on biomedical and COVID-19 scientific documents are also included in the evaluation. All these models are freely available at Hugging Face Transformers library [83].

Our experiments (Table 4.3) are in line with previous results [60]. The shreds of evidence we found points to the utility of dimensionality reduction and ensemble techniques for downstream tasks. Evaluation of SentEval tasks shows how these techniques reduce the embedding size while achieving similar or better performance than original embeddings. Take, for example, paraphrasing (MRPC) and relatedness (SICK-R) detection tasks where the ensemble and dimensionality reduction combined achieve the best results. Besides, subjectivity detection capability is recovered and approaches classical not fine-tuned models trained on language modelling. However, PCA dimensionality reduction affects the performance of the models, such as in TREC task. In this task, the ensemble method does not improve the results compared with single models. The two models used in the best ensemble of 2 models (*xlm-r-bert-base-nli-stsb-mean-tokens* and *xlm-r-distilroberta-base-paraphrase-v1*) undergo a great impact when PCA is applied. Only *LaBSE* model improves with PCA.

As expected, PCA dimensionality reduction also affects some linguistic properties such as predicting the sentence length. Nevertheless, in Tense, SubjNum and ObjNum tasks, the results are improved. It can thus be suggested that detecting the tense, subjects and objects present in a sentence are much more useful than predicting the sentence length for semantic similarity.

[1]STS Benchmark official results http://ixa2.si.ehu.eus/stswiki/index.php/STSbenchmark
[2]STS Benchmark leaderboard: https://gluebenchmark.com/leaderboard

Models	*TREC*	*MRPC*	*SUBJ*	*SICK-R*	*SentLen*	*Tense*	*SubjNum*	*ObjNum*
distiluse-base-multilingual-cased	**92.60**	70.14	91.90	80.78	79.04	84.00	78.68	74.70
xlm-r-distilroberta-base-paraphrase-v1	91.20	75.01	92.73	81.07	66.00	87.68	84.23	82.71
xlm-r-bert-base-nli-stsb-mean-tokens	83.40	74.90	92.48	79.94	61.58	85.37	77.11	75.21
LaBSE	90.60	73.97	92.76	79.15	77.21	87.82	**88.08**	**83.58**
distilbert-multilingual-nli-stsb-quora-ranking	82.20	68.64	91.06	80.89	61.18	85.14	79.18	76.13
distiluse-base-multilingual-cased + PCA	90.20	72.17	90.92	79.23	76.03	83.81	77.74	71.18
xlm-r-distilroberta-base-paraphrase-v1 + PCA	77.00	74.78	90.73	78.96	31.86	86.07	77.27	75.78
xlm-r-bert-base-nli-stsb-mean-tokens + PCA	63.60	74.55	89.44	78.94	44.54	75.83	63.74	64.67
LaBSE + PCA	91.20	75.42	91.82	77.13	76.92	87.28	86.03	82.88
distilbert-multilingual-nli-stsb-quora-ranking + PCA	70.00	72.17	87.58	78.49	39.96	77.72	62.98	62.77
Ensemble of 5 models	92.20	76.81	94.00	**82.71**	75.79	87.61	87.95	82.96
Ensemble of 5 models with PCA	90.20	**77.10**	93.74	81.95	76.83	87.59	86.56	82.24
Best ensemble of 2 models	87.80	76.17	93.46	81.98	69.66	86.93	82.94	81.09
Best ensemble of 2 models with PCA	79.00	76.23	92.43	80.92	48.39	86.14	77.32	75.28
BERT-base	89.80	71.65	94.78	73.50	78.29	**88.79**	84.08	80.98
RoBERTa	92.40	73.86	**95.04**	76.84	**79.05**	88.14	85.25	82.33
clinicalcovid-bert-nli (monolingual)	75.80	75.71	88.23	77.49	62.31	83.52	78.34	77.38
scibert-nli (monolingual)	80.60	71.77	88.00	76.42	64.92	82.97	80.72	78.73
biobert-nli (monolingual)	80.20	73.45	87.88	77.39	56.66	81.27	76.85	75.96

Table 4.3: Evaluation of multilingual sentence embeddings using the SentEval toolkit. Fine-tuned biomedical models and classical not fine-tuned models are also evaluated. Tasks are grouped into downstream (first block) and probing (last block) tasks. Best combination of 2 models = xlm-r-distilroberta-base-paraphrase-v1 & xlm-r-bert-base-nli-stsb-mean-tokens.

4.1.2 TREC-COVID: evaluation on scientific biomedical documents field

This project sheds new light on multilingual models' utility fine-tuned for semantic similarity in the COVID-19 misinformation and infodemic field. TREC-COVID is an information retrieval (IR) shared task using biomedical documents from the COVID-19 Open Research Dataset (CORD-19). IR tasks from TREC are based on systems' ability to find relevant articles containing answers to the questions in the topics. As others have highlighted [56, 64, 80], relevance is not the same as semantic similarity. To get the best performance relevance systems do not require a true understanding of the text [80]. Nevertheless, combining a well-known conventional method used in TREC tasks as a baseline, like BM25 Okapi [67], and rankings from semantic Transformer-based models, we can evaluate the performance of neural models. Using this hybrid approach proposed by [56], we could compare the performance of multilingual models and models with biomedical and task-specific training.

In this project, we used a variation of the procedure proposed in [56] to rank CORD-19 documents from TREC-COVID round 1. Documents created before December 31st 2019 (before the first reported case) are removed. The relevance score for a CORD-19 document considering a specific topic ($\psi(T_i, d)$), is calculated using the 4.1 formula. Where z represents the adjusted log-base such that the highest scoring document has a value of nine, $t \in T_i$ represents possible fields of topic T_i (i.e, query, question and narrative), $f \in d$ represents possible facets of the document (i.e, abstract or title), $BM25$ denotes BM25 Okapi scoring algorithm, $e(t), e(f)$ represent the topic field embedding and facet embedding, respectively, and *cos* denotes cosine similarity.

$$\psi(T_i, d) = \log_z(\sum_{t \in T_i} \sum_{f \in d} BM25(t, f)) + \sum_{t \in T_i} \sum_{f \in d} cos(e(t), e(f)) \tag{4.1}$$

Specifically, in our procedure for the BM25 relevance score, we apply the well-known BM25 Okapi algorithm. For the sentence embeddings computing, we apply the multilingual models

Models + BM25 (scaled)	***p@5***	***ndcg@10***	***map***	***bpref***
distiluse-base-multilingual-cased	0.7067	0.6043	0.2268	0.3964
xlm-r-distilroberta-base-paraphrase-v1	0.7200	0.5812	0.2127	0.3854
xlm-r-bert-base-nli-stsb-mean-tokens	0.6267	0.5354	0.1918	0.3732
LaBSE	0.7200	**0.6316**	**0.2433**	0.4036
distilbert-multilingual-nli-stsb-quora-ranking	0.7267	0.6006	0.2312	0.3773
distiluse-base-multilingual-cased + PCA	0.6733	0.5896	0.2230	0.3989
xlm-r-distilroberta-base-paraphrase-v1 + PCA	0.6533	0.5565	0.1994	0.3816
xlm-r-bert-base-nli-stsb-mean-tokens + PCA	0.5800	0.5012	0.1779	0.3671
LaBSE + PCA	**0.7400**	0.6300	0.2373	0.4045
distilbert-multilingual-nli-stsb-quora-ranking + PCA	0.6267	0.5218	0.1839	0.3552
Best ensemble of 2 models	0.6667	0.5425	0.2002	0.3780
Best ensemble of 2 models with PCA	0.6067	0.5147	0.1861	0.3724
Ensemble 5 models	0.7067	0.5874	0.2165	0.3811
Ensemble 5 models + PCA	0.6200	0.5290	0.1907	0.3732
clinicalcovid-bert-nli (monolingual)	**0.7400**	0.6303	0.2309	**0.4074**
scibert-nli (monolingual)	0.6800	0.5861	0.2037	0.3781
biobert-nli (monolingual)	0.7000	0.5923	0.2103	0.3902

Table 4.4: Evaluation of multilingual sentence embeddings using TREC-COVID round 1. Fine-tuned biomedical models are also evaluated. The official metrics reported are normalised discounted cumulative gain at a cut-offrank of 10 (NDCG@10), precision metric computed at a cut-off rank of 5 (P@5), binary preference (Bpref), and mean average precision (MA). Best combination of 2 models = xlm-r-distilroberta-base-paraphrase-v1 & xlm-r-bert-base-nli-stsb-mean-tokens

with the ensemble architecture previously proposed. The official metrics from TREC-COVID evaluation are reported in Table 4.4.

Based on these results, PCA affects all models except *LaBSE*, and more seriously than STS Benchmark. Compared to general biomedical models (*scibert biobert*), single models show similar results. The analysis did not reveal any improvement using the ensemble architecture or PCA dimensionality reduction. A notable exception is the *LaBSE* model. The best result obtained with the model explicitly trained on the CORD-19, *clinicalcovid-bert-nli*, can be equated with *LaBSE* and PCA. This architecture proposed has the advantage of incorporating multilingualism.

Interestingly, in the TREC task from SentEval where the documents were not biomedical, only *LaBSE* with PCA improved its result. Further analysis should be carried out to measure the impact of TREC tasks in the models' performance. Not surprisingly, the PCA dimensionality reduction impacts the results from TREC-COVID task. This is probably due to the PCA selection explicitly based on the result in STS-Benchmark and confirms the difference between IR and semantic similarity tasks. An important point to note is that, among the three biomedical models, only the model specifically trained in CORD-19 outperforms the multilingual models, with the exception of *LaBSE*. This highlights the importance of using task-specific data. Future work will concentrate on include COVID-19 related biomedical data for the improvement of semantic similarity.

4.2 Sentiment Analysis

In this section, we present the results obtained after developing Machine Learning and Deep Learning Transformer-based models for predicting a text's polarity for English and Spanish languages, following the methodology explained in 3.3. All the parameters used for the different vectorization and classification models applied in this section are available at Appendix B.1.

4.2.1 Binary sentiment classification for English texts

As explained in 3.3.2, the Machine Learning Naïve Bayes, K-Nearest Neighbors, Logistic Regression and Random Forest algorithms are used as classifiers. Two strategies are followed to encode SST2 corpus data into vectors that will be used as inputs for the ML algorithms. In the first one, the word embeddings result from the training of Word2Vec over the SST2 corpus data combined with TF-IDF. In the second one, the word embeddings come from an already pre-trained Word2Vec model on part of the Google News dataset. This pre-trained Word2Vec model has the advantage of incorporating much more text data into the training. In both cases, to represent a sentence into a vector, we take the average of all the word vectors in a sentence. Thus, the average vector represents the sentence embedding. These input sentence embeddings are visualized in PCA 3D and T-SNE 3D projections to represent the distribution of SST2 texts in the space (see Figure 4.2). To track and visualize metrics for the ML algorithms and share results we use Weights & Biases. The metrics visualizations for the ML algorithms using our trained Word2Vec and TF-IDF are available at Sklearn SST2 project, and the metrics visualizations using pre-trained Word2Vec and TF-IDF are available at Sklearn SST2 Gensim project.

Model	***Acc***	***MCC***
xlm-roberta-base	91.4	82.94
distilbert-base-multilingual-cased	86.27	72.55
distilroberta-base	**93.14**	**86.36**
distilbert-base-nli-stsb-mean-tokens	91.21	82.51
Word2Vec TF-IDF + NB	58.65	19.35
Word2Vec TF-IDF + KNN	57.77	18.07
Word2Vec TF-IDF + LR	69.41	39.96
Word2Vec TF-IDF + RF	65.35	30.79
Pre-trained Word2Vec TF-IDF - NB	66.611	39.34
Pre-trained Word2Vec TF-IDF - KNN	65.4	31.67
Pre-trained Word2Vec TF-IDF - LR	77.65	56.13
Pre-trained Word2Vec TF-IDF - RF	75.40	52.81

Table 4.5: Test Metrics on SST2, where NB is Naïve Bayes; KNN is K-Nearest Neighbor; LR is Logistic Regression; RF is Random Forest; Acc is Accuracy; and MCC is Matthews Correlation Coefficient (MCC). All the metrics are reported as $metric \times 100$

Regarding the Deep Learning Transformer-based models developed, the models used are trained while minimizing Binary Cross-Entropy loss function value. Besides, during the training hyperparameters are optimized, selecting those with the lowest loss value in the development set. Four Transformer-based models are used, two multilingual (XLM-RoBERTa and DistilBERT multilingual) and two monolingual (DistilRoBERTa and DistilBERT fine-tuned for NLI and STS Benchmark). A total of 80 runs are launched for the hyperparameter optimization. To

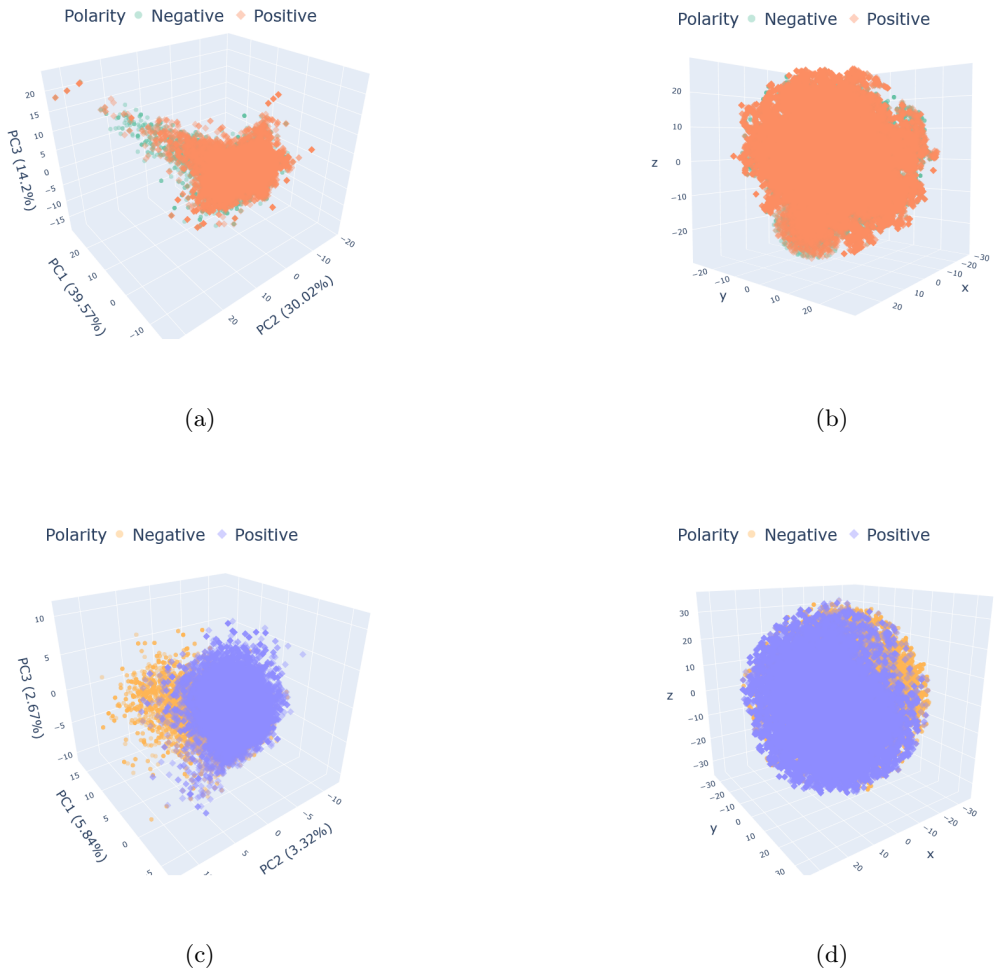
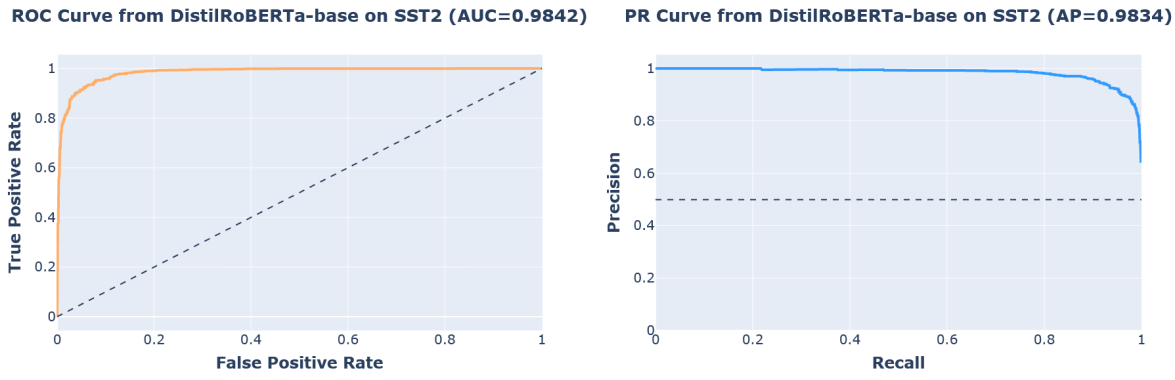


Figure 4.2: Projection of sentence embeddings from SST2 using trained Word2Vec and TF-IDF: PCA 3D (a) and T-SNE 3D (b). Projection of sentence embeddings from SST2 using pre-trained Word2Vec on part of the Google News dataset from Gensim and TF-IDF: PCA 3D (c) and T-SNE 3D (d). Link 1 and Link 2 to interact with the data.

track the hyperparameter optimization experiments, visualize metrics, and share results we use Weights & Biases. The runs for the hyperparameter optimization for each model are logged and available at SST2 DL train project. The best hpyerparameter configuration and the test metrics for each model are available at SST2 DL test project.

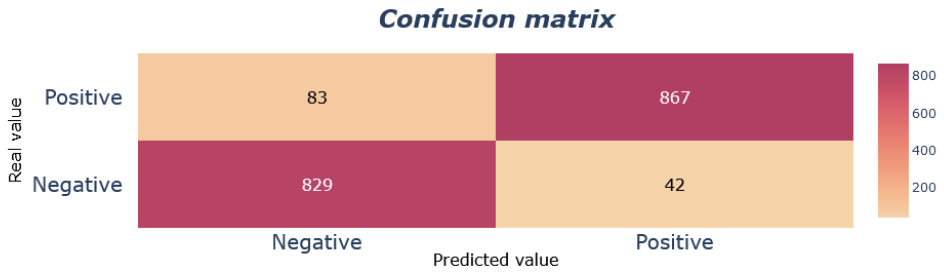
As Table 3.2 shows, the SST2 dataset is balanced, so the accuracy metric is suitable for measuring and comparing the models' classification performance. The accuracy is also the official metric reported to evaluate models performance in the SST2 dataset. Furthermore, the Matthews Correlation Coefficient (MCC) is also computed as a performance indicator. Table 4.5 shows the test metrics for all the different models on SST2. According to the results, all the Deep Learning Transformer-based models outperform the ML algorithms. The DistilRoBERTa model has the best accuracy and MCC values. It is also worth noting that the strategy for encoding SST2 corpus into sentence embeddings affect the results, where all ML algorithms perform better when the SST2 corpus is encoded using the pre-trained Word2Vec. This result proves the major role that plays the vectorization in NLP tasks such as classification.

Another critical point is that distilled versions perform exceptionally well on binary polarity detection. The best model, DistilRoBERTa, is a distilled version of RoBERTa with 82M



(a)

(b)



(c)

Figure 4.3: Visualization test metrics on SST2 for DistilRoBERTa: (a) Receiver operating characteristic (ROC) curve and (b) Precision-Recall curve. We also display the area under curve (AUC) and the baseline as dashed lines; (c) Confusion matrix to evaluate the accuracy of the classification. Link to interact with the data.

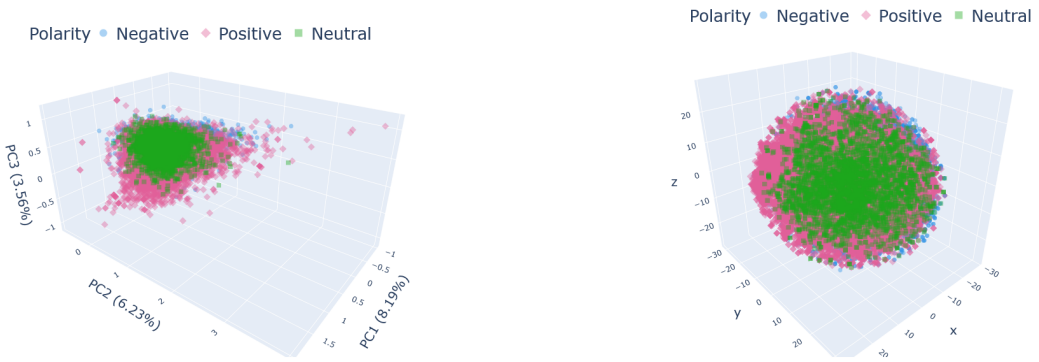
parameters and 331MB size. In contrast, XLM-RoBERTa is a multilingual model with 270M parameters and 1.1GB size. Besides, DistilRoBERTa takes 20 min to train while XLM-RoBERTa takes more than 50 min. Finally, the 93.14 accuracy score is among the top 40 official scores [3]. These results show how lighter, smaller and faster-distilled models can achieve great results. Finally, the distilled version is preferred to be used in the Dashboard to use our computational resources efficiently.

4.2.2 Multi-class sentiment classification for Spanish texts

For Spanish multi-class sentiment classification, we use the same ML algorithms as we did for SST2. To begin with, TASS tweets are preprocessed and cleaned as explained in methodology (see Section 3.3.2). We employ a FastText model pre-trained on Spanish Billion Word Corpus (SBWC) from Universidad de Chile, with 855380 words in the vocabulary, for extracting the Spanish word embeddings. As we did in SST2, we take the average of all the word vectors in a sentence to compute the sentence embedding. The metrics visualizations for the ML algorithms are available at Sklearn TASS project and the 3D projections of the sentence embeddings are shown in Figure 4.4.

Regarding the Deep Learning Transformer-based models, four models are developed. Three of them multilingual (XLM-RoBERTa, DistilBERT multilingual, and DistilBERT multilingual fine-tuned for NLI, STS Benchmark and Quora Ranking), and the monolingual Spanish version of BERT (BETO) uncased. Models are trained while minimizing the Categorical Cross-Entropy

[3] SST2 official leaderboard https://gluebenchmark.com/leaderboard



(a) (b)

Figure 4.4: Projection of sentence embeddings from FastText trained on Spanish Billion Word Corpus: PCA 3D (a) and T-SNE 3D (b). Link to interact with the data.

with logits loss function. As in SST2, during the training hyperparameters are optimized, selecting those with the lowest loss value in the development set. The 115 runs computed for the hyperparameter optimization are logged and released at TASS DL train project. Moreover, the test metrics for each model's best configuration are available at TASS DL test project.

Model	***Acc***	***Macro-F1***	***MCC***	κ
xlm-roberta-base	**84.70**	**63.13**	**71.97**	**71.62**
distilbert-base-multilingual-cased	83.7	62.94	70.15	69.81
distilbert-multilingual-nli-stsb-quora-ranking	84.59	61.13	71.68	71.15
BETO-uncased	82.73	57.28	67.63	67.00
FastText + NB	55.47	47.10	30.39	28.69
FastText + KNN	73.76	51.00	51.87	51.10
FastText + LR	78.05	55.00	59.3	58.78
FastText + RF	75.20	53.00	53.65	52.89

Table 4.6: Test Metrics on TASS, where NB is Naïve Bayes; KNN is K-Nearest Neighbor; LR is Logistic Regression; RF is Random Forest; Acc is Accuracy; Macro-F1 is macro-averaged F1-score; MCC is Matthews Correlation Coefficient (MCC); and κ is Cohen's kappa coefficient. All the metrics are reported as $metric \times 100$.

Unlike SST2 dataset, TASS data is imbalanced (see Table 3.2). Consequently, to evaluate and compare the different classification models, we use the Matthews correlation coefficient (MCC), macro-averaged F1-score, and Cohen's kappa coefficient (κ). In contrast with Accuracy, these metrics can manage imbalanced classes. Macro-F1 score equally weights all classes regardless of the size, so biggest classes have the same importance as small ones have. Consequently, F1 macro is used to report if all classes are properly classified. MCC is a discrete case for the Pearson Correlation Coefficient. It is generally regarded as a balanced measure that considers all the elements from the confusion matrix and can be used even if the classes are very different in size [29, 22]. Finally, κ is a relative metric representing the dependence between the predicted and the true classification. It exploits the dependence obtained by chance between the predicted and the true classification deleting any intrinsic characteristic of the dataset [29]. Thus, we

report it because it allows comparing models applied to different datasets and is similar to MCC in multi-class cases [29].

According to Table 4.6, the Transformer-based models outperforms the ML algorithms combined with FastText embeddings. Based on results presented, the monolingual BETO model does not outperform multilingual models. This shows the importance and usefulness of multilingual models. Finally, unlike in SST2, fine-tuning does not seem to improve DistilBERT multilingual performance multilingual, and macro-F1 shows how the models do not properly classify all the classes. The reason for these results might be justified by the huge difference between Neutral class and the two remaining classes. However, further analysis of the confusion matrix is needed as discussed below (Figure 4.5).

The purpose of this section is to obtain both an accurate and feasible model for the polarity prediction to be used in the dashboard. Therefore, the multilingual DistilBERT fine-tuned for NLI, STS Benchmark and Quora ranking is selected because it is the best according to the MCC together with the size (501 MB with 134M parameters). Based on the test results for the model selected (Figure 4.5), we can conclude that the model poorly classifies Neutral class, but reasonably classifies Negative and Positive classes. The confusion matrix and the area under the Precision-Recall curve (AUPRC) calculated using the average precision (AP) for each class support this idea.

On the other hand, the ROC curve shows an adequate performance classifying the Neutral class. We should sound a note of caution with regard to such findings. In [21] the authors found

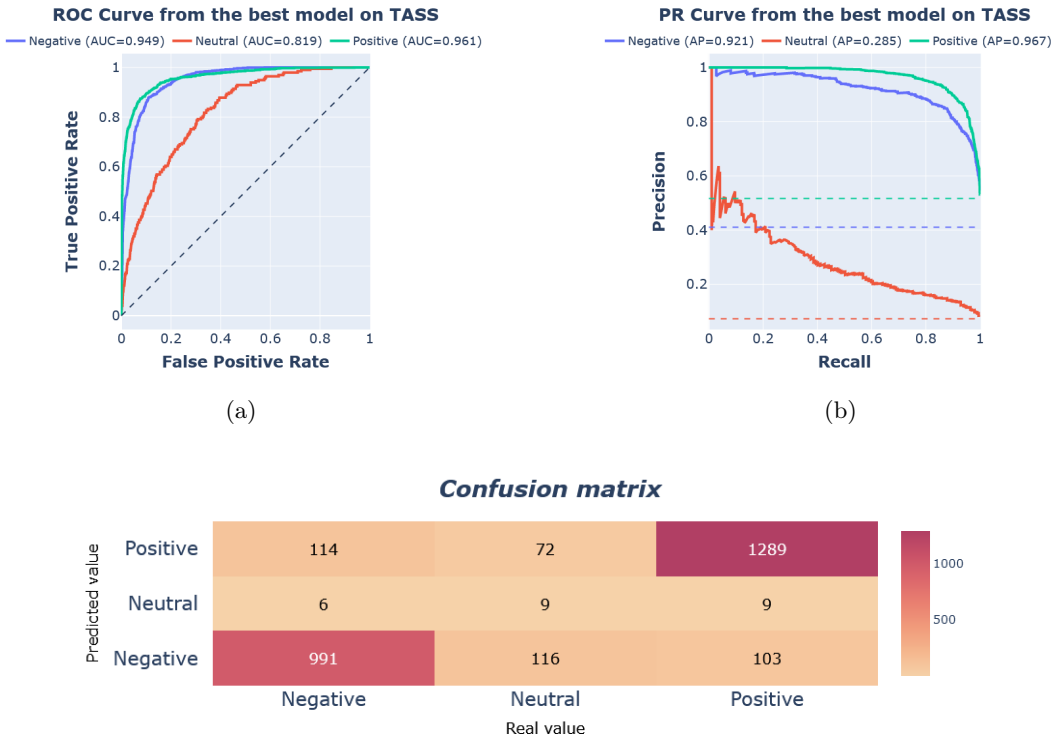
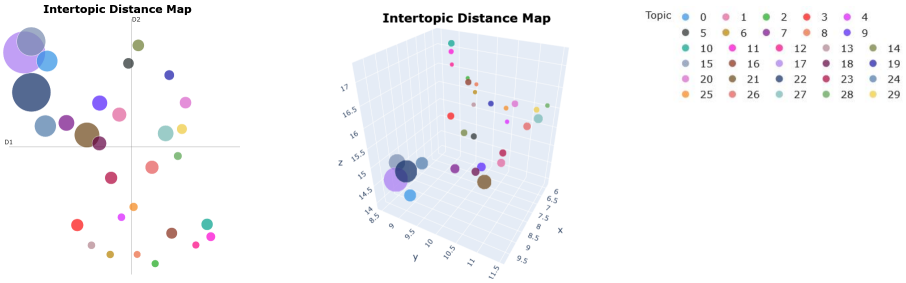


Figure 4.5: Visualization test metrics on TASS for Distilbert-multilingual-nli-stsb-quora-ranking: (a) Receiver operating characteristic (ROC) curve, We also display the area under curve (AUC) to calculate the area under the ROC curve (AUROC). (b) Precision-Recall curve. We also display the average precision metric (AP) to calculate the area under the PR curve (AUPRC). Dashed lines represent baselines for each class; (c) Confusion matrix to evaluate the accuracy of the classification. Link to interact with the data.

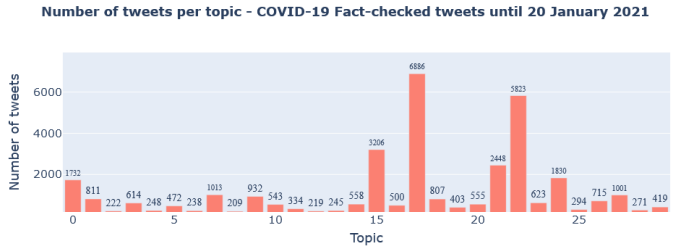
that ROC curves can present an overly optimistic view of an algorithm's performance if there is a large skew in the class distribution. The Neutral class is an illuminating example with a much larger number of negative examples than positive examples (see Table 3.2 and Neutral Class baseline in 4.5). In imbalanced classification tasks, the sample size for each class plays a crucial role in determining the goodness of a classification model. Neutral class presents a 1:13 ratio in TASS dataset. We believe that our results demonstrate the need for introducing more Neutral data instances because as the size of the training set increases, the large error rate caused by the imbalanced class distribution decreases. Other remedies to deal with this imbalanced data could be downsampling or upsampling. Finally, as established in [22], we have also proved that, if the confusion matrix is symmetric, then κ and MCC coincide.

4.3 Topic Modeling

As mentioned previously, interpretability, along with semantic similarity, are pursued in this project to counter COVID-19 misinformation. Topic modeling gives us an overall picture of the thematic organization of our fact-checked tweets collection, gaining insights into how our database is structured. As a matter of fact, using the multilingual ensemble models for the embedding computation step ensures that multilingualism and semantic are considered for the topic modelling, as explained in section 3.4. Furthermore, topic modelling can guide the search of information through the database, since the number of documents per topic is not the same and reveals how much information is available and the most representative words per topic.



(a)



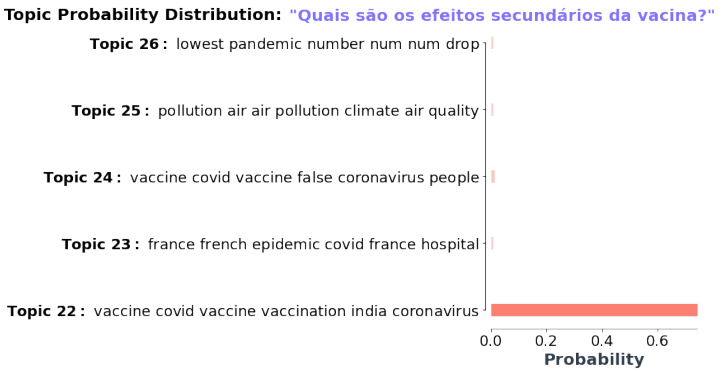
(b)

Figure 4.6: Topic modeling visualization for Fact-checked tweets since October 1, 2020 until January 20, 2021 using multilingual ensemble models xlm-r-distilroberta-base-paraphrase-v1 with PCA and xlm-r-bert-base-nli-stsb-mean-tokens with PCA. Intertopic distance 2D (a) and 3D maps where each circle indicates a topic and its size the frequency of the topic across all tweets. (b) Barplot of the number of tweets per topic. Link to interactively explore topics and the words that describe them.

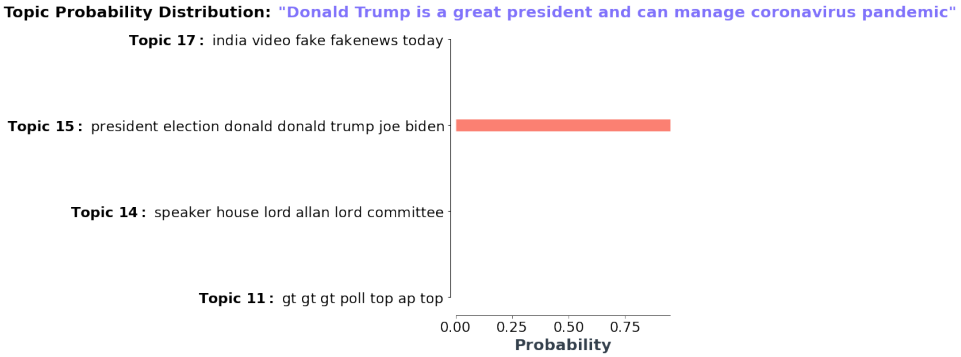
The ensemble model used is selected in connection with the semantic search. Therefore, we chose the combination of xlm-r-distilroberta-base-paraphrase-v1 with PCA and xlm-r-bert-

base-nli-stsb-mean-tokens with PCA because the balance between performance and number of dimensions makes them the most suitable choice for the dashboard application (see 4.2). We highly recommend to analyze interactively the topic modeling results available at Figures repository.

Our fact-checked tweets database, combined with the HDBSCAN clustering algorithm applied over the multilingual ensemble embeddings, with a previous step of dimensionality reduction to 5 dimensions with UMAP, shows how tweets can be organized in 30 topics (see Figure 4.6). The distance between these topics can be employed to assess the similarity between different topics since UMAP keeps a significant portion of the high-dimensional local and global structure in lower dimensionality. It is necessary to point out that forcing tweets in a topic could lead to poor performance and decrease topics quality. This drawback is coped with HDBSCAN. HDBSCAN is a Hierarchical Density-Based Spatial Clustering that detects outliers which do not have a topic assigned. The number of tweets without a topic assigned is 31K. The distribution of the 34K remaining tweets per topic is depicted in Figure 4.6b. In Appendix C.2 the different topics with the top 20 representative words are shown.



(a)



(b)

Figure 4.7: Topic prediction probabilities using the topic modeling with embeddings from multilingual ensemble models xlm-r-distilroberta-base-paraphrase-v1 with PCA and xlm-r-bert-base-nli-stsb-mean-tokens with PCA. (a) Example prediction for "Quais são os efeitos colaterais da vacina?". (b) Example prediction for "Donald Trump is a great president and can manage coronavirus pandemic"

Topic 17 and 22 are the largest clusters with 6886 and 5823 tweets. The top 5 words for each topic are "india, video, fake, fakenews, indiatoday" and "vaccine, covid vaccine, vaccination, india, coronavirus", respectively. It is important to note that our topic modeling fine-grains topics related to COVID-19. Take for example topic 7 ("mask, health, prevent, cure"), topic 3 ("china, laboratory, coronavirus created") or topic 13 ("efficacy, moderna, vaccine candidate").

Moreover, some topics are separated according to geopolitical issues, like topic 4 ("spain, madrid, covid spain"), topic 6 ("brazil, bolsonaro, covid brazil"), topic 15 ("donal trump, joe biden, election") and topic 1 ("germany, corona, infection"). Remarkably, other themes are captured, such as sports, technology and festivities which are embodied respectively in topic 12 ("pba, pba semis, pba finals", where PBA stands from Philippines Basketball Association), topic 16 ("iphone, laptop, apple, launch") and topic 9 ("christmas, thanksgiving, holiday, family").

Even though the number of tweets unassigned to a topic is not ideal, these results show the considerable capability of multilingual ensemble methods to capture a wide variety of themes in the fact-checked tweets collection.

The most striking result of topic modeling is its use to predict a text's topic. Figure 4.7 illustrate this point. The sentence "Donald Trump is a great president and can manage coronavirus pandemic" is assigned to topic 15 with 0.95 probability, and sentence "Quais são os efeitos colaterais da vacina?" is assigned to topic 22 with 0.84 probability. Both sentences not only are assigned to a meaningful topic but with high confidence.

4.4 Dashboard

All in all, according to 4.2, the best ensemble combination for semantic similarity is the ensemble composed of the 5 multilingual models. However, we chose the combination of xlm-r-distilroberta-base-paraphrase-v1 with PCA and xlm-r-bert-base-nli-stsb-mean-tokens with PCA because the balance between performance and number of dimensions makes them the most suitable choice for the dashboard application. The models for the sentiment prediction and topic modeling are also included in the dashboard. Some examples of the dashboard use can be seen in this video.

Those who know nothing of foreign languages know nothing of their own.

Johann Wolfgang von Goethe

5 Conclusion and Future Work

5.1 Conclusions

The conclusions drawn from the work carried out and the results obtained are shown below:

- The evidence from this study suggests that PCA transformation and dimensionality reduction applied to the five multilingual sentence transformer models significantly reduce the embedding dimensions and slightly decrease the performance on semantic similarity tasks. Remarkably, ensemble architecture contributes to improving the performance of the models on the STS Benchmark. Combining PCA transformation and ensemble architecture clearly has an advantage over single multilingual models. It can be applied to mono and cross-lingual tasks.

- Although PCA dimensionality reduction affects some linguistic properties, the shreds of evidence we found according to SentEval downstream tasks points to the utility of dimensionality reduction and ensemble techniques for NLP transfer learning tasks.

- The evaluation on TREC-COVID round 1 did not reveal any improvement using the ensemble architecture or PCA dimensionality reduction. A notable exception was *LaBSE* model which outperforms biomedical models and equates the performance of the model explicitly trained on CORD-19, *clinicalcovid-bert-nli*, both applying and not applying dimensionality reduction.

- Regarding sentiment analysis, we have fine-tuned *distilroberta-base* Transformer-based model for English sentiment analysis with 93.14 accuracy score in SST2, and *distilbert-multilingual-nli-stsb-quora-ranking* for Spanish sentiment analysis with 71.68 MCC score in the compilation of tweets from TASS competitions celebrated from 2012 to 2019. Both models show how lighter, smaller and faster-distilled models can achieve outstanding results. The experiments carried out to obtain these models support the idea that deep learning transformer-based model are state-of-the-art approaches in NLP tasks. Furthermore, we have proved the usefulness of transfer learning using word embeddings from pre-trained models incorporating much more data into the training.

- Our fact-checked tweets database combined with the multilingual topic model developed using the best ensemble combination of 2 models (*xlm-r-distilroberta-base-paraphrase-v1* with PCA and *xlm-r-bert-base-nli-stsb-mean-tokens* with PCA) denotes that we can capture a diversity of topics gaining insights into how our database is structured and explainability. Furthermore, its use to predict a text's topic has been proved.
- Taking advantage of Dash tool, we have developed a dashboard where the data and models mentioned above can be accessible.

In a nutshell, we have proven the excellent results and applicability of multilingualism on semantic similarity. We have improved the results using ensemble methods and reduce the models' dimensionality for accelerating similarity calculations. We have also gained some insight into how the models work and analyzed our multilingual models' capabilities in COVID field. In our view, these results constitute an excellent initial step toward incorporating multilingualism. We believe this solution will assist researchers and people in countering COVID-19 misinformation.

5.2 Future work

Distilled versions from the multilingual ensemble models we have developed will reduce the computation time required to output embeddings regarding semantic similarity search. Another possible solution to explore will be the parallelization of the output embeddings from the ensemble. Moreover, other dimensionality reduction methods should be explored, both supervised and unsupervised, and applying a PCA over the full ensemble method should be compared with the results obtained in this project. Future work should focus on measuring the impact of TREC tasks in the models' performance and enhancing multilingual model performance by fine-tuning them with COVID-19 related biomedical data.

According to sentiment analysis, we hope to extend the polarity analysis to multilingualism. Thus, developing a suitable multilingual dataset is a vital issue for future research. Furthermore, developing models that can fine-grain polarity in more levels is a challenging but worthy project that we encourage to pursue.

The fact-checked tweets database keeps growing and incorporating more data. Therefore, updating topic modelling and semantic search is needed.

Glossary

5.3 General acronyms

- **AI**: Artificial Intelligence
- **ANN**: Artificial Neural Network
- **BERT**: Bidirectional Encoder Representations from Transformers
- **BoW**: Bag-of-Words
- **DL**: Deep Learning
- **DNN**: Deep Neural Network
- **FFNN**: Feedforward Neural Network
- **GRU**: Gated Recurrent Unit networks
- **IR**: Information Retrieval
- **KNN**: K-Nearest Neighbors
- **LR**: Logistic Regression
- **LSTM**: Long Short-Term Memory networks
- **MCC**: Matthews correlation coefficient
- **MIT**: Massachusetts Institute of Technology
- **NB**: Naïve Bayes
- **NLI**: Natural Language Inference
- **NLP**: Natural Language Processing
- **NLU**: Natural Language Understanding
- **RF**: Random Forest
- **RNN**: Recurrent Neural Network
- **SBWC**:Spanish Billion Word Corpus
- **TF-IDF**: Term Frequency-Inverse Document Frequency
- **TF**: Term Frequency
- **UN**: United Nations

5.4 Languages acronyms

- **ar**: Arabic
- **cs**: Czech
- **de**: German
- **en**: English
- **es**: Spanish
- **fr**: French
- **hi**: Hindi
- **it**: Italian
- **ja**: Japanese
- **nl**: Dutch
- **pl**: Polish
- **pt**: Portuguese
- **ru**: Russian
- **tr**: Turkish
- **zh-CN**: Cantonese
- **zh-TW**: Taiwan Chinese
- **zh**: Chinese

References

[1] Steven Abreu. *Automated Architecture Design for Deep Neural Networks*. 2019. arXiv: 1908.10714 [cs.LG].

[2] Jay Alammar. *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)*. 2018. URL: http://jalammar.github.io/illustrated-bert/.

[3] Jay Alammar. *The illustrated transformer*. 2018. URL: http://jalammar.github.io/illustrated-transformer/.

[4] Hunt Allcott and Matthew Gentzkow. "Social media and fake news in the 2016 election". In: *Journal of economic perspectives* 31.2 (2017), pp. 211–36.

[5] Reuben Binns et al. "'It's Reducing a Human Being to a Percentage': Perceptions of Justice in Algorithmic Decisions". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. CHI '18. Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–14. ISBN: 9781450356206. DOI: 10.1145/3173574.3173951.

[6] Christopher M Bishop. *Pattern recognition and machine learning*. Information science and statistics. Softcover published in 2016. New York, NY: Springer, 2006. URL: https://cds.cern.ch/record/998831.

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. "Latent dirichlet allocation". In: *the Journal of machine Learning research* 3 (2003), pp. 993–1022.

[8] Piotr Bojanowski et al. *Enriching Word Vectors with Subword Information*. 2017. arXiv: 1607.04606 [cs.CL].

[9] Samuel R. Bowman et al. *A large annotated corpus for learning natural language inference*. 2015. arXiv: 1508.05326 [cs.CL].

[10] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. "Model compression". In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06*. Philadelphia, PA, USA: ACM Press, 2006, p. 535. ISBN: 9781595933393. DOI: 10.1145/1150402.1150464.

[11] Cristian Cardellino. *Spanish Billion Words Corpus and Embeddings*. 2019. URL: https://crscardellino.github.io/SBWCE/.

[12] Daniel Cer et al. "SemEval-2017 Task 1: Semantic Textual Similarity Multilingual and Crosslingual Focused Evaluation". In: *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (2017). DOI: 10.18653/v1/s17-2001.

[13] CVC Centro Virtual Cervantes. *CVC. Anuario 2020. Informe 2020. El español en cifras*. 2020. URL: https://cvc.cervantes.es/lengua/anuario/anuario_20/informes_ic/p01.htm.

[14] Muthu Chidambaram et al. "Learning Cross-Lingual Sentence Representations via a Multi-task Dual-Encoder Model". In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 250–259. DOI: 10.18653/v1/W19-4330.

[15] Gobinda G. Chowdhury. "Natural language processing". In: *Annual Review of Information Science and Technology* 37.1 (2005), pp. 51–89. ISSN: 00664200. DOI: 10.1002/aris.1440370103.

[16] Christopher Cieri et al. "Selection criteria for low resource language programs". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. 2016, pp. 4543–4549.

[17] Alexis Conneau and Douwe Kiela. *SentEval: An Evaluation Toolkit for Universal Sentence Representations*. 2018. arXiv: 1803.05449 [cs.CL].

[18] Alexis Conneau et al. *Supervised Learning of Universal Sentence Representations from Natural Language Inference Data*. 2018. arXiv: 1705.02364 [cs.CL].

[19] Alexis Conneau et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. arXiv: 1911.02116 [cs.CL].

[20] Peter Dalgaard. *Introductory statistics with r*. Statistics and Computing. New York, NY: Springer New York, 2008. ISBN: 978-0-387-79053-4. DOI: 10.1007/978-0-387-79054-1.

[21] Jesse Davis and Mark Goadrich. "The Relationship between Precision-Recall and ROC Curves". In: ICML '06. Pittsburgh, Pennsylvania, USA: Association for Computing Machinery, 2006, pp. 233–240. ISBN: 1595933832. DOI: 10.1145/1143844.1143874.

[22] Rosario Delgado and Xavier-Andoni Tibau. "Why Cohen's Kappa should be avoided as performance measure in classification". In: *PLOS ONE* 14.9 (2019). Ed. by Quanquan Gu, e0222916. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0222916.

[23] Jacob Devlin and Ming-Wei Chang. *Open sourcing bert: state-of-the-art pre-training for natural language processing*. 2018. URL: http://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html.

[24] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[25] Paul DiMaggio, Manish Nag, and David Blei. "Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of U.S. government arts funding". In: *Poetics* 41.6 (2013), pp. 570–606. ISSN: 0304-422X. DOI: https://doi.org/10.1016/j.poetic.2013.08.004.

[26] Caitlin N. Dreisbach et al. "A systematic review of natural language processing and text mining of symptoms from electronic patient-authored text data". In: *International journal of medical informatics* 125 (2019), pp. 37–46. DOI: 10.1016/j.ijmedinf.2019.02.008.

[27] Fangxiaoyu Feng et al. *Language-agnostic BERT Sentence Embedding*. 2020. arXiv: 2007.01852 [cs.CL].

[28] Nathan Geffen. "Justice after AIDS denialism: should there be prosecutions and compensation?" In: *Journal of acquired immune deficiency syndromes (1999)* 51.4 (2009), pp. 454–455. ISSN: 1525-4135. DOI: 10.1097/qai.0b013e3181ab6da2.

[29] Margherita Grandini, Enrico Bagli, and Giorgio Visani. *Metrics for Multi-Class Classification: an Overview*. 2020. arXiv: 2008.05756 [stat.ML].

[30] Gringer. *Phrase structure rules*. File: Cgisf-tgg.svg. Dec. 2008. URL: https://en.wikipedia.org/wiki/Phrase_structure_rules#/media/File:Cgisf-tgg.svg.

[31] Maarten Grootendorst. *BERTopic: Leveraging BERT and c-TF-IDF to create easily interpretable topics*. Version v0.4.2. 2020. DOI: 10.5281/zenodo.4430182.

[32] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: 1503.02531 [stat.ML].

[33] Shammamah Hossain. "Visualization of Bioinformatics Data with Dash Bio". In: *Proceedings of the 18th Python in Science Conference*. Ed. by Chris Calloway et al. 2019, pp. 126–133. DOI: 10.25080/Majora-7ddc1dd1-012.

[34] Samuel Humeau et al. *Poly-encoders: Transformer Architectures and Pre-training Strategies for Fast and Accurate Multi-sentence Scoring*. 2020. arXiv: 1905.01969 [cs.CL].

[35] Md Saiful Islam et al. "COVID-19–Related Infodemic and Its Impact on Public Health: A Global Social Media Analysis". In: *The American Journal of Tropical Medicine and Hygiene* 103.4 (2020), pp. 1621–1629. DOI: 10.4269/ajtmh.20-0812.

[36] Heejung Jwa et al. "exBAKE: Automatic Fake News Detection Model Based on Bidirectional Encoder Representations from Transformers (BERT)". In: *Applied Sciences* 9.19 (2019). ISSN: 2076-3417. DOI: 10.3390/app9194062.

[37] Andrej Karpathy. *The Unreasonable Effectiveness of Recurrent Neural Networks*. London, 2015. URL: http://karpathy.github.io/2015/05/21/rnn-effectiveness/.

[38] Aman Kedia and Mayank Rasu. *Hands-on python natural language processing*. S.l.: Packt Publishing, 2020. ISBN: 9781838989590.

[39] Simon Kemp. *Digital 2020: october global statshot*. 2020. URL: https://datareportal.com/reports/digital-2020-october-global-statshot.

[40] Kowsari et al. "Text classification algorithms: a survey". In: *Information* 10.4 (2019), p. 150. ISSN: 2078-2489. DOI: 10.3390/info10040150.

[41] Guillaume Lample and Alexis Conneau. *Cross-lingual Language Model Pretraining*. 2019. arXiv: 1901.07291 [cs.CL].

[42] John D. Lee and Katrina A. See. "Trust in Automation: Designing for Appropriate Reliance". In: *Human Factors* 46.1 (2004), pp. 50–80. DOI: 10.1518/hfes.46.1.50_30392.

[43] Robert B. Lees and Noam Chomsky. "Syntactic structures". In: *Language* 33.3 (1957), p. 375. ISSN: 00978507. DOI: 10.2307/411160.

[44] M. Paul Lewis et al. *Ethnologue®: languages of the americas and the pacific*. 2016. ISBN: 9781556714030.

[45] Elizabeth D Liddy. "Natural language processing". In: *Encyclopedia of Library and Information Science*. Vol. 3. New York, NY, USA: Taylor & Francis, 2001. ISBN: 9780824720797.

[46] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[47] Edward Ma. *A robustly optimized bert pretraining approach*. 2019.

[48] Bill MacCartney. "NATURAL LANGUAGE INFERENCE". PhD thesis. Standford University, 2009. URL: https://nlp.stanford.edu/~wcmac/.

[49] Billl MacCartney. "Understanding Natural Language Understanding". In: 2014. URL: https://nlp.stanford.edu/~wcmac/.

[50] Chris McCormick and Nick Ryan. *BERT Word Embeddings Tutorial*. 2019. URL: https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/#31-running-bert-on-our-text.

[51] Marion Meyers, Gerhard Weiss, and Gerasimos Spanakis. "Fake news detection on twitter using propagation structures". In: *Disinformation in Open Online Media*. Ed. by Max van Duijn et al. Vol. 12259. Cham: Springer International Publishing, 2020, pp. 138–158. ISBN: 9783030618407. DOI: 10.1007/978-3-030-61841-4_10.

[52] Tomas Mikolov et al. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].

[53] Geethu Mohan and M. Monica Subashini. "Chapter 4 - Medical Imaging With Intelligent Systems: A Review". In: *Deep Learning and Parallel Computing Environment for Bioengineering Systems*. Ed. by Arun Kumar Sangaiah. Academic Press, 2019, pp. 53–73. ISBN: 978-0-12-816718-2. DOI: https://doi.org/10.1016/B978-0-12-816718-2.00011-7.

[54] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. "Natural language processing: an introduction". In: *Journal of the American Medical Informatics Association* 18.5 (2011), pp. 544–551. DOI: 10.1136/amiajnl-2011-000464.

[55] Salman Bin Naeem and Rubina Bhatti. "The Covid-19 'infodemic': a new front for information professionals". In: *Health Information & Libraries Journal* 37.3 (2020), pp. 233–239. ISSN: 1471-1834, 1471-1842. DOI: 10.1111/hir.12311.

[56] Vincent Nguyen et al. *Searching Scientific Literature for Answers on COVID-19 Questions*. 2020. arXiv: 2007.02492 [cs.IR].

[57] Manimbulu Nlooto and Panjasaram Naidoo. "Traditional, complementary and alternative medicine use by HIV patients a decade after public sector antiretroviral therapy roll out in South Africa: a cross sectional study". In: *BMC Complementary and Alternative Medicine* 16.1 (2016), p. 128. ISSN: 1472-6882. DOI: 10.1186/s12906-016-1101-5.

[58] Bo Pang and Lillian Lee. "Opinion Mining and Sentiment Analysis". In: *Found. Trends Inf. Retr.* 2.1–2 (2008), pp. 1–135. ISSN: 1554-0669. DOI: 10.1561/1500000011.

[59] Kyubyong Park. *Kyubyong/nlp_tasks*. 2020. URL: https://github.com/Kyubyong/nlp_tasks.

[60] Vikas Raunak, Vivek Gupta, and Florian Metze. "Effective Dimensionality Reduction for Word Embeddings". In: *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. Florence, Italy: Association for Computational Linguistics, 2019, pp. 235–243. DOI: 10.18653/v1/W19-4328.

[61] Radim Řehůřek and Petr Sojka. "Software Framework for Topic Modelling with Large Corpora". English. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010, pp. 45–50.

[62] Nils Reimers, Philip Beyer, and Iryna Gurevych. "Task-Oriented Intrinsic Evaluation of Semantic Textual Similarity". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, 2016, pp. 87–96.

[63] Nils Reimers and Iryna Gurevych. *Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation*. 2020. arXiv: 2004.09813 [cs.CL].

[64] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL].

[65] Benjamin Riedel et al. *A simple but tough-to-beat baseline for the Fake News Challenge stance detection task*. 2018. arXiv: 1707.03264 [cs.CL].

[66] Kirk Roberts et al. "TREC-COVID: rationale and structure of an information retrieval shared task for COVID-19". In: *Journal of the American Medical Informatics Association* 27.9 (2020), pp. 1431–1436. ISSN: 1527-974X. DOI: 10.1093/jamia/ocaa091.

[67] Stephen Robertson et al. "Okapi at TREC-3". In: *Overview of the Third Text REtrieval Conference (TREC-3)*. Gaithersburg, MD: NIST, 1995, pp. 109–126.

[68] Sebastian Ruder, Anders Søgaard, and Ivan Vulić. "Unsupervised Cross-Lingual Representation Learning". In: *Proceedings of ACL 2019, Tutorial Abstracts*. 2019, pp. 31–38.

[69] Julio Villena-Román y Sara Lana-Serrano y Eugenio Martínez-Cámara y José Carlos González-Cristóbal. "TASS - Workshop on Sentiment Analysis at SEPLN". In: *Procesamiento del Lenguaje Natural* 50.0 (2013), pp. 37–44. ISSN: 1989-7553. URL: http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/4657.

[70] Holger Schwenk et al. *WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia*. 2019. arXiv: 1907.05791 [cs.CL].

[71] Grigori Sidorov et al. "Soft similarity and soft cosine measure: similarity of features in vector space model". In: *Computación y Sistemas* 18.3 (2014). ISSN: 1405-5546. DOI: 10.13053/cys-18-3-2043.

[72] Richard Socher et al. "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, 2013, pp. 1631–1642. URL: https://www.aclweb.org/anthology/D13-1170.

[73] Anders Søgaard et al. "Cross-lingual word embeddings". In: *Synthesis Lectures on Human Language Technologies* 12.2 (2019), pp. 1–132. ISSN: 1947-4040, 1947-4059. DOI: 10.2200/S00920ED2V01Y201904HLT042.

[74] Robyn Speer and Joshua Chin. *An Ensemble Method to Produce High-Quality Word Embeddings (2016)*. 2019. arXiv: 1604.01692 [cs.CL].

[75] Jalaj Thanaki. *Python Natural Language Processing: explore NLP with machine learning and deep learning techniques*. Birmingham Mumbai: Packt, 2017. ISBN: 9781787121423.

[76] Jörg Tiedemann. "Parallel Data, Tools and Interfaces in OPUS". In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA), 2012, pp. 2214–2218. URL: http://www.lrec-conf.org/proceedings/lrec2012/pdf/463_Paper.pdf.

[77] Alexandre Trilla. "Natural Language Processing techniques in Text-To-Speech synthesis and Automatic Speech Recognition". In: *Departament de Tecnologies Media* (2009), pp. 1–5.

[78] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[79] Rutvik Vijjali et al. *Two Stage Transformer Model for COVID-19 Fake News Detection and Fact Checking*. 2020. arXiv: 2011.13253 [cs.CL].

[80] Ellen M. Voorhees and Dawn M. Tice. "Building a question answering test collection". In: *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '00*. Athens, Greece: ACM Press, 2000, pp. 200–207. ISBN: 9781581132267. DOI: 10.1145/345508.345577.

[81] Lucy Lu Wang et al. *CORD-19: The COVID-19 Open Research Dataset*. 2020. arXiv: 2004.10706 [cs.DL].

[82] Joseph Weizenbaum. "ELIZA—a computer program for the study of natural language communication between man and machine". In: *Communications of the ACM* 9.1 (1966), pp. 36–45. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/365153.365168.

[83] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, 2020, pp. 38–45. DOI: 10.18653/v1/2020.emnlp-demos.6.

[84] Wenpeng Yin and Hinrich Schütze. *Learning Meta-Embeddings by Using Ensembles of Embedding Sets*. 2015. arXiv: 1508.04257 [cs.CL].

[85] Tom Young et al. *Recent Trends in Deep Learning Based Natural Language Processing*. 2018. arXiv: 1708.02709 [cs.CL].

[86] Jezia Zakraoui, Moutaz Saleh, and Jihad Al Ja'am. "Text-to-picture tools, systems, and approaches: a survey". In: *Multimedia Tools and Applications* 78.16 (2019), pp. 22833–22859. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-019-7541-4.

[87] Vitalii Zhelezniak et al. "Correlation coefficients and semantic textual similarity". In: *Proceedings of the 2019 Conference of the North*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 951–962. DOI: 10.18653/v1/N19-1100.

[88] Ming Zhou et al. "Progress in Neural NLP: Modeling, Learning, and Reasoning". In: *Engineering* 6.3 (2020), pp. 275–290. ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2019.12.014.

A

Semantic Search Appendix

PCA cumulative explained variance percentage as a function of the number of components for the multilingual Sentence-Transformers models used for the ensemble in parallel multilingual train data from OPUS-NewsCommentary, TED2020 and Wikimatrix. Looking at this plot we can see how the first 200 components contain approximately 90% of the variance of the embeddings for each model.

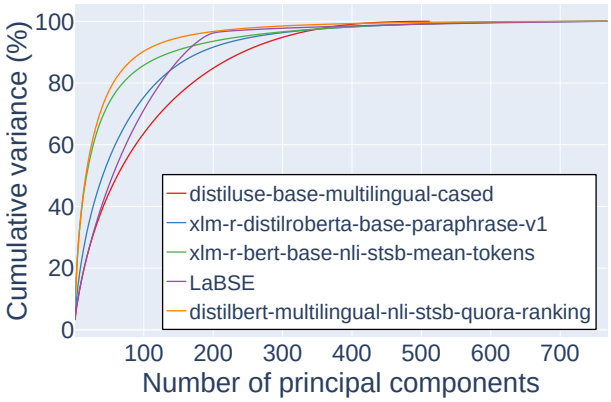


Figure A.1: PCA cumulative explained variance percentage as a function of the number of components for the multilingual Sentence-Transformers models used. Link to interact with the data.

Twitter accounts, nationalities and languages spoken by the Fact-Checker used for the semantic search dashboard:

Fact-Checker	**Location**	**Language**
Fatabyyano_com	Jordan	ar
MaharatNews	Lebanon	ar

Fact-Checker	**Location**	**Language**
correctiv_org	Germany	de
dpa	Germany	de

Fact-Checker	**Location**	**Language**
Chequeado	Argentina	es
cotejoinfo	Venezuela	es
ECUADORCHEQUEA	Ecuador	es
EFEVerifica	Spain	es
ElSabuesoAP	Mexico	es
lasillavacia	Colombia	es
maldita_ciencia	Spain	es
malditobulo	Spain	es
Newtral	Spain	es

Fact-Checker	**Location**	**Language**
CheckCongo	Congo	fr
CheckNewsfr	France	fr
franceinfo	France	fr
lemondefr	France	fr
Observateurs	France	fr

Fact-Checker	**Location**	**Language**
PagellaPolitica	Italy	it

Fact-Checker	**Location**	**Language**
Nunl	Netherlands	nl

Fact-Checker	**Location**	**Language**
DemagogPL	Poland	pl

Fact-Checker	**Location**	**Language**
aosfatos	Brasil	pt
estadaoverifica	Brasil	pt
JornalPoligrafo	Protugal	pt
observadorpt	Portugal	pt

Fact-Checker	**Location**	**Language**
StopFakingNews	Ucrany	ru

Fact-Checker	**Location**	**Language**
dogrulukpayicom	Turkey	tr
teyitorg	Turkey	tr

Fact-Checker	**Location**	**Language**
AAPNewswire	Australia	en
ABCFactCheck	Autralia	en
AfricaCheck	Africa	en
AP	United States of America	en
boomlive_in	India	en
Check_Your_Fact	United States of America	en
ClimateFdbk	International	en
dubawaNG	Nigeria	en
eye_digit	India	en
factchecknet	International	en
FactCheckNI	United Kingdom	en
FactCrescendo	India	en
FactlyIndia	India	en
FerretScot	United Kingdom	en
FullFact	United Kingdom	en
ghana_fact	Ghana	en
GlennKesslerWP	United States of America	en
Indiatoday	India	en
LogicallyAI	United Kingdom	en
mediawise	United States of America	en
NewsMeter_In	India	en
NewsMobileIndia	India	en
newsvishvas	India	en
PesaCheck	Kenya	en
Poynter	International	en
rapplerdotcom	Philippines	en
ReutersAgency	United States of America	en
snopes	United States of America	en
SouthAsiaCheck	Nepal	en
thedispatch	United States of America	en
thejournal_ie	United Kingdom	en
TheQuint	India	en
ThipMedia	India	en
USATODAY	United States of America	en
verafiles	Philippines	en

B

Sentiment Analysis Appendix

B.1 Sentiment Analysis

In this Appendix we display the parameters used for the different vectorization and classification models applied in Sentiment Analysis. All other parameters not shown are set as default. The code is available at this Github repository

Parameters used for Word2Vec + TF-IDF trained in SST2

- Word2Vec
  - skip-gram architecture
  - context (window) size = 10
  - minCount = 10
  - epochs = 5
  - dim = 300
- TF-IDF
  - max features = 5000
  - min df = 0
  - max df = 0.8
  - ngram range = 1, 3
  - strip accents = unicode
- KNN
  - n neighbors = 1
- LR
  - regularization = L1 (Lasso)

  - Inverse of regularization strength (C) = 1.585
  - solver = liblinear
  - max iter = 1000
- RF
  - max depth = 10
  - n estimators = 30

Parameters used for Word2Vec from Gensim + TF-IDF in SST2

- Word2Vec
  - training data: Google News corpus
  - skip-gram architecture
  - context (window) size = 10
  - dim = 300
- TF-IDF
  - max features = 5000
  - min df = 0
  - max df = 0.8
  - ngram range = 1, 3
  - strip accents = unicode
- KNN
  - n neighbors = 1
- LR
  - regularization = L1 (Lasso)
  - Inverse of regularization strength (C) = 1.585
  - solver = liblinear
  - max iter = 1000
- RF
  - max depth = 30
  - n estimators = 400

Parameters used for Transformer-based models in SST2

- xlm-roberta-base
  - lr = 0.00005
  - epochs = 2
  - weight decay = 0.005

  - scheduler = linear schedule with warmup
  - batch size = 16
  - gradient accumulation steps = 2

- distilbert-base-multilingual-cased
  - lr = 0.00001
  - epochs = 2
  - weight decay = 0.005
  - scheduler = linear schedule with warmup
  - batch size = 16
  - gradient accumulation steps = 2

- distilroberta-base
  - lr = 0.000057
  - epochs = 2
  - weight decay = 0.005
  - scheduler = linear schedule with warmup
  - batch size = 16
  - gradient accumulation steps = 2

- distilbert-base-nli-stsb-mean-tokens
  - lr = 0.000009
  - epochs = 2
  - weight decay = 0.005
  - scheduler = linear schedule with warmup
  - batch size = 16
  - gradient accumulation steps = 2

Parameters used for FastText pre-trained on SBWC in TASS

- FastText
  - skip-gram architecture
  - min subword-ngram = 3
  - max subword-ngram = 6
  - minCount = 5
  - epochs = 20
  - dim = 300

- KNN
  - n neighbors = 1

- LR

  - regularization = L1 (Lasso)
  - Inverse of regularization strength (C) = 10
  - solver = liblinear
  - max iter = 1000
- RF
  - max depth = 40
  - n estimators = 500

Parameters used for Transformer-based models in TASS

- xlm-roberta-base
  - lr = 0.00005
  - epochs = 2
  - weight decay = 0.005
  - scheduler = linear schedule with warmup
  - batch size = 16
  - gradient accumulation steps = 2
- distilbert-base-multilingual-cased
  - lr = 0.0000256
  - epochs = 2
  - weight decay = 0.005492
  - scheduler = linear schedule with warmup
  - batch size = 16
  - gradient accumulation steps = 2
- distilbert-multilingual-nli-stsb-quora-ranking
  - lr =0.0000318
  - epochs = 2
  - weight decay = 0.005432
  - scheduler = linear schedule with warmup
  - batch size = 32
  - gradient accumulation steps = 3
- BETO-uncased
  - lr = 0.00005
  - epochs = 2
  - weight decay = 0.0005
  - scheduler = linear schedule with warmup
  - batch size = 16
  - gradient accumulation steps = 2

C

Topic Analysis Appendix

C.1 Parameters used for Topic modeling based on BERTopic

- UMAP
  - number of components = 5
  - size of local neighborhood = 12
  - metric = cosine
  - low memory = True
  - random state = 42
  - remaining parameters are set as default.
- HDBSCAN
  - min topic size = 200
  - min samples size = 70
  - metric = euclidean
  - remaining parameters are set as default.
- c-TF-IDF
  - n gram range = 1, 2
  - min df = 0.005
  - max df = 0.98
  - remaining parameters are set as default.

C.2 Topics and most representative words

Topic	Top 20 representative words
0	'farmer', 'farmersprotest', 'farmlaws', 'talk', 'protest', 'farm', 'round talk', 'farmer protest', 'law', 'farm law', 'minister', 'govt', 'farmersprotest farmlaws', 'boomfactcheck', 'government', 'protesting', 'agriculture', 'talk farmer', 'farmer leader', 'farmersprotests'
1	'germany', 'german', 'corona', 'good morning', 'new', 'infection', 'morning', 'important', 'today', 'news blog', 'blog', 'covid germany', 'important today', 'corona infection', 'blog via', 'new corona', 'lockdown', 'infection germany', 'death', 'morning important'
2	'top news', 'news', 'watch top', 'itlivestream watch', 'newsmobile', 'headline', 'top headline', 'news story', 'story', 'watch newsmobile', 'prime time', 'newsmobile prime', 'news day', 'time bulletin', 'primetime','bulletin top', 'headline hour', 'news primetime', 'story itlivestream', 'day news'
3	'china', 'chinese', 'laboratory', 'coronavirus', 'scientific', 'vaccine', 'coronavirus created', 'created laboratory', 'published scientific', 'scientific journal','evidence', 'chinese virologist', 'covid china', 'virus', 'published', 'virologist', 'suggests coronavirus', 'evidence published', 'laboratory scientific', 'scientific evidence'
4	'spain', 'madrid', 'map', 'covid spain', 'community madrid', 'coronavirus', 'vaccination', 'spain world', 'coronavirus spain', 'vaccination covid', 'data', 'infection', 'basic health', 'question answer', 'second wave', 'infection spain', 'spanish', 'health', 'wave', 'campaign coronavirus'
5	'typhoon', 'rain', 'ulyssesph', 'photo', 'city', 'rollyph', 'tropical', 'storm', 'november num', 'november', 'pepitoph', 'area', 'heavy', 'moon', 'province', 'relief', 'weatheralert', 'october', 'water', 'luzon'
6	'brazil', 'brazilian', 'bolsonaro', 'vaccine', 'covid brazil', 'rio', 'coronavac', 'trial', 'jair bolsonaro', 'chinese', 'volunteer', 'clinical', 'president jair', 'chinese vaccine', 'coronavirus', 'china', 'de janeiro', 'janeiro', 'rio de', 'pandemic'
7	'mask', 'health', 'face', 'face mask', 'coronavirus', 'wearing mask', 'prevent', 'cure', 'water', 'healthy', 'wear mask', 'use', 'hand', 'vitamin', 'help', 'diet', 'virus', 'wash', 'washing', 'yoga'
8	'nigeria', 'dubawachecks', 'nigerian', 'fact', 'read', 'read dubawachecks', 'protest', 'endsars protest', 'ghana', 'read fact', 'fact check', 'check', 'fact checking', 'checking', 'fact dubawachecks', 'ghanaelections', 'protest nigeria', 'nigeria read', 'fake', 'ghanaelections num'
9	'christmas', 'thanksgiving', 'holiday', 'family', 'coronavirus', 'santa', 'season', 'celebrate', 'celebration', 'ski', 'new year', 'winter', 'new', 'restriction', 'celebrate christmas', 'resort', 'santa claus', 'risk', 'holiday season', 'christmas new'
10	'cricket', 'australia', 'smith', 'num num', 'ipl num', 'steve smith', 'test', 'india', 'stevesmith', 'num run', 'num over', 'run', 'team', 'match', 'batsman', 'sydney', 'day num', 'captain', 'ball', 'num wicket'
11	'gt', 'gt gt', 'poll', 'top', 'ap top', 'top num', 'full poll', 'pick', 'poll gt', 'football', 'alabama', 'coverage gt', 'game', 'college football', 'gt coverage', 'football writer', 'season', 'nfl', 'maverick', 'pick gt'
12	'pba', 'pba num', 'pbasemis', 'pbasemis pba', 'pbafinals', 'park', 'pbafinals pba', 'barangay ginebra', 'barangay', 'philippine', 'philippine cup', 'filipino', '4q', 'tnt pbafinals', 'num 3q', 'tnt pbasemis', 'game', '3q', 'game num', 'num 4q'
13	'effective', 'num effective', 'vaccine', 'vaccine num', 'moderna', 'efficacy', 'covid vaccine', 'vaccine candidate', 'oxford', 'result', 'candidate', 'num efficacy', 'effective preventing', 'per cent', 'num percent', 'percent', 'coronavirus', 'covidvaccine', 'effectiveness', 'candidate num'
14	'speaker', 'house', 'lord allan', 'lord', 'committee', 'senate', 'budget', 'representative', 'alan peter', 'speaker lord', 'peter', 'president', 'house speaker', 'chair', 'senator', 'bill', 'house representative', 'deputy', 'deputy speaker', 'num budget'
15	'president', 'election', 'donald', 'donald trump', 'joe biden', 'vote', 'president trump', 'house', 'republican', 'president donald', 'election num', 'voter', 'white house', 'white', 'senate', 'presidential', 'ballot', 'fraud', 'democrat', 'elect'
16	'oneplus', 'laptop', 'top', 'oneplus num', 'iphone', 'phone', 'apple', 'launch', 'iphone num', 'amazon', 'oneplus 8t', 'nord', 'smart', 'oneplus nord', 'best', 'num pro', 'new', 'smartphone', 'num oneplus', 'top num'
17	'india', 'video', 'fake', 'fakenews', 'today', 'minister', 'bihar', 'india today', 'itvideo', 'fact', 'bengal', 'social', 'boomfactcheck', 'false', 'news', 'boomfactcheck fakenews', 'police', 'viral', 'factcheck', 'home minister'
18	'school', 'student', 'education', 'teacher', 'class', 'learning', 'pandemic', 'class num', 'reopen', 'exam', 'month', 'parent', 'college', 'coronavirus', 'government', 'distance', 'university', 'high school', 'new', 'distance learning'
19	'died', 'passed away', 'age', 'age num', 'complication', 'died covid', 'num year', 'year old', 'old', 'died age', 'demise', 'death', 'dy covid', 'family', 'dead', 'covid complication', 'veteran', 'dy num', 'related', 'due'
20	'positive', 'tested positive', 'tested', 'positive covid', 'test', 'hospital', 'test positive', 'admitted', 'positive coronavirus', 'coronavirus', 'minister', 'admitted hospital', 'positive new', 'quarantine', 'stable', 'covid positive', 'negative', 'testing', 'contact', 'coronavirus positive'
21	'curfew', 'december', 'government', 'macron', 'emmanuel macron', 'closed', 'school', 'night', 'december num', 'night curfew', 'curfew num', 'january', 'confinement', 'jean castex', 'restriction', 'close', 'lockdown', 'num pm', 'closure', 'ban'
22	'vaccine', 'covid vaccine', 'vaccination', 'india', 'coronavirus', 'covidvaccine', 'coronavaccine', 'covaxin', 'covishield', 'coronavirusvaccine', 'vaccination drive', 'use', 'say', 'coronavirus vaccine', 'minister', 'covid vaccination', 'country', 'said', 'vaccinated', 'government'
23	'france', 'french', 'epidemic', 'covid france', 'hospital', 'patient', 'france covid', 'vaccination', 'wave', 'nurse', 'french people', 'second', 'crisis', 'second wave', 'covid epidemic', 'covid patient', 'vaccination campaign', 'health crisis', 'european', 'wave covid'
24	'vaccine', 'covid vaccine', 'false', 'coronavirus', 'people', 'fakenews', 'virus', 'vaccination', 'social', 'said', 'vaccinated', 'health', 'facebook', 'vaccine covid', 'medium', 'social medium', 'misinformation', 'doctor', 'disease', 'coronavirus vaccine'
25	'pollution', 'air', 'air pollution', 'climate', 'air quality', 'airpollution', 'quality', 'climate change', 'change', 'category', 'delhi air', 'level', 'pollution control', 'poor category', 'control', 'global', 'quality index', 'delhi pollution', 'greenhouse gas', 'india'
26	'lowest', 'pandemic', 'number', 'num num', 'drop', 'month', 'india', 'unemployment', 'coronavirus', 'job', 'million', 'decline', 'economy', 'week', 'fell', 'new', 'active case', 'num lakh', 'lakh', 'wall street'
27	'death', 'num death', 'num hour', 'death covid', 'num new', 'last num', 'report num', 'infection', 'new', 'new infection', 'new case', 'register num', 'num num', 'infected', 'infection num', 'num infected', 'add num', 'death num', 'num infection', 'rise num'
28	'india', 'tally country', 'reported num', 'coronavirus case', 'country num', 'toll increased', 'num new', 'increased num', 'coronavirus', 'death toll', 'increased', 'new', 'num death', 'death', 'new coronavirus', 'country', 'num fresh', 'fresh', 'november reported', 'num read'
29	'case num', 'num death', 'num num', 'death', 'total', 'recovery', 'active case', 'new covid', 'num recovery', 'num new', 'recovery num', 'new', 'report num', 'num total','discharge num', 'total case', 'including num', 'num discharge', 'num including', 'death toll'