UNIVERSIDAD DE INVESTIGACIÓN DE TECNOLOGÍA EXPERIMENTAL YACHAY

Escuela de Ciencias Matemáticas y Computacionales

TÍTULO: Illicit tweet detection using Transformers

Trabajo de integración curricular presentado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información

Autor:

Román Niemes Stadyn

Tutor:

Ph.D. Erick Cuenca

Urcuquí, noviembre de 2023

Autoría

Yo, **STADYN JOSUÉ ROMÁN NIEMES**, con cédula de identidad 0704523554, declaro que las ideas, juicios, valoraciones, interpretaciones, consultas bibliográficas, definiciones y conceptualizaciones expuestas en el presente trabajo; así cómo, los procedimientos y herramientas utilizadas en la investigación, son de absoluta responsabilidad de el/la autor/a del trabajo de integración curricular. Así mismo, me acojo a los reglamentos internos de la Universidad de Investigación de Tecnología Experimental Yachay.

Urcuquí, noviembre de 2023.

Stadyn Josué Román Niemes

CI: 0704523554

Autorización de publicación

Yo, **STADYN JOSUÉ ROMÁN NIEMES**, con cédula de identidad 0704523554, cedo a la Universidad de Investigación de Tecnología Experimental Yachay, los derechos de publicación de la presente obra, sin que deba haber un reconocimiento económico por este concepto. Declaro además que el texto del presente trabajo de titulación no podrá ser cedido a ninguna empresa editorial para su publicación u otros fines, sin contar previamente con la autorización escrita de la Universidad.

Asimismo, autorizo a la Universidad que realice la digitalización y publicación de este trabajo de integración curricular en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación

Urcuquí, noviembre de 2023.

Firmado electrónicamente por:
STADYN JOSUE ROMAN NIEMES

Stadyn Josué Román Niemes

CI: 0704523554

Dedication

I dedicate this work to all the people who have accompanied me in my journey through my career. First, of course, my parents, who have supported me in every personal and professional decision I have made, from studying at Yachay Tech to paying for my passport in case I get an international internship. Next, my girlfriend, Dianita, who has been my main moral support in the year it took me to write my thesis. Without her help, attention, and care, this work would not have been possible. In third place, all my friends, especially my two best friends, Nardy and Sary. Last but not least, my professors, who have guided me through all my career and helped me shape into the professional I am now. Professor Israel Pineda, the first person who inspired me to follow my current path and supported me in my personal projects. Professor Fredy Cuenca, who trusted me to be his teaching assistant, and thanks to whom I am a good C programmer, a skill that has helped me in formal work. Professor Oscar Chang, who had faith in my research abilities, which was rewarded with my first published research paper. Professor Manuel Morocho, who guided me in the two courses I had with him and made me learn about development and AI concepts, skills that are useful for my current job. And, of course, Professor Erick Cuenca, my thesis advisor, who provided all I needed to develop my work and helped me to extend it in the form of research papers.

Stadyn Román

Acknowledgment

Special thanks to my roommate Saúl Figueroa, whom with I had a lot of conversations about my work and ideas, and his feedback was instrumental for the structure and final version of my thesis. Thanks also to Mike Bermeo, another thesis student like me, with whom I compiled and refined the dataset used for this work. Without his help, the results of my thesis would not have been as good as they are. Finally, thanks to my advisor, Erick Cuenca, who provided access to the Twitter API that was used to retrieve the data used in the work.

Stadyn Román

Resumen

Twitter es una red social muy amplia que permite a las personas comunicarse entre sí y expresar sus ideas gracias a su enfoque corto y rápido en las publicaciones. Desafortunadamente, no está exenta de asuntos ilícitos que ocurren en la plataforma. Un problema que surge en las redes sociales en general es cómo se utilizan para promover y difundir servicios ilegales, como la trata de personas, la prostitución, las drogas ilegales, entre otros, gracias al alcance de esas plataformas. Por lo tanto, es importante identificar esta clase de mensajes para detectar actividades ilegales y actuar al respecto. En este trabajo, se presenta y desarrolla un marco para dicha detección utilizando 4 modelos basados en Transformers, la arquitectura más potente actualmente para trabajar en procesamiento del lenguaje natural. Para alimentar y entrenar a los modelos, se seleccionó y etiquetó un dataset de tweets para identificar cuáles contienen ofertas o contenido ilícito en su texto. Dos modelos no basados en Transformers también fueron usados para propósitos de comparación. Los experimentos mostraron que los modelos basados en Transformers son bastante buenos para adaptarse a las particularidades del idioma español y a la estructura que suelen tener los tweets, siendo los modelos BERTweet y DistilBERT los mejores. Además, se observó que los modelos basados en Transformers se pueden adaptar a datasets que no tengan un desbalance fuerte (para este trabajo, una proporción de casi 2:1) y no son afectados cuando se usan datos sintéticos.

Palabras Clave:

Transformer, redes neuronales, tweets ilícitos, procesamiento del lenguaje natural, Twitter

Abstract

Twitter is a very broad social network, allowing people to communicate with each other and express their ideas, thanks to its short and quick approach to posting. Unfortunately, it is not exempt from illicit affairs occurring on the platform. One arising problem in social networks, in general, is how they are used to promote and spread illegal services, such as human trafficking, prostitution, illegal drugs, etc., thanks to those platforms' reach. Thus, it is important to identify those kinds of messages in order to detect illegal activities and act upon them. In this work, a framework for such detection is presented and developed using four Transformer models, the currently most powerful architecture to work in natural language processing. To feed and train the models, a dataset of Spanish tweets was curated and labeled to identify which tweets contained illicit offerings or content in their text. Two non-Transformer models were also used for comparison. The experiments showed that Transformer models are very good at adapting to the particularities of the Spanish language and the structure of tweets, with BERTweet and DistilBERT obtaining the highest results. Also, the Transformer models can adapt to not heavily imbalanced datasets (in this work, a proportion of near 2:1) and are not affected by the use of data augmentation.

Keywords:

Transformer, neural networks, illicit tweets, natural language processing, Twitter

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The boom that social networks experimented with during the last decade has brought them into the mainstream and opened a lot of possibilities for communication and information sharing. But, as much as social networks help to spread viral, useful, and harmless information, they can also be used for malicious purposes by people who promote or share illegal activities and/or content. One of the social networks in which this problem is very relevant is Twitter. This social network contains accounts and tweets promoting human trafficking, prostitution, illegal drugs, and child pornography, among others.

Fortunately, a social network is a giant dataset for natural language processing, which means that artificial intelligence solutions can be developed to detect and report illicit activity and content. Natural language processing allows computers to understand and generate human language, and as such, it can be used to observe illicit messages and identify their defining components. In this work, the transformer architecture will be used to analyze and identify the tweets as illicit or non-illicit. Transformers were introduced in 2017, and since then, they have protagonized major breakthroughs in natural language processing thanks to their attention mechanism, which allows them to retain and understand the particularities of language and its structure. By using transformers, improvements can be achieved compared to previous works, and better detectors can be programmed.

1.2 Problem statement

Detection of illicit activities and content in social networks is a problem that needs to be addressed, as these networks facilitate the spreading of information in general, including illicit messages. While social networks have content filters and enforced guidelines, they can be easily circumvented using acronyms, keywords, or changing characters [5]. This means that criminals have easy-to-access platforms to promote their services and activities, which can reach thousands, if not millions, that provide them with anonymity and keep them hidden from the public eye and law enforcement. As mentioned before, the use of natural language processing allows us to implement better filters that can detect that kind of message and thus help with stopping illegal activities.

There is a wide variety of illicit activities carried out on Twitter, so for this work, the focus will be put on human trafficking, concretely, child prostitution services that are typically related to this issue. The intention is to narrow the research scope of this work and test the performance of different models in this new scope.

Currently, there are websites that offer sexual services from young girls as a sort of "marketplace". These girls generally come from human trafficking, where they are abused physically, psychologically, and sexually [6], and the people who control them advertise their services using social networks, such as Twitter, to hide their traces and activities by using innocuous terms and keywords.

1.3 Objectives

The objectives of this work are mainly aimed at testing the message detection frameworks and how the use of transformers improves them, thanks to their attention capabilities.

1.3.1 General Objective

Implement a natural language processing pipeline for illegal message detection using Twitter data that includes a detection step using different AI models in order to evaluate how they perform with Spanish tweets.

1.3.2 Specific Objectives

1. Find a Transformer model that best suits the specifics of the problem, and can obtain good results with Spanish tweets.

2. Compare the performance of Transformers against other common natural language processing algorithms, such as recurrent neural networks, convolutional neural networks, and statistical methods.

3. Build a detector that can account for the context and intention of the tweets.

1.4 Contributions

This work is the first research of this kind, as previous papers on this topic (Spanish illicit tweets) are scarce, and none of them use Transformers. To complement the point, a derivative article from this work was submitted to the IEEE Ecuador Technical Chapters Meeting (ETCM), which was focused on the literature review presented in the State of the Art chapter, and the paper was accepted with minor corrections.

This work can also help the general research landscape of NLP and Transformers in Spanish, an important aspect because most NLP works and frameworks are focused on the English language.

1.5 Document Organization

This thesis is divided into six chapters, with several sections inside them. Firstly, chapter 1 introduces the problem statement and the objective that this work accomplished, along with the main contributions produced. Chapter 2 presents and delves into the concepts and theories necessary to understand the methodology used for the experiments. Chapter 3 corresponds to the description of the state of the art that applies to this work, divided into two sections. Chapter 4 presents the general framework for the work and explains how the Twitter data was obtained and curated. It also introduced the AI models used for the experiments. Chapter 5 shows the results of the experiments on the 6 AI models in the problem of tweet detection and how their metrics compared. Finally, chapter 6 concludes

that Transformer models were the best option for the task considered, and proposes new research ideas, mainly on the dataset used and the adjustment of the AI models.

Chapter 2

Theoretical Framework

This chapter introduces concepts, ideas, and techniques that are used in the work or are relevant for contextualization and history. This includes a progressive introduction and conceptualization into natural language processing and transformers, the two main techniques used.

2.1 Natural Language Processing

Natural language processing (NLP) is a field of artificial intelligence that deals with the interaction between computers and human languages. NLP is built on the foundations of linguistics, computer science, and artificial intelligence and involves the development of algorithms and models that enable computers to understand and generate human language [7]. One of the key concepts in NLP is the representation of language, which refers to the way in which linguistic information is encoded and represented in a form that can be processed by a computer [8]. Another important aspect of NLP is the development of algorithms and models that can analyze and manipulate natural language data, which can be used for tasks such as part-of-speech tagging, syntactic parsing, and semantic analysis [9]. They can also be applied to more advanced tasks such as machine translation, dialogue systems, and text summarizing.

The history of NLP dates back to the 1950s when researchers began exploring ways to teach computers to understand and generate human languages. One of the earliest milestones in NLP was the development of the ELIZA program by Joseph Weizenbaum in

1964 [10]. ELIZA was a natural language processing program that used pattern matching and substitution to simulate conversation with a human. While it was incapable of true understanding, it could engage in simple conversation and even deceive some users into thinking they were interacting with a real person.

Another significant development in NLP was the creation of the first statistical machine translation systems in the 1980s [11]. These systems used statistical models to translate text from one language to another, by being trained on large amounts of parallel text data in order to learn the translation probabilities between languages. The approach was evaluated with various language pairs, including French-English and Spanish-English, and the statistical machine translation systems ended up outperforming previous rule-based systems.

While the previously mentioned works show the inception and first steps of NLP, its recent history has been marked by significant advancements in developing algorithms and models. One of the key developments in this period has been the use of deep learning and neural networks, which have greatly improved the performance of NLP systems [12, 13]. Another important trend in recent years has been the increasing availability of large amounts of data and computing power, enabling the development of more complex and effective NLP models. Also, the widespread use of NLP technology in areas such as social media and search engines has contributed to the growth of the field, as it has generated large amounts of data that can be used to train and improve NLP models. Additionally, the development of open-source libraries and frameworks, such as TensorFlow[1] and PyTorch[2], has made it easier for researchers and developers to create and share NLP algorithms and models [14, 15].

2.2 Artificial Intelligence

Artificial intelligence (AI) is a field of computer science and engineering that focuses on creating machines that can perform tasks that would typically require human intelligence, such as visual perception, speech recognition, decision-making, and language translation

[1]https://www.tensorflow.org
[2]https://pytorch.org

[16]. It is a broad field encompassing many subdisciplines, including machine learning, natural language processing, robotics, and computer vision [17]. The ultimate goal of AI research is to create systems that can perform any intellectual task that a human can and to extend human capabilities through technology.

2.2.1 Artificial Neural Networks

Artificial neural networks (ANNs) are a type of machine learning model that is inspired by the structure and function of the human brain [18, 19]. They consist of layers of interconnected nodes, called artificial neurons, which process and analyze large input datasets. This way, NNs are able to learn from the data and make predictions or decisions [20].

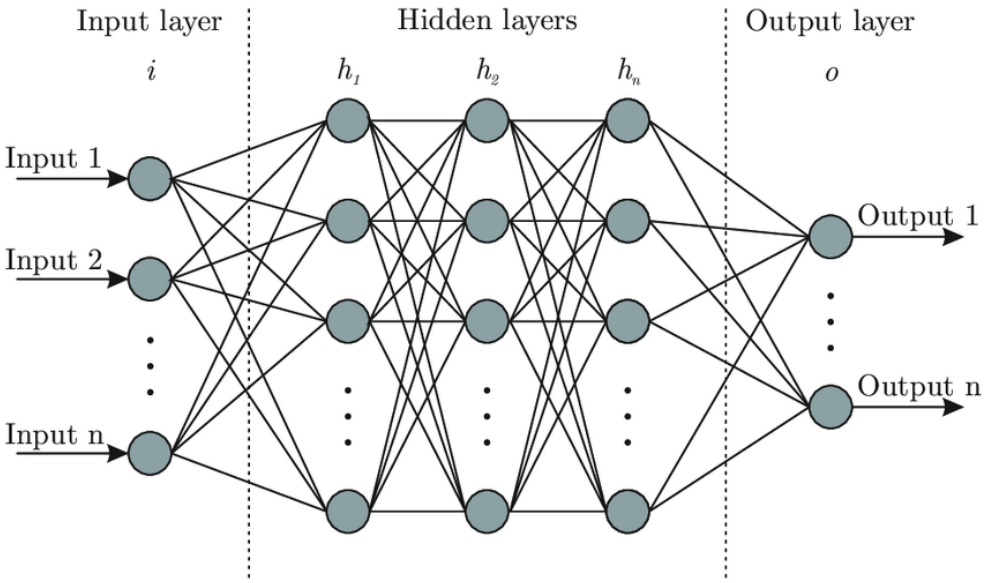


Figure 2.1: Architecture of a typical artificial neural network. Taken from [1]

Figure 2.1 shows the structure of artificial ANNs. In a typical ANN, the input data or input nodes are connected to one or more hidden layers composed of hidden neurons. In these hidden layers, the ANN performs a set of computations using parameters called weights. The final layer produces the final prediction or decision according to the expected outputs. The network learns by adjusting the weights to minimize the error between predicted and actual output. Once trained, the network can make predictions on new input data.

2.2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are neural networks powered by the convolution operation. According to Ian Goodfellow [20], convolution is the integral of the product of the two functions, given that one of them is shifted. In the field of neural networks, convolution is taking a predefined small matrix and "sliding" it through the input (another matrix), multiplying their components and adding them [21]. Said operation allows CNNs to learn spatial properties, features and patterns [20, 22, 23].

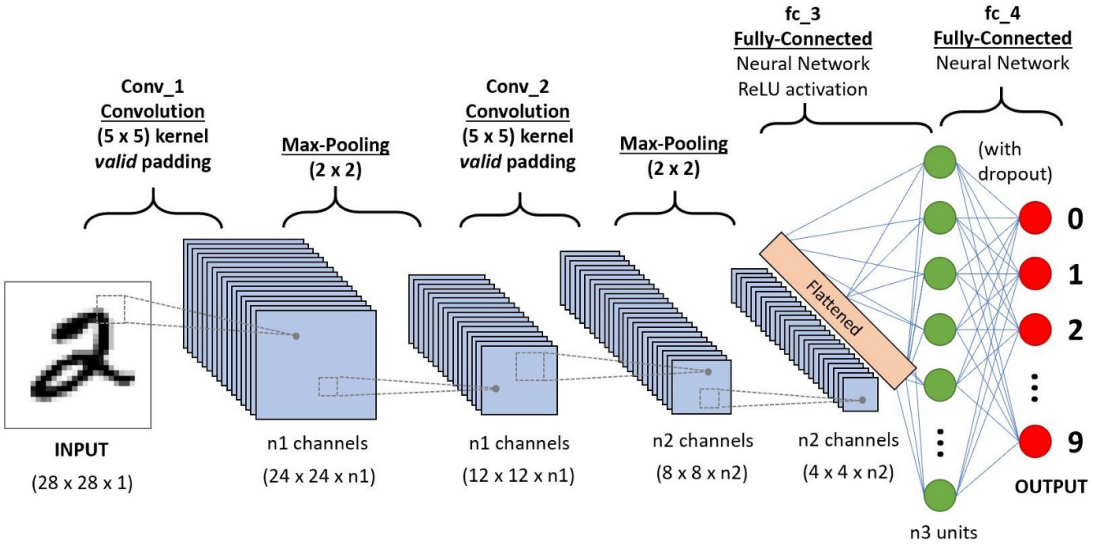


Figure 2.2: Structure of a convolutional neural network. Taken from [2]

Figure 2.2 shows how a convolutional neural network is usually structured. CNNs are composed of three main layers: convolution, pooling, and fully connected layers. In the convolutional layers, the kernel, the matrix that does the convolution, slides through the input, generating scalar values with every slide, repeating the process until no more sliding is possible. The accumulation of the scalar values is called the feature map, which contains information about the patterns of the input image [21, 22, 23].

Next are the pooling layers, which take the feature maps produced by the convolutional layers and shrink them to create new smaller feature maps that contain the most important information overall [21, 22, 23]. This is done by using a kernel again, sliding it through the feature maps, and performing a pooling operation. One of the most used pooling methods is max pooling, which consists of taking the highest value from a group of values in the

feature map [24]. This process is done iteratively until there are no more groups of values. This process improves and complements the feature extraction from convolutional layers and helps improve the performance of the network [23].

Finally, fully convolutional layers have the task of mapping the features obtained with convolutional and pooling layers into the final output. This objective is achieved using an activation function, which decides what neurons contribute to the knowledge of the network [21, 22, 23]. Typically, non-linear activation layers are used because they help the network learn more complicated things [23, 25]. Among the most used activation functions are the sigmoid, rectified linear activation function (ReLU), leaky ReLU and hyperbolic tangent (tanh) [23].

2.2.3 Recurrent Neural Networks

A Recurrent Neural Network (RNN) is a type of NN that is designed to process sequential data, such as time series or natural language [26]. Unlike a traditional neural network, which processes data independently, an RNN processes data in a sequential manner, taking into account the previous input in the sequence when processing the current input [27]. This makes RNNs well-suited for NLP tasks such as language translation and speech recognition, where the context of previous words or sounds is important for understanding the meaning of the current word or sound.

Figure 2.3 shows how RNNs are structures and their recurrent component. RNNs are capable of learning and making predictions based on sequential data. This is enabled by the use of feedback connections within the network, which allow information to flow between time steps and facilitate the learning of temporal dependencies [28], which is in fact the recurrent component of RNNs, and the difference between RNNs and typical NNs In addition to their dynamic and predictive capabilities, RNNs often have many more connections within each layer than other types of neural networks, which can allow them to capture more complex patterns in the data [29].

RNNs have been developed and worked on since the 20th century. One of the first papers on RNNs was published by Jeffrey Elman in 1990, in which he described the use of a simple recurrent neural network for predicting the next word in a sentence based

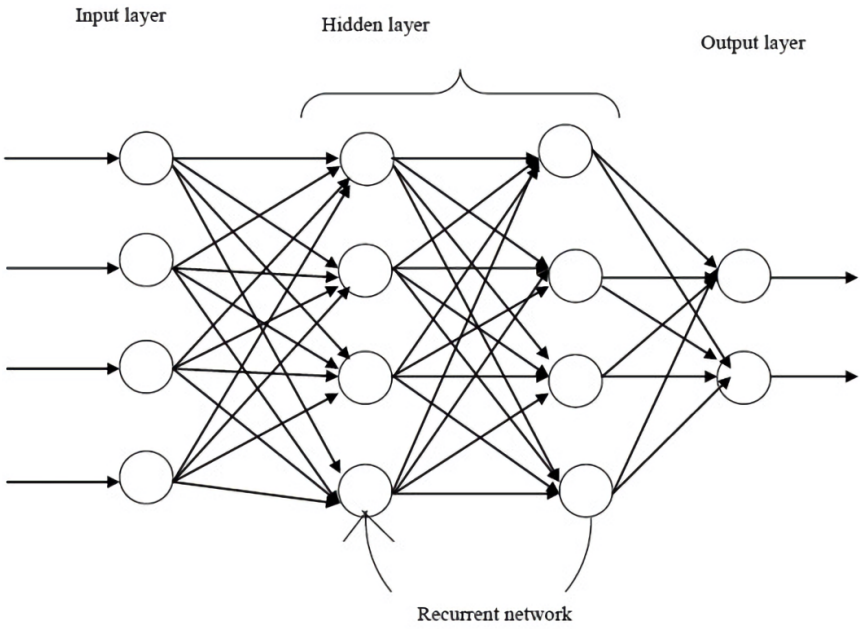


Figure 2.3: Structure of a recurrent neural network. Taken from [3]

on the previous words [30]. This work was followed by several other papers in the early 1990s that further explored the use of RNNs for natural language processing tasks, and the development of new techniques and improvements. One of such works and one of the most influential ones was the development of long short-term memory (LSTM) networks by Hochreiter and Schmidhuber, in order to overcome the vanishing gradient problem [31]. This approach introduced the use of gating mechanisms to capture long-term dependencies in sequential data better. One of the key contributions of the LSTM paper was the introduction of the forget gate, which allows the network to preserve or discard information selectively.

2.2.4 Transformers

In 2017, a major advancement in AI was made with the introduction of transformers. These models were initially developed to improve natural language processing (NLP) solutions. Previously, recurrent neural networks (RNNs) were the primary architecture used for NLP tasks. Although RNNs achieved good results, they were not optimal for dealing with large sentences due to their sequential nature, which resulted in the loss of information. This disadvantage is caused by the fact that RNNs only "remember" the last word processed, which is then used in the encoding process of the next word [27, 32]. This means that each

step only considers the previous step in processing, which can lead to suboptimal results for NLP tasks, as the context and meaning of a word in a sentence depends not only on the previous word but on all words in the sentence. For example, when processing a sentence, each word in the sentence is passed through the network one at a time, and the hidden state of the network is updated with the information from the current word. As the sentence gets longer, the hidden state becomes increasingly complex, as it has to hold information from all the previous words in the sentence. This increases the amount of computation required and can cause the network to become slow and inefficient. Additionally, as the sentence gets longer, the probability of errors or vanishing gradients increases, which can make it difficult for the network to learn the correct representation of the sentence.

To combat this limitation, the attention mechanism was introduced. The attention mechanism functions by extracting information from the entire sentence/sequence by using a weighted sum of all the past states of the encoder, generating a matrix. This means that all words are treated by their real importance, and the overall context is considered, prioritizing words with higher weight and allowing the model to focus on the right element of the input to predict the next element of the output [4, 32].

Figure 2.4 shows how the transformer architecture is composed, including the attention mechanism. The attention mechanism is improved on by using multi-head attention, which applies self-attention to different segments of words, allowing the transformer to have better discrimination capabilities. As each head will produce its own resulting matrix, all matrices are concatenated and multiplied by an additional weight matrix, generating an output matrix that contains information from all the heads [4, 33].

The improvements of transformers regarding NLP tasks were demonstrated in various papers. One of the more influential papers was the work titled as "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" [34]. BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based model that uses a technique called masked language modeling, which involves randomly masking a percentage of the input tokens and then training the model to predict the original values of the masked tokens. This allows the model to learn contextual relationships between words in the input sequence. BERT also makes use of "bidirectional" self-attention, which allows the model to consider the relationships between all input tokens at each step of the processing rather

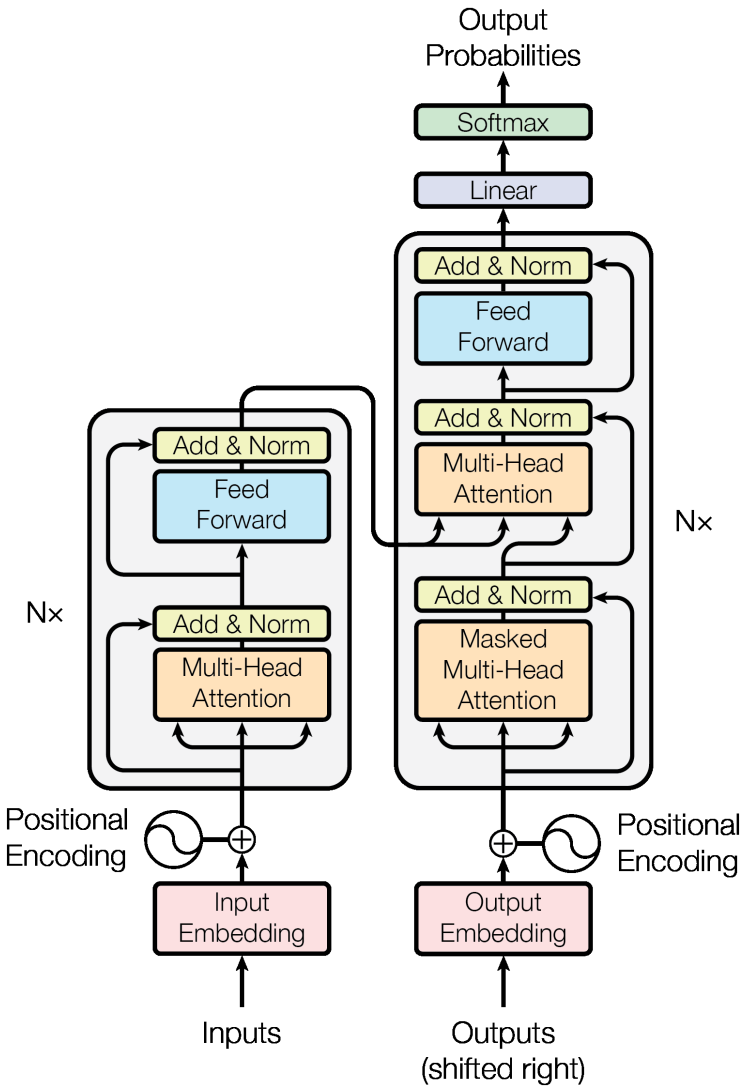


Figure 2.4: Model overview of the Transformer architecture. Taken from [4]

than just the ones preceding a given token, as in traditional language models. Thanks to these specifications, BERT achieved state-of-the-art results on NLP tasks such as question answering and language understanding.

Another important breakthrough using transformers is the paper of Radford and Wu, called "Language Models are Unsupervised Multitask Learners" [35]. This paper introduced GPT-2 (Generative Pre-training 2), a transformer-based language model that was trained on a large dataset of web pages in a similar way to BERT but designed to generate text in a more open-ended manner. This meant GPT-2 was able to perform a wide range of language generation tasks such as translation, summarization, and question answering.

2.3 Illicit Messages Detection

Illicit messages refer to any form of communication that violates the law or goes against ethical norms. This can include activities such as cybercrime, cyberstalking, cyberbullying, and the sharing of illegal materials, like illegal drugs, child pornography, or offers about selling people [36, 37, 38]. Because those kinds of messages can be concealed using a variety of methods, it is important to develop techniques and frameworks to detect them.

One strategy to detect this kind of message is to search for keywords or specific queries in messages and flag texts containing them as illicit [39]. Another way is to monitor network traffic for suspicious activities, such as the transfer of large amounts of data, which can indicate the presence of illegal activities [40]. But overall, these approaches have limitations, mainly that there are indicators of possible illicit conduct, but are not enough to determine anything by themselves. Thus, new and better approaches are necessary to improve detection, and the main field used for this task is AI. This is done by using ML algorithms that analyze patterns in language use, such as word choice and syntax. Mainly, by using NLP techniques such as sentiment analysis and text classification, it is possible to identify content corresponding to negative or illicit activities such as cyberbullying, online harassment, promotion or selling of drugs, and human trafficking [41].

The use of AI for illicit message detection dates back to the early 2000s when machine learning algorithms were first applied to the problem of spam filtering. In 2002, Paul Graham published an online article named "A Plan for Spam", in which he described the use of Bayesian statistics to classify email messages as either spam or non-spam based on a set of numerical features extracted from the text of the message [42]. This approach was the foundation of later improvements and techniques, such as better Bayesian techniques and the use of support vector machines (SVMs).

More recently, artificial intelligence has been used for illicit message detection tasks, including the detection of hate speech, cyberbullying, and other forms of online harassment. For example, in 2020, De Angelis and Pelasso published a paper titled "Cyberbullying Detection Through Machine Learning: Can Technology Help to Prevent Internet Bullying?" [43], in which they reviewed and listed the different techniques and approaches used to detect these harmful texts, including SVMs, naive bayes statistics and CNNs, and their

results. Even when AI can be used for detecting harmful messages and illicit activities, there are concerns about bias on detection and the potential for misuse. For example, in 2019, researchers at the University of Washington published a paper titled "The Risk of Racial Bias in Hate Speech Detection" [44], in which they demonstrated that several commercially available hate speech detection systems were significantly more accurate at identifying hate speech directed at white people than at identifying hate speech directed at other racial groups, by analyzing the propagation of the bias against a dataset for african american english, and how neural networks that used the annotations of that dataset tended to exacerbate said bias, interpreting african american dialects as hate speech or offensive.

Chapter 3

State of the Art

This chapter presents all relevant and important advancements and developments related to illicit message detection using AI and Twitter data, both without transformers (for contextualization and review of historical and alternative techniques) and with transformers, which is the focus of this work.

3.1 Approaches for NLP using Twitter data

As Twitter has abundant publicly available data and real-time nature, it has been used for NLP tasks in recent years, developing different approaches, including traditional, deep learning-based, and transformer-based methods.

3.1.1 Without Transformers

One of the earliest studies in this area was published by Go et al. [45] in 2009, which proposed a feature-based approach for sentiment analysis on Twitter messages that contained emoticons. The messages were classified as either positive or negative with keywords, using ML algorithms such as naïve-Bayes classification, maximum entropy and support vector machines (SVM). This study demonstrated the feasibility of using Twitter data for NLP tasks and laid the foundation for future research. Since then, a number of other studies have also explored the use of Twitter data for NLP tasks. For example, Pak and Paroubek (2010) [46] presented a machine learning-based approach for sentiment analysis

in Twitter data, which achieved promising results on the dataset that the authors collected. Similarly, Kouloumpis et al. (2011) [47] proposed a lexicon-based approach for sentiment analysis on Twitter data, using interpretation and scores for certain words and hashtags aside from using emoticons, which also demonstrated good performance, showing that the use of interpretative scores for words can be an improvement for sentiment analysis.

In the last decade, there have also been approaches for NLP in Twitter using deep learning. One such approach is the work of Dos Santos and Gatti [48], which developed a deep CNN that examines the component parts of sentences from two datasets of tweets from Stanford, to perform sentiment analysis of short texts. The authors obtained state-of-the-art results for single-sentence sentiment prediction in binary classification. Also, in 2016, Weerasooriya et al. [49] also worked on improving ML techniques for extracting essential keywords from a tweet. The framework was tested using 6 test cases, each consisting of a human keyword generator and a supervisor. The authors obtained state-of-the-art results on the Turing Test score of the system.

In recent years, more research has been on NLP and sentiment analysis using neural networks. Such is the work of Jianqiang and Xiaolin and Xuejun [50], which proposed a "word embeddings method (...) using latent contextual semantic relationships and co-occurrence statistical characteristics between words in tweets". These embeddings are combined with other classical sentiment analysis techniques to form a sentiment feature set of tweets. This is then integrated into a deep convolution neural network to train and predict sentiment classification labels. The authors found that their approach outperformed current techniques in sentiment analysis.

In 2021, two other papers were published regarding NLP using neural networks. The first one was made by Gharge and Chavan [51], which proposed a new method for detecting spam on Twitter based on two aspects: identifying spam tweets without knowing the user's previous background and analyzing language for detecting spam on Twitter in trending topics. The authors did this by collecting tweets related to certain trending topics and labeling them as spam texts or not. The labels of these tweets were extracted using language models, and finally, they were classified using the SVM architecture. The authors obtained an accuracy of 97% on the classification of tweets.

The other work is another approach to sentiment analysis using Twitter data. The

authors focused on analyzing the sentiment around the situation during the pandemic years, mainly how people felt about the different COVID-19 vaccines developed during the pandemic [52]. The authors collected a high amount of raw tweets and preprocessed them using NLP, which were then passed to a supervised k-nearest neighbor (KNN) classification algorithm. This KNN algorithm classified the data into positive, negative, and neutral sentiments. The authors found that for that time, the percentage of positive and negative perceptions of COVID-19 vaccines were almost the same, showing that the view on vaccines from general people is evenly divided.

3.1.2 With Transformers

Considering the improvements that transformers bring, in recent years, there has been a significant amount of research on using transformers for NLP tasks on Twitter data, such as sentiment analysis, spam detection, fake news detection, and malicious tweets detection.

For example, in 2020, Naseem et al. [53] published an article that presented a transformer-based model for sentiment analysis of short tweet, a challenging task as these tweets are informal, noisy, and rich in language ambiguities. The model executed encoding and application of deep intelligent contextual embedding to enhance the tweets, removing noise and considering different aspects like word sentiment, polysemy, syntax, and semantic knowledge. The framework also used bidirectional LSTM to determine the final sentiment of a tweet. The authors tested the framework and found that it outperformed state-of-the-art methods in sentiment classification.

Another work from the same year is the article of González, Hurtado, and Pla [54], which developed a model for irony detection of tweets using transformers. This was done by contextualizing pre-trained Twitter word embeddings based on how BERT works. The authors evaluated the model on two datasets of text data, one for English and another for Spanish, obtaining the best results in Spanish and the second-best results in English.

In the same vein, Mutanga, Naicker, and Oludayo [55] published a paper that presented a transformer-based method of detecting hate speech in tweets, focusing on a parallelization approach. The method was compared against attention-based recurrent neural networks and other transformer baselines for hate speech detection in Twitter documents, outper-

forming baseline algorithms while allowing parallelization.

Two years later, in 2022, Khan, Razzak, Dengel, and Ahmed [56] approached experimented with different Transformer models and how they understood tweets, a task complicated by the emojis, links, hashtags, and other tweet-exclusive text components. The authors experimented with how nine different Transformer models interpreted tweets that contained mention of health-related terms and how the models identify if said tweets use the health-related terms as a way to describe diseases, problems, or general medical situations and conditions or if the terms are not used for a medical context. The authors found that the RoBERTa architecture was the best one, achieving an F1 score of 93%. Table 3.1 shows a summary of the papers considered in Section 3.1.

Table 3.1: Summary of Transformer and non-Transformer Approaches for NLP with Twitter data

Authors	Year	Techniques	Number of tweets	Metrics	Score
Go et al. [45]	2009	naïve-Bayes classifier, maximum entropy and SVM	359	prediction accuracy	80% - 85%
Pak and Paroubek [46]	2010	naïve-Bayes classifier, conditional random field and SVM	216	prediction accuracy and F1 score	higher than 80%
Dos Santos and Gatti [48]	2014	CNN	223154	prediction accuracy	85% - 86%
Weerasooriya et al. [49]	2016	Stanford CoreNLP	258	Turing test score	83.33%
Jianqiang, Xiaolin and Xuejun [50]	2020	Transformer, mixed pooling	40000	training and testing accuracy	67% in English and 78% in Spanish
Naseem et al. [53]	2020	Transformers + LSTM	50167	prediction accuracy	94% - 96%
González, Hurtado, and Pla [54]	2020	Transformers	7792	F1 score	70% - 74%
Mutanga, Naicker and O [55]	2020	Transformer	24783	prediction accuracy and precision	92% and 75%
Koulompis et al. [47]	2021	n-grams and part-of-speech tagging	607966	prediction accuracy and F1 score	65% - 75%
Gharge and Chavan [51]	2021	SVM	70000	prediction accuracy	97%
Shamrat et al. [52]	2021	KNN	30000	does not apply	does not apply
Khan, Razzak, Dengel, and Ahmed [56]	2023	Transformer	15742	F1 score	93%

3.2 Illicit Message Detection in Twitter

While NLP in Twitter can be useful for detecting how people feel according to what they write and can also be used to detect certain elements, such as spam texts or fake news, it can also be vital for detecting and revealing more malicious activity and content in the platform, such as illegal drug sales, human trafficking, child pornography, and other crimes.

As before, methods exist to achieve that, using deep learning and CNNs, and transformers.

3.2.1 Without Transformers

In 2018, Kumar, Kshitiz, and Shailendra published a paper that determined ways to identify bullying in different tweets using a streaming API. They proposed an algorithm that identified aggressive comments as a previous step to classification architectures [57]. The architectures used by the authors are logistic regression, SVM, random forest, and gradient boosting machines. While this work did not focus on illegal activities, it is nonetheless an application of NLP in Twitter.

In the same year, Mackey et al. developed an ML framework that analyzed numerous streams of tweets to accurately detect the marketing and sale of opioids by illicit online sellers via Twitter [58]. The tweets were filtered using common prescription opioid keywords. An unsupervised ML–based approach was developed to summarize the tweets and isolate the clusters associated with illegal online marketing and sale. This was done using the biterm topic model (BTM) technique. The isolated tweets were analyzed to see if they contained hyperlinks associated with illegal online sellers.

A paper focused on a different illegal activity is the work of Hernández and Granizo, aimed at detecting tweets related to human trafficking activities [59]. The authors developed a method that used NLP and image processing techniques. The system has two phases: the first one captures Twitter messages that are suspicious of being related to the crime according to the hashtags, and the second one in which the system recognizes gender and age groups using facial features and or upper body geometry and proportions. Both phases are powered by using the SVM algorithm. The authors obtained accuracies higher than 80% on the recognition tests they performed.

Another paper that worked on detecting human trafficking is the article of Bilal et al. [60]. This paper proposes a generalized approach for detecting and categorizing darknet traffic using deep learning, evaluating various feature selection techniques and machine learning algorithms, including decision trees, gradient boosting, random forest regressor, and extreme gradient boosting (XGB), to select the optimal features for darknet traffic detection. The authors then apply modified convolutional LSTM and convolutional gra-

dient recurrent unit (GRU) deep learning techniques to recognize network traffic more accurately. The results show that the proposed approach outperforms existing methods with a maximum accuracy of 96% for darknet traffic detection and 89% for darknet traffic categorization using XGB as the feature selection approach and CNN-LSTM as the recognition model.

3.2.2 With Transformers

While transformers are a powerful tool for NLP, and Twitter can provide ample datasets, there is not a lot of research aimed specifically at Twitter data and detecting illicit messages using transformers, with general research more focused on detecting hate speech.

One such work is the paper written by Stappen, Brunn, and Schuller (2020), in which the authors developed a framework based on frozen, pre-trained transformers to examine cross-lingual zero-shot and few-shot learning, in addition to uni-lingual learning, on the HatEval challenge data set [61]. The main improvement of the framework is the classification block called AXEL. AXEL can efficiently condense task-specific representations from a sequence of general text representations obtained from a Transformer Language Model (TLM). Its capabilities derive from adopting recent state-of-the-art attention modules used in image super-resolution tasks for text representation compression. The authors obtained results of 71.65% in the F1 score for this model.

Similar work was carried out by Mozafari, Farahbakhsh, and Crespi in 2022. This paper proposes a meta-learning approach for detecting hate speech and offensive language in low-resource languages (with few datasets to work with) [62]. The lack of sufficient labeled data in these languages and the inconsistent generalization ability of transformer-based language models make detecting abusive online content challenging. The meta-learning approach leverages optimization-based Model-Agnostic Meta-Learning (MAML) and metric-based (Proto-MAML) models. The framework is applied in cross-lingual few-shot hate speech detection. Its performance is evaluated using two separate collections of publicly available datasets: 15 datasets across eight languages for hate speech and six datasets across six languages for offensive language. The results show that the meta-learning-based models outperform transfer learning-based models in most cases. Proto-

MAML is the best-performing model, as it can quickly adapt to new languages with little data.

Going back one year, in 2021, Huertas-García et al. developed a different method for detecting hate speech by using transformers to profile the authors of those speeches [63]. The authors used datasets in the English and Spanish languages. The system uses transformer-based models as feature extractors at the tweet level in combination with mixed pooling techniques. This approach focuses on author-specific embedding regarding their tweets, which are later fed to an ML classifier. The authors also explore using features from other transformer-based models, sentiment analysis techniques, and hate lexicons to boost the feature extraction process, obtaining accuracies of 67% and 78% in the English and Spanish test datasets, respectively. Table 3.3 shows a summary of the papers considered in Section 3.2.

Table 3.3: Summary of Transformer and non-Transformer Approaches for Illicit Message Detection

Authors	**Year**	**Techniques**	**Number of tweets**	**Metrics**	**Score**
Kumar, Kshitiz, and Shailendr [57]	2018	logistic regression, SVM, random forest and gradient boosting	2235	AUC and cross-validation	55 - 65%
Mackey, Kalyanam, Klugman, Kuzmenko and Gupta [58]	2018	biterm topic model	213041	not used	not applicable
Stappen, Brunn, and Schuller [61]	2020	Transformer	19600	F1 score	71.65%
Hernández and Granizo [59]	2021	SVM	55123	prediction accuracy, recall and F1 score	80%

Sarwar, Hanif, Talib, Younas and Sarwar [60]	2021	extreme gradient boosting, convolutional gradient recurrent unit, LSTM	141534	prediction accuracy, recall and F1 score	96% for darknet traffic detection and 89% for darknet traffic categorization
Anwar [63]	2021	Transformer, mixed pooling	40000	training and testing accuracy	67% in English and 78% in Spanish
Mozafari, Farahbakhsh and Crespi [62]	2022	MAML and Proto MAML (Transformers)	206453	F1 score	60% - 70%

Chapter 4

Methodology

This chapter deepens into the research problem and presents the general road map of implementing the illicit message detection framework that will be used for this work.

4.1 Proposal

Figure 4.1 shows a diagram of the proposed process, including the detection framework. First, a set of tweets is gathered using the Twitter API for Python, enclosing the search using a query that contains relevant hashtags and keywords related to human trafficking and child prostitution activities. After that, a noise removal process is done as not all tweets using the keywords and hashtags are negative or suspicious. In this step, all unrelated tweets are deleted. Then, the remaining tweets are cleaned. This means that duplicate tweets are removed.

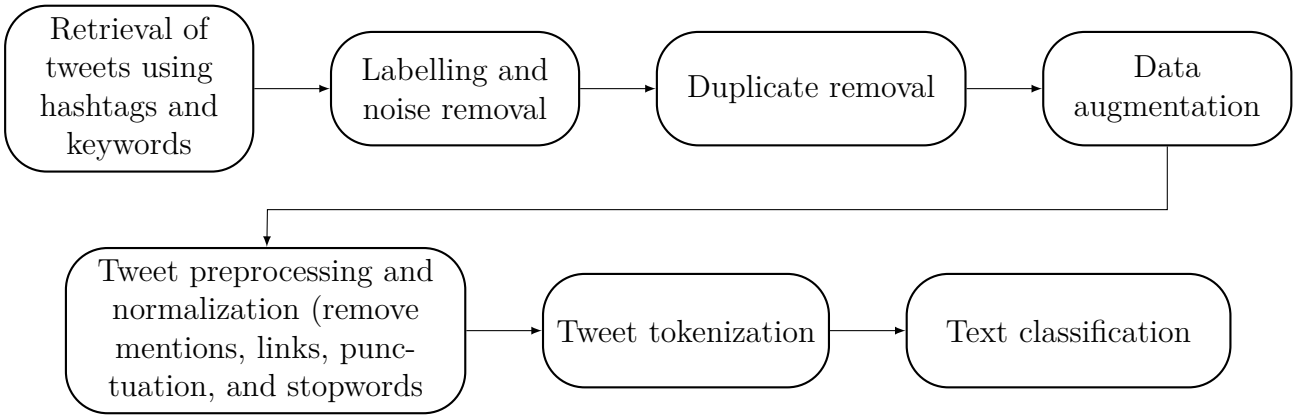


Figure 4.1: Diagram of the entire process for this work

With only relevant tweets remaining, tweet preprocessing is carried on. In this step, only relevant information is preserved; this means that mentions, links, punctuation, and stopwords are removed from the tweets. Only the plain text and the hashtags are preserved, as they are the most important information in the tweet. In the last step before training, the tweets are tokenized, which means that the important information about the text is separated, extracting its main features, such as verbs, adjectives, and special terms. Finally, the processed dataset is fed into the models. Two groups of models were used: four Transformer-based models and two non-Transformer-based models. Each step is detailed in the following subsections.

4.1.1 Retrieval of Tweets

The retrieval was done using the `twarc`[1] module on Python, which allows the use of pagination in order to bypass the tweet limit that Twitter imposes on queries. All the tweets retrieved were in Spanish. The retrieval query included hashtags and keywords associated with child pornography and human trafficking, such as #cp (child pornography), #loli, hashtags and keywords related to young women as #joven (young), #fresca (fresh), niña (little girl), peladitas (colloquial term for young girls), and coupling those terms with sexual terms such as sexy. The timeframe for the query was one year, from June 2022 to June 2023. This query resulted in an initial dataset of 16995 tweets.

4.1.2 Labelling and Noise Removal

The initial dataset was labeled using three numerical labels. These labels are:

- **Label 0 - Sexual related but not suspicious:** These are tweets that refer or are directly about sex, sexual services, sexual material, etc., but show no indication of being related to human trafficking, child pornography, child sexual exploitation or child sexual abuse. These tweets include the promotion of sexual services or the selling of sexual content (Onlyfans, escorts, etc.) when the tweet author is an adult and is promoting voluntarily, sexual roleplay, denouncement of possible sexual

[1]https://github.com/DocNow/twarc

crimes, accusing other people of possessing, watching or promoting legal or illegal sexual content, and sexual comments towards another user.

Example: @LuzPavicich Feliz dia niña hermosa y sexy 🌻😘😍 (@LuzPavicich Good day you gorgeous and sexy girl 🌻😘😍).

- **Label 1 - Suspicious of human trafficking, child pornography, or illegal sexual services:** These tweets show clear indications of possible human trafficking, production, possession, or distribution of child pornography, and underage exploitation and abuse. These include the promotion of underage sexual services, sharing of child pornography, and sharing of chats or groups in social networks (such as WhatsApp, Telegram, or Reddit) that distribute underage pornographic material or offer illegal sexual services.

  Example: Alguien cerquita del hotel Xanadú...! #lolita #disponible #cdmx #promo 5529472781 $1,000 Una Hora (Anyone near Xanadu hotel...! #lolita #available #cdmx #promo 5529472781 $1000 for one hour).

- **Label 2 - Unrelated:** These are the tweets that are not related at all to the problem and are not necessary for the dataset. These include tweets that use the keywords and hashtags in an unrelated context, such as using cp as código penal (criminal code) or código postal (postal code) instead of child pornography, tweets with mentions to other users that have keywords in their username, tweets that talk about topics outside of human trafficking, child pornography and underage sexual exploitation, spam and promoted tweets.

  Example: Esta es la nueva camiseta del Sporting CP para la temporada 2023-24 (This is the new Sporting CP jersey for the 2023-24 season).

The tweets labeled as 2 are basically noise and are not useful for classification, so they were deleted to clean the dataset. This iteration of the dataset had 6312 tweets.

4.1.3 Duplicate Removal

The last cleanup of the dataset was to find and remove duplicate tweets, as they inflated the dataset unnecessarily. This was done by "atomizing" the tweets, which means removing

hashtags, emojis, special characters, line breaks, whitespaces, punctuation, and everything that is not plain words. The result is a list of single strings, each string being each word of the processed tweet joined. After atomizing the tweets, the duplicate tweets are removed using the `pandas` module, a task made easier thanks to the atomization step. After this process, the final dataset had 5360 tweets. Figure 4.2 shows the number of observations for each of the two classes in the dataset.

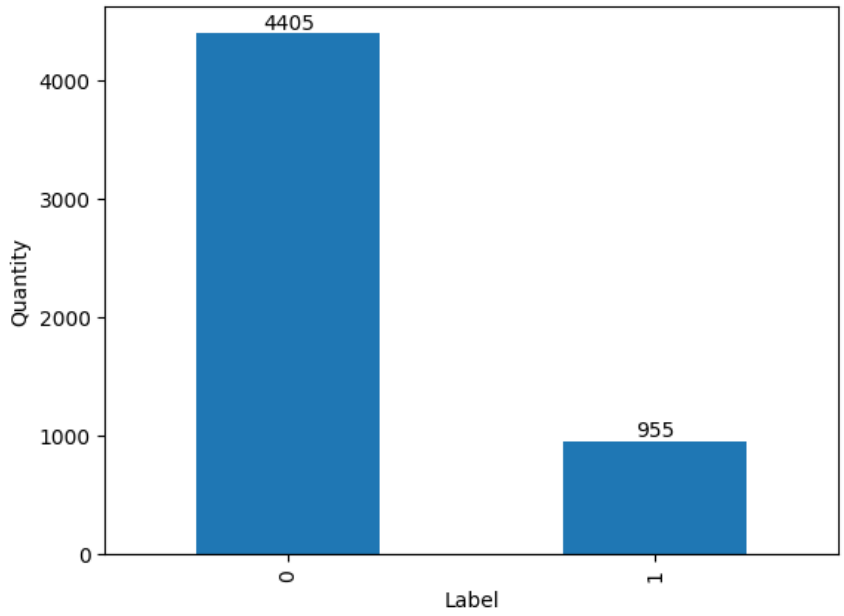


Figure 4.2: Quantity of observations for each label in the dataset

4.1.4 Data Augmentation

As seen in the previous section, there is a heavy imbalance between the classes, down to a proportion of nearly 5:1 of class 0 over class 1. Class imbalance is a common problem in AI and ML that causes models to present biases against minority classes and thus reduced accuracy when predicting them [64]. While some levels of imbalance can be acceptable depending on the problem and the AI models used, this dataset is too imbalanced to produce good results. Thus, the dataset has to be augmented.

Data augmentation is a technique used to artificially increase the size of a dataset by modifying or transforming the data in different ways. This is done in order to correct class imbalance, reduce the biases of AI models, and allow for better generalization [65].

The dataset was augmented using the `nlpaug` Python library [66], which specializes

in text data augmentation for NLP tasks, and has configurations to execute data augmentation on Spanish texts. Concretely, synonym data augmentation was used, in which each tweet is read, and every word that can be replaced by a synonym is replaced. This process was done two times to the label 1 tweets in order to obtain a considerable amount of observations without exaggerating the quantity of synthetic data, even if the data is not completely balanced, as that can be actually detrimental to the performance of the model. This incremented the total length of the dataset to 7270 tweets. Figure 4.3 shows the new distribution of the labels.

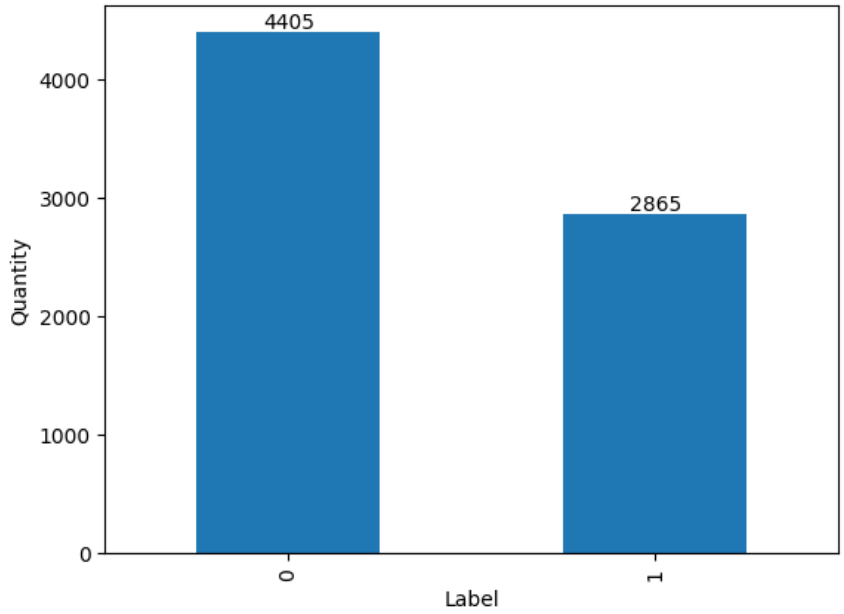


Figure 4.3: Quantity of observations for each label in the augmented dataset

For better comparison and insight into the performance of every model, two experiments will be done with each model: one with the original dataset and the other with the augmented dataset.

4.1.5 Tweet Preprocessing and Normalization

Now that the dataset is constructed, the tweets must be preprocessed to be easy to classify for the detector. This step is similar to duplicate removal, as the tweets have to be heavily edited. Concretely, the tweets are converted to lowercase normalized to eliminate accents and special characters, the mentions and the links are removed as they are unimportant,

and punctuation, extra whitespace, stopwords, and linebreaks are also deleted. In the end, the pure normalized text and the hashtags remain as they are the most important content of the tweets.

For example, the tweet "@ValeLovee25 @BustiDelBlog @RecMexLin @Gengy1818 @rt_sexys @SeguirBellas @Azulito10 @Luiz_Garfiel @costeradelamor @LasScorts @lechero070 @RecomiScort2 @CdmxCasuales @MasLobitas @sex_escortmx @Sexy_Promotions Ey te cotizas niña 🍓" turns into "ey te cotizas nina". The accents, mentions and emojis were removed by the preprocessing, and the text was lower-cased.

4.1.6 Tweet Tokenization

After preprocessing the data, an additional process called tokenization needs to be performed. Basically, tokenization breaks a given piece of text into smaller units called tokens, which can be words, phrases, characters, or subwords [67]. This way, complex strings and sentences can be separated into their main components, and the language models can perform better at recognizing the text [68].

Transformer Models

One important thing to note when working with Transformers is that they do not have a generalized tokenization method. While most convolution and recurrence-based models can work with the same tokenization techniques, each specific transformer architecture has its own tokenization process and work only with that. Some of the tokenization techniques used by transformers are: WordPiece, which breaks words into subwords based on frequency and combines them to produce unrecognized words [69, 70], byte-pair-encoding, which is similar to WordPiece but stores more information in the tokens [71, 72], and Unigram, which considers each word as a token disregarding frequency and removes tokens until the number of tokens is adequate [73].

Non-Transformer Models

Since non-Transformer models do not need to be tokenized in particular ways, a streamlined and common method was used for both models. The dataset was tokenized and vectorized

using the `TextVectorization` layer from `keras`, which split the tweets and assigned each token a numerical value according to the vocabulary and frequency of the tokens. In the end, it is the same process used in Transformer tokenizers.

4.1.7 Text Classification

Transformer Models

For this work, four different transformer models were used:

1. **BERTweet:** A BERT-based model pre-trained on a dataset of English tweets that outperformed other BERT-based models like RoBERTa [74] and XLM-RoBERTa [75] on tweet classification [76]. While the model is pre-trained in English tweets instead of Spanish ones (which make the dataset), BERTweet was chosen because being trained on Twitter data meant that the model could be more precise at classifying tweets than other transformer models.

2. **DistilBERT:** Another BERT-based model with the objective to be small, fast, cheap, and light. It has "40% less parameters than bert-base-uncased" [77], and "runs 60% faster while preserving over 95% of BERT's performances" [77]. For this project, a DistilBERT model trained and optimized for Spanish was used [78].

3. **XLM-RoBERTa:** Another BERT-based model that was trained in 100 different languages and is capable of recognizing the language of a text by itself, outperforming base multilingual BERT [75].

4. **GPT-2:** Transformer model trained on a dataset of web pages that contains 1.5 billion parameters, focused on text prediction and generation, but it is also suitable for other tasks, such as text classification [35]. This model was also trained with a dataset of different languages, but the specific model used for this work was adapted for Spanish, so it was chosen for this work.

Every model was implemented using the `huggingface` API[2], which facilitates the creation, deployment, and training of the models. Thanks to how transformers are implemented and how the central parts of the models are focused on reading and interpreting

[2]https://huggingface.co

text, they are very flexible for different tasks, even tasks they were not originally developed for. This means that every model can be repurposed for the task of this work, text classification, and also that the models can be fine-tuned using the dataset that was constructed, all of this without having to re-train the model from scratch and obtaining good results in a low amount of training epochs. Every model was fine-tuned for ten epochs using the same configurations for better comparison. The loss functions are particular to every model, and all models used the Adam optimizer [79].

Non-Transformer Models

Additionally, for comparison, two non-Transformer models were implemented:

1. **CNN-based model:** A neural network that contains two 1D convolutional layers, one with 64 filters and the other with 32 filters.

2. **LSTM-based model:** A model that contains a single bidirectional LSTM layer in order to improve information processing and retaining, and get better results than using just LSTM.

Both models were implemented using `tensorflow` and `keras`. As the models are not pre-trained, full training from scratch was done for both. For this reason, the models were trained on 50 epochs instead of 10. The loss function used was binary cross-entropy, and the optimizer was Adam, as with the Transformer models.

Chapter 5

Results and Discussion

This chapter presents the results obtained in the two main groups of experiments: experiments without data augmentation and experiments with data augmentation. All six models were tested in both conditions.

5.1 Analysis Method

Four metrics were measured and considered to compare the results of all the architectures tested and determine the best model: test accuracy, precision, recall, and F1 score.

5.1.1 Accuracy

Accuracy, in short terms, is an indicator of the correctly predicted instances out of the total instances in a dataset. More specifically, accuracy is calculated as the ratio of true positive (TP) and true negative (TN) predictions to the total number of predictions, i.e., true positives, false positives (FP), true negatives, and false negatives (FN) [80]. This means that the formula for accuracy is:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Accuracy assesses the overall performance of a model, i.e., how many predictions were correct in total. This is useful when the classes in the dataset are balanced, but if the dataset is imbalanced, the accuracy can be misleading, even more when the imbalance is

heavy [81]. In such cases, other metrics like precision, recall, and F1-score are often used to understand the performance of the model better.

5.1.2 Precision

Precision is an indicator of how many positive predictions are correct regarding the total number of positive predictions the model did. More specifically, it is the ratio of true positive predictions over the sum of true positives and false positives [82]. This means that the formula for precision is:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Precision quantifies the ability of the model to avoid making false positive predictions, which is useful when the dataset is imbalanced. High precision indicates that when the model predicts a positive outcome, it is likely to be correct [81].

5.1.3 Recall

Recall is the metric that measures the capacity of the model to identify all positive instances on the dataset. It is the proportion of true positive predictions relative to the total number of positive instances (true positives and false negatives) [82]. This means that the formula for recall is as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Recall quantifies the ability of the model to avoid missing true positives. A high recall value means the model is good at detecting positive instances, though it does not account for false positives. This is why recall is usually measured along with precision, to have a better insight into the performance of the model [81].

5.1.4 F1 score

F1 score is a metric that combines precision and recall to generate a balanced measure of the performance of a model. Concretely, it is the harmonic mean of precision and recall,

calculated with the formula [82]:

$$\text{F1 Score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

The F1 score is especially valuable when dealing with imbalanced datasets as it considers the metrics that fixate on both false positives and false negatives, rewarding models that minimize those instances [83]. Thus, a high F1 score means that the model identifies true positives correctly, and there is a low number of false positives and false negatives [81].

5.2 Results Without Data Augmentation

The non-augmented dataset was split into three sets: training, validation, and testing, with a split of 70%, 12%, and 18%, respectively. In raw numbers, the training split has 3752 tweets, the validations split has 643 tweets, and the test split has 965 tweets. Table 5.1 shows a summary of the metrics on all models.

Table 5.1: Summary of the results of each model. Best results highlighted

Model	**Training Accuracy**	**Training Loss**	**Test Accuracy**	**Test Loss**	**Precision**	**Recall**	**F1 Score**
BERTweet	92.54%	9.42%	92.33%	25.64%	84.31%	72.07%	**77.71%**
Distil-BERT	90.82%	3.37%	**92.44%**	**23.93%**	**87.86%**	68.72%	77.12%
XLM-RoBERTa	92.22%	9.00%	**92.44%**	25.56%	85.81%	70.95%	77.68%
GPT-2	92.38%	13.54%	89.74%	29.58%	73.81%	69.27%	71.47%
CNN model	**99.60%**	**0.64%**	79.59%	99.04%	63.34%	**79.59%**	70.54%
LSTM model	99.47%	0.86%	79.59%	99.12%	63.34%	**79.59%**	70.54%

In regards to training, non-Transformer models got the best performance, with the CNN-based model achieving 99.6% accuracy and the LSTM-based model obtaining 99.47% accuracy. It seems that these networks can rapidly adapt to the training set. Next, we have the Transformer models, in which BERTweet got the best performance, with an accuracy of 92.54%. Even so, the other architectures have accuracies ranging from 90% to near

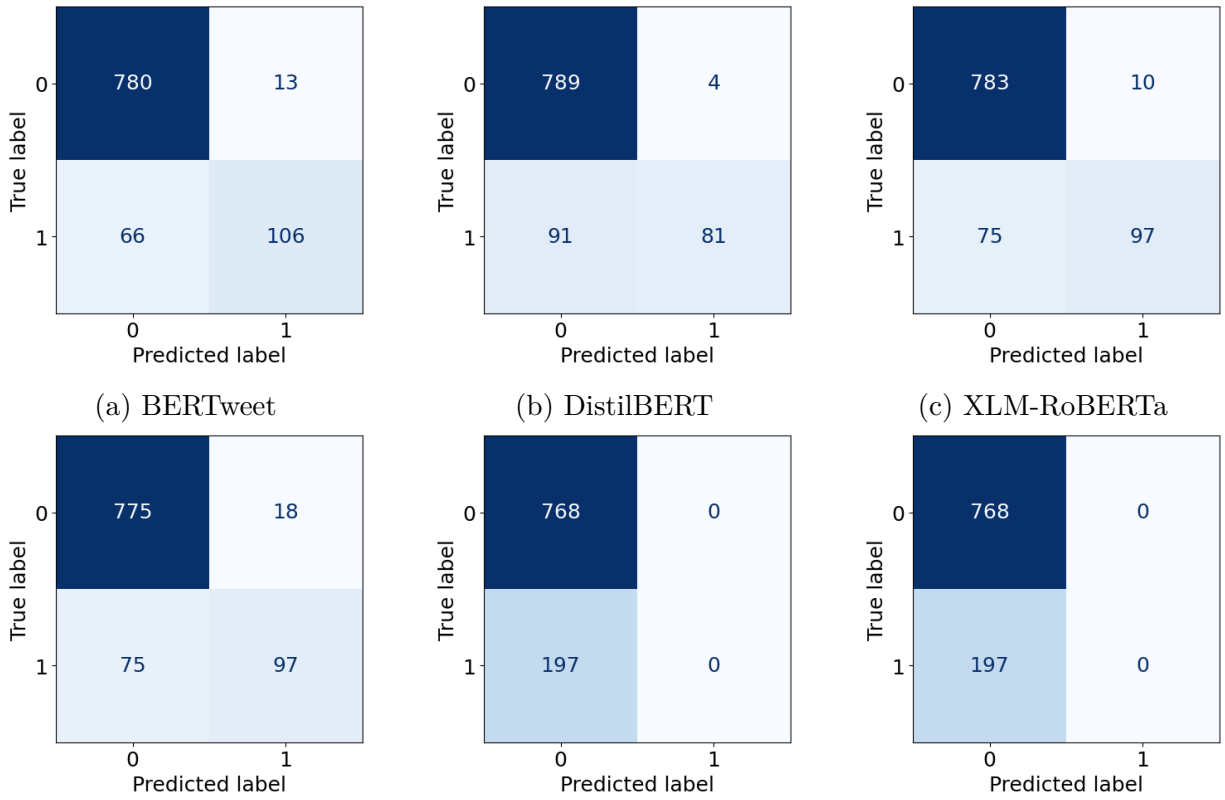
92.5%, showing the versatility of Transformers that are capable of getting high metrics in a few epochs. Also, as BERTweet specializes in Twitter data, the model is able to distinguish the particularities of tweets better, even when the model has not been trained in the Spanish language. The losses are also balanced, with DistilBERT getting 3.37% of training loss. GPT-2 has the poorest performance of all models, with a 13.54% loss, which can be because GPT-2 is focused on text generation rather than text classification.

While training gives good insight into the models, testing is the important step in which they truly demonstrate their capabilities. As seen in Table 5.1, the best architectures in this regard are DistilBERT and XLM-RoBERTa, both getting a value of 92.44%. As explained in the previous chapter, these models are capable of recognizing the Spanish language (the DistilBERT model used was actually pre-trained with Spanish language content) and thus are better at understanding the tweets and classifying them. BERTweet is in second place with 92.33%, a good result that demonstrates its adaptability to tweets. While the previously mentioned Transformer models got high accuracies, GPT-2 is the exception, with a score of 89.74%. This result can be attributed to the small size of the dataset and its generative nature. Finally, the non-Transformer models got the lowest accuracy, both with a value of 79.59%.

The test loss shows that the models struggle in classification, mainly the non-Transformer models. In their case, the LSTM-based model obtained a loss of 99.12%, making the model completely unreliable for the task, as any new data fed to the model will probably be misclassified. It is the same case for the CNN-based model, with a loss of 99.04%, being almost as unreliable as the LSTM model. The Transformer-based models have better results but are not completely optimal. In this group, GPT-2 has the highest loss, with a value of 29.58%, following the trend observed in the other metrics. BERTweet and XLM-RoBERTa go next, with similar values of 25.64% and 25.56%, respectively. Finally, DistilBERT obtained the best results in that regard, with a test loss of 23.93%, showing the advantages of using models pre-trained in the Spanish language. Even so, it is important to note that while Transformer models got the lowest loss values, they are still not low enough to be 100% reliable. But in the end, this information, coupled with the test accuracies, is enough not to discard these models and consider them for real-world use.

For the analysis of the remaining metrics, it is important to examine the confusion

matrices of all models along with the values in Table 5.1 in order to understand how each model handles positives and negatives truly. For this purpose, Figure 5.1 presents the confusion matrices produced by all the models.



(a) BERTweet (b) DistilBERT (c) XLM-RoBERTa

(d) GPT-2 (e) CNN-based model (f) LSTM-based model

Figure 5.1: Confusion Matrices of all the models

Let us examine every model in order, starting with BERTweet. If we examine the confusion matrix produced by the model (Figure 5.1a), we can see that the model is very good at predicting class 0, with 780 true negatives that represent 98.36% of all negative predictions. In the case of positives (class 1), the model correctly predicted 106 instances that represent 61.63% of all positive predictions. This means that the model is not very good at predicting the class of interest for this problem, and that because of class imbalance, the model is heavily biased towards class 0, struggling to predict class 1.

This behavior is confirmed by the precision, recall, and F1 score of BERTweet, metrics that give insight into how a model performs when a dataset is imbalanced. As explained in the previous section, precision is an indicator of the ability of the model to avoid false positives. The value obtained by BERTweet, 84.31%, indicates that, while the model has

an acceptable probability of correctly predicting both classes, it is not reliable enough for the problem at hand because class 1 is not correctly classified in all instances. Next is recall, the metric that quantifies if the model is good at detecting all positive instances. BERTweet obtained a value of 72.07% in this metric. This means that the class imbalance causes the model to struggle to identify all the correct instances of both classes, as shown in the confusion matrix of BERTweet. This means that the model is not fit to detect all the positive instances, a very important trait for this problem. Finally, the F1 score, which balances precision and recall, determining if a model is capable of identifying true positives and minimizing false positives and false negatives. BERTweet achieved an F1 score of 77.71%, the best score in comparison to the rest of the models. This complements the other metrics in showing that the model is not apt enough for the task at hand because of its problems detecting and classifying true positives.

In the case of DistilBERT, the model is even more precise in predicting class 0, with a 99.50% true negative ratio, but more unfit in predicting class 1, with a 47.09% true positive ratio. This means that the model is more flawed than BERTweet for the task at hand. The precision value, 87.86%, is the highest of all models, which is probably caused by the higher precision at classifying class 0. Nevertheless, as mentioned, the model is still unfit to detect class 1 correctly. The 68.72% recall value gives a better insight into the performance of the model, verifying its unreliability on the task caused by class imbalance. The F1 score metric of 77.12% also shows the poor performance of the model.

XLM-RoBERTa shows a similar performance to DistilBERT, both in its confusion matrix and its metrics. It has a 98.74% true negative ratio and a 56.40% true positive ratio. A bit better overall than DistilBERT, but not enough. The precision (85.81%) is lower than DistilBERT, meaning the model is also affected by class imbalance. The recall (70.95%) is higher, as the model is a bit better at detecting positives. Finally, the F1 score (77.68%) is similar to the score of DistilBERT, showing that overall, their performance is similar. This is also indicated by both models having the same test accuracy.

GPT-2 is the last Transformer model on the list, with a lower performance in positive and negative classification. It has a 97.73% true negative ratio and a 56.40% true positive ratio (same as XLM-RoBERTa). The precision (73.81%) of GPT-2 is the lowest in Transformer-based models, further proving the disadvantages of using a model created for

text generation to do text classification, along with the problems from class imbalance. The recall (69.27%) follows the same trend, being the lowest value of all models, meaning the model has the most problems at detecting positives. This trend continues with an F1 score of 71.47%, also the lowest value in Transformer-based models for this score.

Finally, there are the CNN-based and LSTM-based models, which share the same confusion matrices and metrics of precision, recall, and F1 score. These models are the worst-performing in the entire list, with the exception of recall. They have a 100% true negative ratio and a 0% true positive ratio, making the models completely unfit for the problem, as they are incapable of detecting and classifying tweets suspicious of human trafficking and child pornography (class 1). Their precisions (63.34%) are the lowest of all models, meaning that the models struggle to detect true instances, concretely, true positives. The recall values (79.59%) are better. Actually, both models have the highest recall, but these values are most probably influenced by the 100% true negative rate, meaning that in this case, the high recall does not mean that the models are good. The F1 scores (70.54%) support the previous claim.

5.3 Results With Data Augmentation

As the previous section demonstrates, class imbalance heavily impacts model performance, resulting in models that are only capable of detecting the majority class while struggling to detect the minority class, which is generally the class of interest. Thus, data augmentation is needed to alleviate this issue, generating a more balanced dataset. This section presents the results obtained when the augmented dataset was used. This dataset was split using the same portions as the non-augmented dataset: 70% for training (5089 tweets), 12% for validation (872 tweets), and 18% for testing (1309 tweets). Table 5.3 shows a summary of the metrics on all models.

From the start, we can see the improvements data augmentation brings to the training metrics, mainly in the Transformer models. The CNN-based model and LSTM-based model are the best-performing again, with 99.41% and 99.33% training accuracy, respectively. Next is DistilBERT, the best-performing Transformer model, with a training accuracy of 96.10%, followed by BERTweet and RoBERTA, which share a value of 95.41%. Thanks

Table 5.3: Summary of the results of each model, using data augmentation. Best results highlighted

Model	Training Accuracy	Training Loss	Test Accuracy	Test Loss	Precision	Recall	F1 Score
BERTweet	95.41%	5.79%	94.88%	26.04%	**93.89%**	93.36%	93.63%
Distil-BERT	96.10%	2.08%	**95.49%**	**22.07%**	93.66%	**95.26%**	**94.45%**
XLM-RoBERTa	95.41%	6.69%	91.75%	26.45%	91.82%	87.29%	89.49%
GPT-2	93.00%	12.31%	89.00%	30.45%	86.20%	86.53%	86.36%
CNN-based model	**99.41%**	**1.06%**	59.05%	53.27%	34.87%	59.05%	43.85%
LSTM-based model	99.33%	1.15%	59.05%	42.05%	34.87%	59.05%	43.85%

to data augmentation, the models seem to be better for the task. Finally, there is GPT-2 with an accuracy of 93%, which had less improvement than the other Transformer models, most probably because of the already mentioned focus that GPT-2 has on text generation and because GPT-2 is made to create synthetic texts, making it identify synthetic texts itself would be a problem for the model.

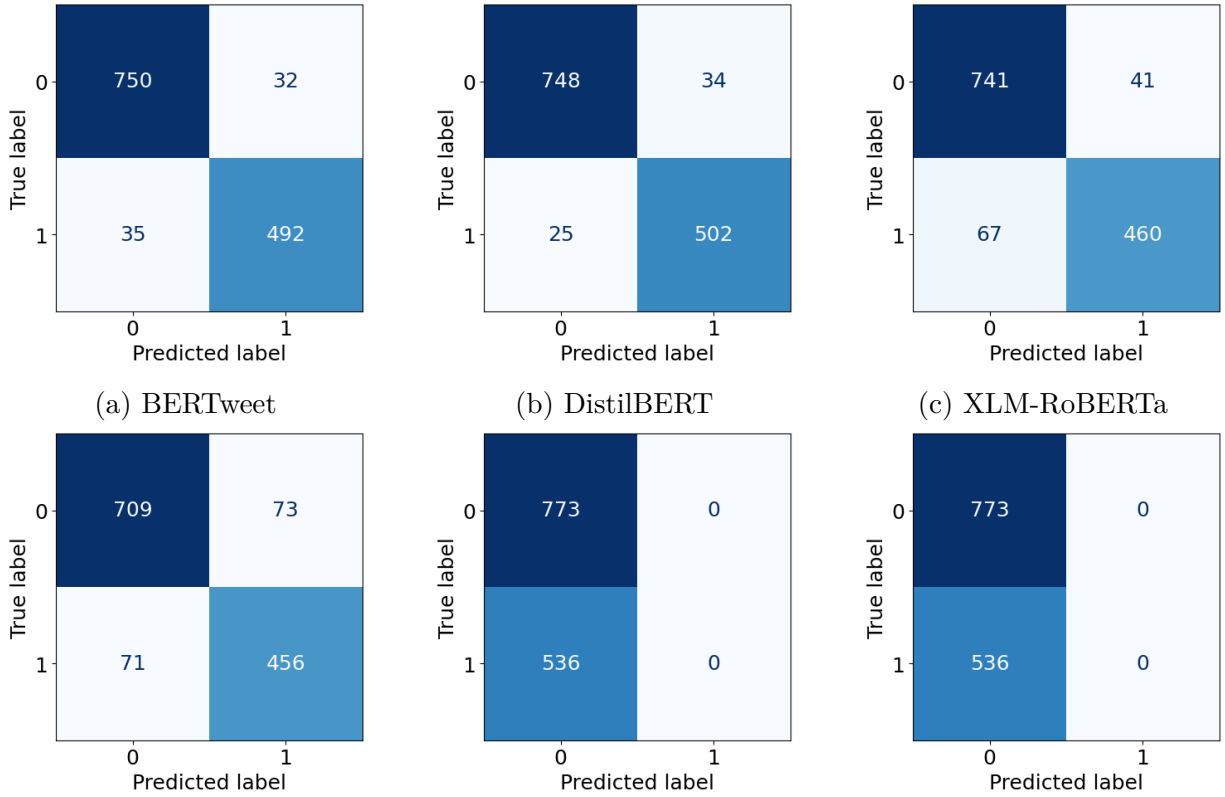
The training loss gives more information on how the models have improved in training. For example, while the CNN-based model and LSTM-based model are the best ones regarding training loss, with values of 1.06% and 1.15%, respectively, they are higher than when data augmentation was not used. This can be a sign of deterioration in learning because of the synthetic data used in this experiment. In the case of Transformer models, all losses where reduced. DistilBERT has a training loss of 2.08%, the lowest in the Transformer models, BERTweet has a loss of 5.79%, going lower than the previous 9.42%, and XLM-RoBERTa achieved a value of 6.69%, reducing the previous value of 9%. These results show that Transformer models have adapted to the augmented dataset and the synthetic data it contains, showing their versatility. Again, GPT-2 improved by a little, with a training loss of 12.31%, in comparison to the previous value of 13.54%, supporting the idea that synthetic data actually harms the ability of GPT-2 to do text classification.

In regards to testing, the best model is, again, DistilBERT, with a test accuracy of 95.49%, improving the previous results obtained without augmentation (92.44% test ac-

curacy). BERTweet is in second place with an accuracy of 94.88%, another improvement considering the previous value of 92.33%. It is evident that both architectures are the most suited for the problem, DistilBERT being capable of processing Spanish texts, and BERTweet having the ability to understand and process tweets. XLM-RoBERTa is in third place, with an accuracy of 91.75%, less than the previous result of 92.44%, but it can be a product of synthetic data. Even so, the result is not too far from the previous one. GPT-2 is, again, the last place in the Transformer models, with a test accuracy of 89%, almost the same as the accuracy obtained in the non-augmented dataset (89.74%). This result reinforces the idea that GPT-2 is not good at classifying synthetic data. Finally, the CNN-based and LSTM-model had significantly worse test accuracies in the augmented dataset, both with values of 59.05%, more than a 20% reduction from the previous results (79.59%). The results may be indicative that these models deteriorate when handling synthetic data and are not capable of adapting to the problem.

The test losses are higher in most cases instead of lower. In Transformer models, only DistilBERT, the model that has the lowest loss, is the exception, with a value of 22.07%, lower than the previous 23.93%. The rest are higher than the values obtained with the non-augmented dataset. This can be produced by data augmentation, as synthetic text data is not always consistent and/or understandable. Another reason is the effect that data augmentation has in class imbalance, now the model is not as good as before for the majority class, so the general loss is a bit higher. In the case of the CNN-model and LSTM-based model, the losses lowered, getting values of 53.27% and 42.05%, respectively. This can be attributed to the synthetic data used to augment the dataset, which incremented the number of instances of class 1, which allowed the models to reduce their losses and bias towards the majority class.

As with the previous section, the remaining metrics will be analyzed along with the confusion matrices of the models. Figure 5.2 presents the confusion matrices produced by the models.



(a) BERTweet (b) DistilBERT (c) XLM-RoBERTa

(d) GPT-2 (e) CNN-based model (f) LSTM-based model

Figure 5.2: Confusion Matrices of all the models, with data augmentation

From the start, we can see how data augmentation has helped the Transformer models to detect positive and negative instances. BERTweet has a true negative rate of 97.02% and a true positive rate of 91.79%. While the true negative rate is a bit lower than with the non-augmented dataset, the true positive rate is significantly higher, going 30% up. This shows that the model is not biased anymore and is capable of detecting both classes with enough confidence. This claim is supported by the precision, recall, and F1 score values of 93.89% (the highest of all models), 93.36%, and 93.63%, respectively. The model has a more stable performance and is good at detecting both true positives and true negatives.

DistilBERT also benefits from data augmentation. It has a true negative rate of 96.77% and a true positive rate of 93.66%, an increment of 46%. This model is better at detecting class 1, most probably because of its ability to understand the Spanish language. Its improved detection performance is validated by its values of precision (93.66%), recall (95.26%), and F1 score (94.45%), the last two ones being the best values for all models. This model has also stabilized thanks to data augmentation and performs better at detecting

true positives.

Following the same trend, XLM-RoBERTa improved in its results. Its true negative rate is 95.86%, and its true positive rate is 85.82%, going almost 30% up. While it is an improvement, the results suggest that this model is not as good as the others for the task at hand because of its lower true positive rate. The values of the metrics follow this idea, with a precision of 91.82%, a recall of 87.29%, and an F1 score of 89.45%, good values considering the previous results with the non-augmented dataset, but not as good as BERTweet or DistilBERT.

GPT-2, again, had the least improvement of the Transformer models. Its true negative rate is 91.72%, and its true positive rate is 85.07%, going almost 30% up from before. While the model showed signs of having problems in text classification and in processing synthetic data, the rates show that the model is still competent for the task, not as good as the other models, but still a good result. DistilBERT is the best Spanish pretrained Transformer model for this task. The values of the metrics verify the trend, with a precision of 86.2%, a recall of 86.53%, and an F1 score of 86.36%, the lowest of all the Transformer models.

Finally, there are the CNN-based and LSTM-based models, which, again, share the same confusion matrices and metrics of precision, recall, and F1 score. From the start, we can see that the capabilities of the models to detect true positives and true negatives have stayed the same as with the non-augmented dataset. They have a 100% true negative ratio and a 0% true positive ratio, making data augmentation ineffective in correcting the detection problems of the models. Their metrics have actually lowered in comparison to before, with their precision being 34.87% (a reduction of almost 30%), their recall being 59.05% (a reduction of 20%), and their F1 score being 43.85% (a reduction of almost 30%). This makes the models completely unfit for the task, as they cannot detect suspicious tweets even with data augmentation. These models seemingly need alternative approaches to improve their performance.

Chapter 6

Conclusions

The main conclusion for this research project is that Transformers are a good suit for natural language processing and can quickly adapt to online expressions. For the specific problem of detecting illicit tweets in Spanish, DistilBERT and BERTweet were the absolute best models, being able to discern non-suspicious and suspicious tweets easily. While the other Transformer models were not as good, that can be attributed to how they were pretrained and how they process data, which means there is room for improvement for those architectures too. Thanks to how Transformer architectures are powered by their pretraining data, any flaws or problems the models present can be corrected by using data specific to the problem in order to improve the adaptability of the models.

It is important to note that DistilBERT and BERTweet are both BERT-based models, which means that, with the appropriate adjustments, we can implement the advantages of BERTweet into DistilBERT or vice-versa. The second option could be the best, as BERTweet is already adapted to tweets and their structure; we would need to replace the original dataset of English tweets in which BERTweet was trained with a new and very big dataset of Spanish tweets. A new comparison between DistilBERT and this Spanish BERTweet would be a good way to determine if this approach is the definitive solution for the problem.

Another conclusion for this work is how Transformers do not suffer as much as other models with class imbalance. While the transformer models had problems in the original, heavily unbalanced dataset, their performance, and most importantly, their ability to discern between classes improved significantly when the augmented dataset was used. And

again, that new dataset was not balanced either. Class 0 had almost double the instances as class 1. Even then, that was not a problem for the Transformer models, while the non-Transformer models actually performed worse. This means that, for other problems and datasets that present imbalance, Transformers can be used without issues and without the need to perform data augmentation, assuming the imbalance is not too high, a consideration that is very important and useful in the field of NLP when most problems suffer from data imbalance by default.

While the CNN and the LSTM models obtained very poor results, it is most likely a byproduct of overtraining and overfitting. Thus, these architectures can also be explored and improved, but they would need specific techniques that would require additional preprocessing before training, such as using trained word embeddings.

While the results have been significant in paving the way to develop AI detection for illicit tweets, future work can be done to improve the detection rate and other metrics and to adapt future models to the problem better. At first, it would be useful to generate a significantly larger and more refined dataset, not just to fine-tune Transformer models but to re-train them from scratch with data that is pertinent to the problem. While the architectures are as powerful as they are, re-training them will definitely improve how they understand illicit tweets, facilitating the process of fine-tuning and the end results. This would be a process that will take a lot of time and computational resources, but the payoff will be worth it.

If it is impossible to re-train the models, a new and larger dataset can also be used to fine-tune them. While the improvement will probably not be as good as re-training, fine-tuning with more concise data will yield better results. Generating this large dataset was one of the objectives for this work, but because of how Twitter changed the terms and conditions on the access and limits of its API[1], the dataset ended up being considerably smaller.

Finally, more advanced and heavy architectures can be tested and implemented, such as the GPT-based architectures, i.e., GPT-3 or GPT-4 (which are not open source yet), GPT-J and GPT-NeoX (which are open source). Other heavy architectures that are not generative can be experimented with as well. Those models require a huge amount of

[1]https://www.cnbc.com/2023/02/02/twitter-to-start-charging-developers-for-api-access.html

resources, memory, and storage, which are out of reach for this work but can produce better and more precise detectors.

Bibliography

[1] F. Bre, J. Gimenez, and V. Fachinotti, "Prediction of wind pressure coefficients on building surfaces using artificial neural networks," *Energy and Buildings*, vol. 158, 11 2017.

[2] K. Dutta, R. Lenka, and S. Sarowar, "Improvement of Denoising in Images Using Generic Image Denoising Network (GID Net)," 02 2022.

[3] B. Kumaraswamy, "6 - neural networks for data classification," in *Artificial Intelligence in Data Mining*, D. Binu and B. Rajakumar, Eds. Academic Press, 2021, pp. 109–131. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128206010000112

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017. [Online]. Available: https://arxiv.org/pdf/1706.03762.pdf

[5] J. M. Gómez Hidalgo, E. P. Sanz, F. C. García, and M. D. B. Rodríguez, "Chapter 7 web content filtering," in *Social Networking and The Web*, ser. Advances in Computers. Elsevier, 2009, vol. 76, pp. 257–306. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0065245809010079

[6] M. R. Candes, "The victims of trafficking and violence protection act of 2000: Will it become the thirteenth amendment of the twenty-first century?" *The University of Miami Inter-American Law Review*, vol. 32, no. 3, pp. 571–603, 2001. [Online]. Available: http://www.jstor.org/stable/23317741

[7] D. W. Otter, J. R. Medina, and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2021.

[8] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts: The MIT Press, 1999. [Online]. Available: http://nlp.stanford.edu/fsnlp/

[9] D. Jurafsky, J. Martin, P. Norvig, and S. Russell, *Speech and Language Processing*. Pearson Education, 2014. [Online]. Available: https://books.google.com.ec/books?id=Cq2gBwAAQBAJ

[10] J. Weizenbaum, "Eliza–a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, p. 36–45, jan 1966. [Online]. Available: https://doi.org/10.1145/365153.365168

[11] P. F. Brown, J. Cocke, S. A. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin, "A statistical approach to machine translation," *Comput. Linguist.*, vol. 16, no. 2, p. 79–85, jun 1990.

[12] C. Orăsan and R. Mitkov, "Recent Developments in Natural Language Processing," in *The Oxford Handbook of Computational Linguistics*. Oxford University Press, 06 2022. [Online]. Available: https://doi.org/10.1093/oxfordhb/9780199573691.013.005

[13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'13. Curran Associates Inc., 2013, p. 3111–3119.

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, ser. OSDI'16. USA: USENIX Association, 2016, p. 265–283.

[15] P. Singh and A. Manure, *Natural Language Processing with TensorFlow 2.0*, 01 2020, pp. 107–129.

[16] P. Norvig and S. Russell, *Artificial intelligence: A modern approach*. Prentice Hall, 2010.

[17] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[19] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[20] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, http://www.deeplearningbook.org.

[21] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[22] R. Yamashita, M. Nishio, R. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, 06 2018.

[23] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, 2021.

[24] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011, pp. 342–347.

[25] T. Szandała, *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*. Singapore: Springer Singapore, 2021, pp. 203–224. [Online]. Available: https://doi.org/10.1007/978-981-15-5495-7_11

[26] A. Sherstinsky, "Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167278919305974

[27] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[28] A. Graves, "Generating sequences with recurrent neural networks," *ArXiv*, vol. abs/1308.0850, 2013.

[29] D. Sussillo and O. Barak, "Opening the black box: Low-dimensional dynamics in high-dimensional recurrent neural networks," *Neural computation*, vol. 25, 12 2012.

[30] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990. [Online]. Available: https://www.sciencedirect.com/science/article/pii/036402139090002E

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[32] S. Karita, X. Wang, S. Watanabe, T. Yoshimura, W. Zhang, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. Yalta, and R. Yamamoto, "A comparative study on transformer vs rnn in speech applications," 12 2019, pp. 449–456.

[33] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Scao, S. Gugger, and A. Rush, "Transformers: State-of-the-art natural language processing," 01 2020, pp. 38–45.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *ArXiv*, vol. abs/1810.04805, 2019.

[35] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[36] T. Holt, A. Bossler, and K. Seigfried-Spellar, *Cybercrime and Digital Forensics: An Introduction*. Taylor & Francis, 2015. [Online]. Available: https://books.google.com.ec/books?id=NsWgBgAAQBAJ

[37] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Cybersecurity threats, countermeasures and mitigation techniques on the iot: Future research directions," *Electronics*, vol. 11, no. 20, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/20/3330

[38] S. L. Granizo, A. L. Valdivieso Caraguay, L. I. Barona López, and M. Hernández-Álvarez, "Detection of possible illicit messages using natural language processing and computer vision on twitter and linked websites," *IEEE Access*, vol. 8, pp. 44 534–44 546, 2020.

[39] D. Kazemi, B. Borsari, M. Levine, and B. Dooley, "Systematic review of surveillance by social media platforms for illicit drug use," *Journal of public health (Oxford, England)*, vol. 39, pp. 1–14, 03 2017.

[40] P.-O. Brissaud, J. Francçis, I. Chrisment, T. Cholez, and O. Bettan, "Transparent and service-agnostic monitoring of encrypted web traffic," *IEEE Transactions on Network and Service Management*, vol. 16, no. 3, pp. 842–856, 2019.

[41] W. A. Al-Khater, S. Al-Maadeed, A. A. Ahmed, A. S. Sadiq, and M. K. Khan, "Comprehensive review of cybercrime detection techniques," *IEEE Access*, vol. 8, pp. 137 293–137 311, 2020.

[42] P. Graham, "A plan for spam," Aug 2002. [Online]. Available: http://www.paulgraham.com/spam.html

[43] J. De Angelis and G. Perasso, "Cyberbullying detection through machine learning: Can technology help to prevent internet bullying?" *International Journal of Management and Humanities*, vol. 4, p. 57, 07 2020.

[44] M. Sap, D. Card, S. Gabriel, C. Yejin, and N. Smith, "The risk of racial bias in hate speech detection," 01 2019, pp. 1668–1678.

[45] A. Go, R. Bhayani, and L. Huang, "Twitter sentiment classification using distant supervision," *Processing*, vol. 150, 01 2009.

[46] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), May 2010. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/385_Paper.pdf

[47] E. Kouloumpis, T. Wilson, and J. Moore, "Twitter sentiment analysis: The good the bad and the omg!" *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 5, no. 1, pp. 538–541, Aug. 2021. [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/1418

[48] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland: Dublin City University and Association for Computational Linguistics, Aug. 2014, pp. 69–78. [Online]. Available: https://aclanthology.org/C14-1008

[49] T. Weerasooriya, N. Perera, and S. Liyanage, "A method to extract essential keywords from a tweet using nlp tools," in *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, 2016, pp. 29–34.

[50] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis," *IEEE Access*, vol. 6, pp. 23 253–23 260, 2018.

[51] S. Gharge and M. Chavan, "An integrated approach for malicious tweets detection using nlp," in *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2017, pp. 435–438.

[52] F. Shamrat, S. Chakraborty, M. M. Imran, J. Muna, M. Billah, P. Das, and M. Rahman, "Sentiment analysis on twitter tweets about covid-19 vaccines using nlp and supervised knn classification algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, pp. 463–470, 07 2021.

[53] U. Naseem, I. Razzak, K. Musial, and M. Imran, "Transformer based deep intelligent contextual embedding for twitter sentiment analysis," *Future Generation Computer Systems*, vol. 113, pp. 58–69, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X2030306X

[54] J. Ángel González, L.-F. Hurtado, and F. Pla, "Transformer based contextualization of pre-trained word embeddings for irony detection in twitter," *Information Processing & Management*, vol. 57, no. 4, p. 102262, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457320300200

[55] T. Mutanga, N. Naicker, and O. O, "Hate speech detection in twitter using transformer methods," *International Journal of Advanced Computer Science and Applications*, vol. 11, 01 2020.

[56] P. I. Khan, I. Razzak, A. Dengel, and S. Ahmed, "Performance comparison of transformer-based models on twitter health mention classification," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 3, pp. 1140–1149, 2023.

[57] H. Kumar Sharma, K. Kshitiz, and Shailendra, "Nlp and machine learning techniques for detecting insulting comments on social networking platforms," in *2018 International Conference on Advances in Computing and Communication Engineering (ICACCE)*, 2018, pp. 265–272.

[58] T. Mackey, J. Kalyanam, J. Klugman, E. Kuzmenko, and R. Gupta, "Solution to detect, classify, and report illicit online marketing and sales of controlled substances

via twitter: using machine learning and web forensics to combat digital opioid access," *Journal of medical Internet research*, vol. 20, no. 4, p. e10029, 2018.

[59] M. Hernández-Álvarez and S. L. Granizo, "Detection of human trafficking ads in twitter using natural language processing and image processing," in *Advances in Artificial Intelligence, Software and Systems Engineering*, T. Ahram, Ed. Cham: Springer International Publishing, 2021, pp. 77–83.

[60] M. B. Sarwar, M. K. Hanif, R. Talib, M. Younas, and M. U. Sarwar, "Darkdetect: Darknet traffic detection and categorization using modified convolution-long short-term memory," *IEEE Access*, vol. 9, pp. 113 705–113 713, 2021.

[61] L. Stappen, F. Brunn, and B. W. Schuller, "Cross-lingual zero- and few-shot hate speech detection utilising frozen transformer language models and axel," *ArXiv*, vol. abs/2004.13850, 2020.

[62] M. Mozafari, R. Farahbakhsh, and N. Crespi, "Cross-lingual few-shot hate speech and offensive language detection using meta learning," *IEEE Access*, vol. 10, pp. 14 880–14 896, 2022.

[63] T. Anwar, "Identify hate speech spreaders on twitter using transformer embeddings features and automl classifiers." in *CLEF (Working Notes)*, 2021, pp. 1808–1812.

[64] H. He and Y. Ma, *Imbalanced Learning: Foundations, Algorithms, and Applications*, 1st ed. Wiley-IEEE Press, 2013.

[65] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, Jul 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0197-0

[66] E. Ma, "Nlp augmentation," https://github.com/makcedward/nlpaug, 2019.

[67] D. Jurafsky and J. H. Martin, "Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition," 2009.

[68] C. D. Manning and H. Schütze, "Foundations of statistical natural language processing," in *MIT Press*, 1999.

[69] Y. Wu, M. Schuster, Z. Chen, Q. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, u. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," 09 2016.

[70] S. Rothe, S. Narayan, and A. Severyn, "Leveraging pre-trained checkpoints for sequence generation tasks," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 264–280, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.18

[71] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://aclanthology.org/P16-1162

[72] P. Gage, "A new algorithm for data compression," *C Users J.*, vol. 12, no. 2, p. 23–38, feb 1994.

[73] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 66–75. [Online]. Available: https://aclanthology.org/P18-1007

[74] L. Zhuang, L. Wayne, S. Ya, and Z. Jun, "A robustly optimized BERT pre-training approach with post-training," in *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. Huhhot, China: Chinese Information Processing Society of China, Aug. 2021, pp. 1218–1227. [Online]. Available: https://aclanthology.org/2021.ccl-1.108

[75] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," in *Proceedings of the 58th Annual*

Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, Jul. 2020, pp. 8440–8451. [Online]. Available: https://aclanthology.org/2020.acl-main.747

[76] D. Q. Nguyen, T. Vu, and A. Tuan Nguyen, "BERTweet: A pre-trained language model for English tweets," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 9–14. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.2

[77] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[78] A. Abdaoui, C. Pradel, and G. Sigel, "Load what you need: Smaller versions of mutililingual BERT," in *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*. Online: Association for Computational Linguistics, Nov. 2020, pp. 119–123. [Online]. Available: https://aclanthology.org/2020.sustainlp-1.16

[79] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[80] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, no. 1, pp. 1–37, Jan 2008. [Online]. Available: https://doi.org/10.1007/s10115-007-0114-2

[81] M. Hossin and S. M.N, "A review on evaluation metrics for data classification evaluations," *International Journal of Data Mining & Knowledge Management Process*, vol. 5, pp. 01–11, 03 2015.

[82] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457309000259

[83] C. Van Rijsbergen, *Information Retrieval*. Butterworths, 1979. [Online]. Available: https://books.google.com.ec/books?id=t-pTAAAAMAAJ