uc3m | Universidad Carlos III de Madrid

Bachelor's Degree in Computer Science and Engineering
Academic year 2020-2021

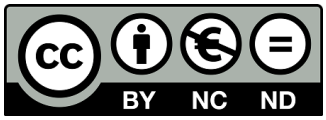
Bachelor's Thesis

"Misogyny Identification in Spanish texts using Deep Learning models"

Ignacio Talavera Cepeda

Advisor/supervisor
Lisardo Prieto González

Madrid, June 21st, 2021



Acknowledgments.

First and foremost, to my family and friends. All that I've accomplished is thanks to them. Specially to my mother, Pilar, Gema, Isabel, Juan and Nena.

To Lucía, a partner in all aspects of life.

To Lisardo, for giving me the oportunity to work at IRSST and being the second in command of this thesis.

To my friends at Recognai.
To Daniel, for sharing his creative vision, his endless ideas and his support in my project. To David, for being the person that have taught me the most about Deep Learning and NLP. To Paco, for answering my dumb questions again and again, and for his support in software development. To Amélie, Leire, Víctor and Javier, for being the heart and soul of the annotation team. Nothing on this thesis could have been achieved without them.

To Dr. Elisabetta Fersini and Dr. Paolo Rosso for giving me the opportunity to use the training and test datasets from IberEval 2018, the cornerstone of this thesis, and for promoting AMI shared tasks.

Abstract

This Bachelor's Thesis aims to obtain Natural Language Processing models capable of detecting, classifying, and alerting misogynistic behaviors in texts written in the Spanish language and to document a process of creation, launch, and subsequent improvement of the existing model by retraining. This work's efforts are focused on improving job safety and reducing precariousness related to misogyny at work, providing a RESTful API that can be easily embedded into other digital environments, and with a classifier capable of detecting these behaviors in any Spanish text.

Keywords:

Deep Learning; Misogynistic behaviours detection; Natural Language Processing; RESTful API; Sentiment Analysis

Spanish Abstract

Este proyecto tiene como objetivo la obtención de modelos de Procesamiento del Lenguaje Natural capaces de detectar, clasificar y alertar de conductas misóginas en textos escritos en castellano, así como de documentar un proceso creación, lanzamiento y posterior mejora del modelo existente con reentrenamientos. Los esfuerzos de este trabajo se enfocan en mejorar la seguridad laboral y reducir la precariedad relacionada con la misoginia en el trabajo, proporcionando una RESTful API, de fácil integración en otros entornos digitales, con un clasificador capaz de detectar esas conductas en cualquier texto escrito en español.

Keywords:

Aprendizaje Profundo; Detección de Misoginia; Procesamiento de Lenguaje Natural; RESTful API; Análisis de Sentimiento

Contents

List of Figures

List of Tables

1

Introduction

This chapter will introduce the project by exploring the problem to be addressed, the motivation to do so, the context that surrounds it, the objectives that are sought and the structure that guided the project.

1.1 Context

Using the jargon of the Technology industry, harassment is considered to be a feature of the many platforms we use to interact online. Every major social network has to deal with this special layer that negatively affects the experience of all of the users, specially the ones to which the harassment is targeted. According to a study performed by the Pew Research Center [1], 41% of Americans have been personally subjected to harassment in online platforms, and 66% has witnessed these behaviours. What is most worrying is that 18% of Americans have suffered severe harassment, such as physical threats, sustained harassment, sexual harassment or stalking.

The study also reveals that women undergo the worst part of this harassment, being more likely to be targeted than men, 11% for women against 5% for men. [1]. Misogynous behaviours can be manifested in numerous ways online, including hostility, discrimination, threats, sexual objectification and social exclusion [2], which are usually transmitted through written content. The growth of social media entails a growth in online misogyny and the ways it can be spread through [3]. Thus, this problem is becoming more relevant by the day, and ways of fighting misogyny online are being developed.

1.2 Problem

Controlling and reducing hate speech, in all its forms, is a problem that many virtual environments and platforms are currently facing, and one that becomes more difficult as the size of those platforms constantly grows. It is also one of the most difficult issues regarding the maintenance of online platforms, as the limits of what is and what is not hate speech are specific to each region and depend on different social, historical and cultural factors.

Almost every platform and application with social features includes a moderation system, to which users can report content that they think is breaking the community guidelines of the platform, and a combination of automatic and manual moderation resources is put into action. Their goal is to be as accurate as possible with the content that should be deleted from the site, or the users that should be punished, while saving human resources. Cases like Facebook or Twitter are extreme ones, since it is far from possible to have a human team manually moderating all the reports in those big platforms, so automatic systems are placed as the vanguard. Automatic agents cannot take the responsibility of the entire workflow, and at the end of the report system of big social media platforms like Twitter there are still human teams deciding which behaviour has to be punished [4]. Thus, the main goal of these systems is to ensure that nobody gets a "false positive" and is banned by mistake, while trying to punish and delete as much hate speech as possible. But, for this very same reason, nowadays it is not difficult to see hate speech on public threads.

The balance is difficult to get. An automatic system is very scalable, and allows sites to grow without having huge human teams in different countries, but at the same time it is not as fair and can be exploited [5]. Hence, while there should always be a human team behind a report system in all virtual platforms, most of the time it is impossible to get all the reports reviewed by actual people. Even if possible, there would be problems related to these teams, like having different ones for each country or language, or protecting them from big quantities of false reports in a Distributed Denial-of-Service (DDoS) attack.

For those reasons, automatic agents designed for hate speech detection should always work by helping a team of humans, which will have the final word and will apply critical thinking. But, the quality of those agents and their accuracy finding possible hate speech content and dismissing "false positives" would keep those required teams as scalable and uniform as possible and allow them to focus their resources on the content that matters.

English, as the most international language used online, has been the focus of the majority of the hate speech detection systems. The development of Natural Language Processing (NLP) and the distribution between languages will be discussed in the next chapter, but the number of agents developed in other languages is significantly lower, even in those with more native speakers, like Chinese or Spanish [6]. The specific characteristics of hate speech are different in each language, and English models, while pow-

erful in their field, have a difficult adaptation to other languages. The development of the agents should consider the target language from beginning to end to ensure their quality and accuracy, so that they can be trusted with such an important task.

1.3 Motivation

The motivation behind this work is to fight hate speech, specially misogyny, in an environment in which it is widely spread. By creating an open-source Spanish misogyny detection model, capable of analyzing written text, the goal is to develop a tool that software developers can easily include in their online platforms to help them in moderation tasks, and to avoid misogynistic hate speech and aggressions, making virtual environments more inclusive and safe. The lack of existing public models in the Spanish language is also one of the main reasons to work on the subject.

This problem has been chosen as part of the collaboration between the Madrid regional goverment and Universidad Carlos III de Madrid, under the cathedra *IRSST-UC3M: I+D+I Para una transformación digital inteligente de la seguridad y salud laboral*[1]. This thesis belongs to the cathedra, which is working towards improving job safety and reducing precariousness related to misogyny at work. Women are one of the most vulnerable collectives in the work environment in Spain: poverty and exclusion affects more women than men, and there is a gap in salary, mental illness with job-related causes and number of aggressions at the workplace [7]–[10].

1.4 Objective

The objective of this Bachelor's Thesis is to obtain NLP models capable of detecting, classifying and alerting misogynistic behaviors in texts written in Spanish, along with the documentation of the process of designing, creating, launching and subsequently improving and maintaining the existing model with retraining. Finally, a RESTful Application Programming Interface (RESTful API) is presented, which could be easily embedded into other digital environments, allowing them to use those classifiers over Spanish written texts.

1.5 Phases of development

The development of this project has been divided into Problem Analysis and State of the Art, Design and comparison of proposed models, Annotation and follow-up training

[1] https://catedrairsst.uc3m.es

for producing the final model and Design and implementation of the RESTful API. This process is shown in Figure 1.1; its non-linearity is due to the nature of the training phases of the proposed model, which have been retrained to include more data to its data corpus and obtain more accurate and reliable results.

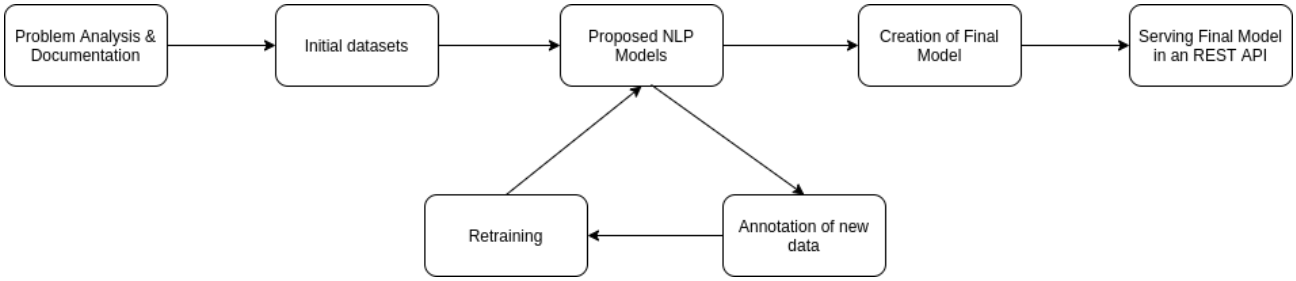


Fig. 1.1. Cycle of Development of this thesis

A documentation task has been carried out during all phases of the project to cover the work done. Firstly, the documentation process will be covered in the State of the Art chapter. After the different aspects related to this thesis has been studied, and the current approaches to Automatic Misogyny Identification (AMI) has been analyzed, the first Deep Learning models will be explained (its design, the datasets that feed them, its data preprocessing, the optimization phase, its final training and the obtained results). Then, a chapter will keep record of the process followed to introduce new instances to those models, and how the retraining is done to improve their results. Ultimately, a Final Model obtained from the retraining phase will be served into a RESTful API, and in Chapter 5 this process and how app developers and users can make calls and obtain predictions will be explained.

1.6 Resources used

Biome.text[2] has been used for the development of the NLP models of this project. It is a practical NLP open source library for Python 3 which allows its users to synthesize the design and training of the models into pipelines. It is based on AllenNLP[3] [11] and Pytorch[4] [12]. Furthermore, HuggingFace[5] [13] provides pre-trained NLP models, which have been used to produce the fine-tuned, final models. This process will be elaborated in following sections.

For the follow-up training and design of the RESTful API, the open source platform

[2] https://www.recogn.ai/biome-text/
[3] https://allennlp.org/
[4] https://pytorch.org/
[5] https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased

Rubrix[6] [14] has been used, a system for storing, labeling, evaluating and deploying models.

NLP models have been built and serve into the RESTful API using Python as the programming language. In addition to all these libraries, several other were used for minor tasks, and will be addressed in their use cases.

1.7 Structure of the document

The structure of this document is as follows:

- An **state of the art** chapter to discuss the current trends on AI, NLP and AMI techniques.
- The **problem analysis**, where the problem to be solved will be explained, alongside the approach to make it.
- A chapter with all **proposed models**, fed with data from IberEval 2018 and IberLEF 2021 conferences.
- A chapter where the **retraining and fine-tuning** process are explained, and the final model of this thesis is obtained.
- A chapter dedicated to how this final model is going to be **served** and accessed.
- The **socio-economic environment** of this thesis.
- The applicable **regulatory framework** of this thesis.
- **Conclusions** and **further work**.

[6] https://github.com/recognai/rubrix

1 Introduction

2

State of the Art

2.1 Artificial Intelligence

Since AI was born, it has taken many different forms, from linear regressions and perceptron-based networks to Deep Learning (a leap that was made mainly on the 2010s). Many different AI techniques are being used in a variety of domains, popularizing this field and attracting a lot of interest, from both researchers and investors. A quick glance at the timeline of events in AI is showed in Figure 2.1, also summarizing the history of this field.

In the State of AI Report 2019 [15], Benaich and Hogarth made several predictions that turned out to be true in 2020:

- NLP companies raise $100 million dollars in one year.
- Universities starts building undergrad AI degrees.
- Google demonstrates a breakthrough in quantum computers, startups around quantum computation start forming.
- Privacy legislation around Machine Learning is adopted by F200 companies outside GAFAM (Google, Apple, Facebook, Amazon and Microsoft).
- No autonomous driving companies drives more than 15 million miles.

Fig. 2.1. Timeline of important events in AI [16].

State of AI 2020 also revealed important facts in **research** and **industry** concerning Artificial Intelligence.

2.1.1 Research

In the AI research field, only 15% of the published papers per year make their code public, and this fact has not improved since 2016. Companies are less likely to publish their code than universities and academic circles, with the examples of OpenAI and DeepMind. Big tech companies usually work with proprietary code [15].

Artificial Intelligence has also approached big computational, economic and environmental costs to gain smaller improvements than before. Studies estimate that dropping the error rate of ImageNet [17] from 11.5% to 1% would require one hundred billion billion dollars. Further breakthroughs in inference capabilities are becoming less and less likely.

But, even as Deep Learning is consuming more data with the flow of time, it is continuing to get more efficient. The computations needed to train neural networks with the same accuracy on ImageNet have been consistently decreasing by a factor of 2 every 16 months.

2.1.2 Industry

Artifical Intelligence is being applied to solve many different problems of the real world. Industries that have increased their use of AI are, mainly, pharmaceutical, medicine, genetics, automobile (specially autonomous driving), scanners and vision, social networks and content-based platforms, in between many others. In Figure 2.2, the business forecast made by Garnet can be seen, dividing the companies by the type of AI which they use. It reveals a continuous growth until, at least, 2025.

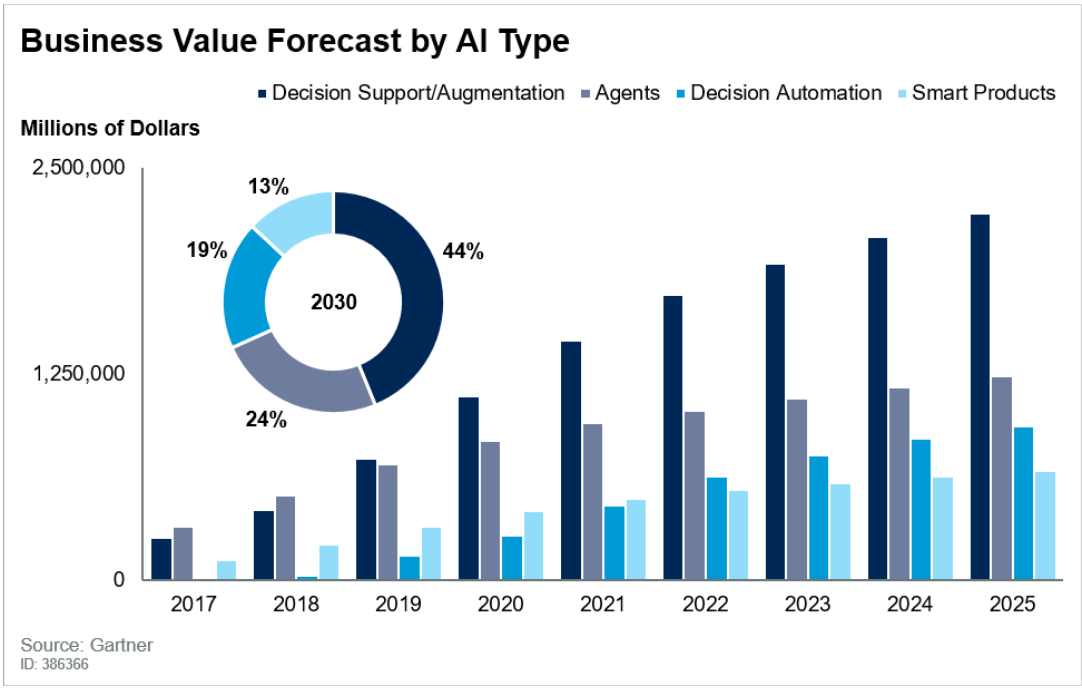


Fig. 2.2. Business value forecast by AI type, made by Gartner [18].

2.2 Neural Networks

One of the main computational models of AI and the most popular nowadays is the Artificial Neural Network, or Neural Networks. They are composed by an interconnected set of artificial neurons capable of transmitting signals through the network [19]. As seen in Figure 2.3, the data input is converted into signals, which travel through the topology of the network until the output layer, which is interpreted as the system's answer to that

input.

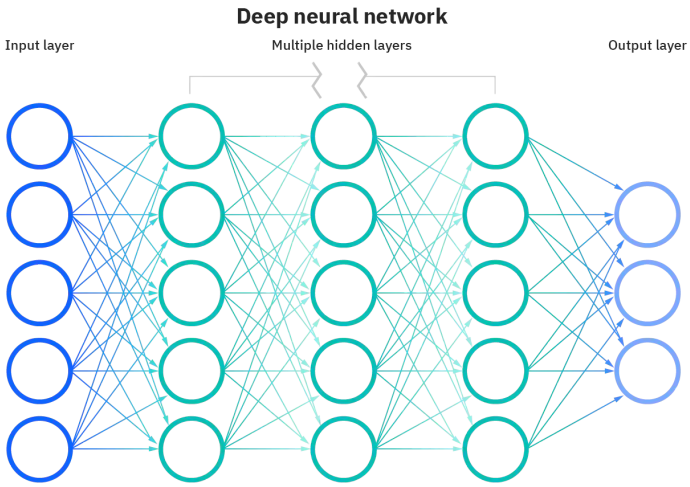


Fig. 2.3. Diagram of a Neural Network [19].

Neurons are organized in layers (the internal layers of the networks are called *hidden layers*), and each of them present several input and output connections, which are weighted. They receive a given input from the previous layer (weighted by the connections), compute a sum function (adding a specific bias for that neuron), and send the result of the activation function applied to that sum to the next layer . There are several popular activation functions, such as *sigmoid*, *tanh* or *ReLu* [20]. An example can be seen in Figure 2.4

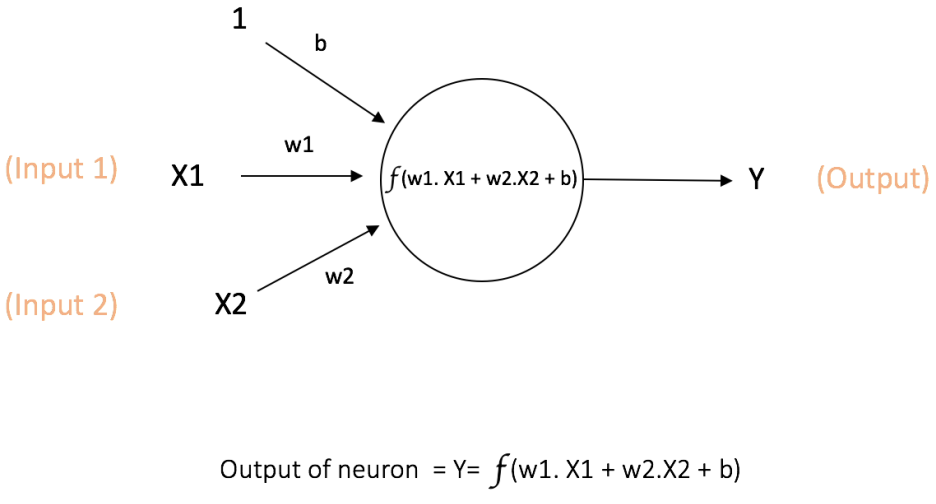


Fig. 2.4. Diagram of Neuron inside a Neural Network [20].

These weights and biases are tuned through a training process. The network receives labelled data, *i.e.* examples of the problem to solve with the result obtained in the real world (also known as *ground truth*), and tries to guess that result. Weights and biases are

changed depending on how close the prediction was from the *ground truth*. After several training cycles, the Neural Networks should obtain good inference results if the problem is compatible and the architecture suits it.

Neural Networks with an important amount of hidden layers, whose architecture is supposed to be more complex than traditional networks, are called Deep Neural Network and studied in the Deep Learning field.

2.3 Deep Learning

Neural Networks were theorized around the middle of the XX century, but it was not until the early 2010s when they surpassed their hardware and data constraints and started solving real-world problems with excellent results [21]. When the computing and data capabilities caught up with the theoretical work around neural networks, researchers started incrementing the complexity of these networks with more layers and new activation functions. These systems started producing better results than rule-based systems, and they opened a window of chance for domains where rules cannot be produced, as their underlying characteristics could not be encoded into a set of rules (because of the complexity of the task).

The development of GPUs also allowed this jump in complexity, as they are more suited to this highly parallelizable type of computation than traditional CPUs. TPUs were also designed to outperform both CPUs and GPUs, building them from scratch with the TensorFlow architecture in mind [22]. A comparison between Tesla V100 GPUs and Google's TPUs can be seen in Figure 2.5

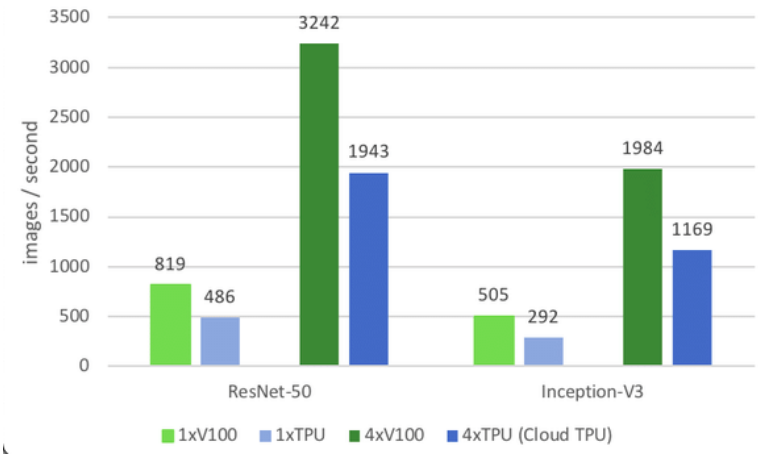


Fig. 2.5. Comparison of the performance of GPUs and TPUs in two popular Neural Networks [23].

2.3.1 Convolutional and Recurrent Neural Networks

The progress made on Deep Learning showed that Neural Networks could solve problems in the fields of Computer Vision and NLP. While NLP is discussed in the following subsections, Computer Vision benefited from the development of special types of Neural Networks: Convolutional and Recurrent Neural Networks. They were also key for the development of NLP [21].

Recurrent Neural Networks were built around the idea that any given output of a network is dependent on the previous one. Therefore, a time step variable could be assigned to each output of a Neural Network. This network topology feeds the past outputs to the network as inputs in each time step (except for the first one), either in the front end of a network or in the form of internal matrix operations as in LSTM [24]. A comparison between regular Neural Networks, Recurrent Neural Networks and LSTMs can be seen in Figure 2.6

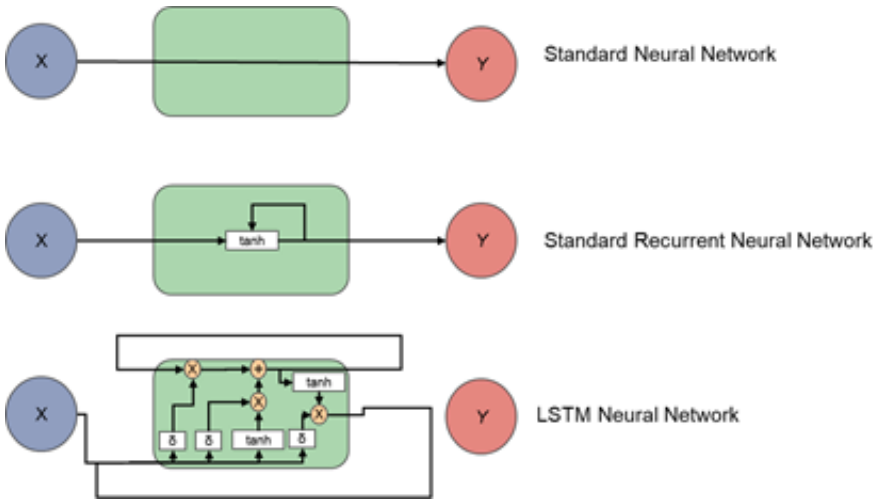


Fig. 2.6. Comparison of the internal operations of regular Neural Networks, recurrent Neural Networks and LSTMs [25].

Alongside the development of Recurrent Neural Networks, Convolutional Neural Network were also developed, allowing the solutions for many problems, specially the ones involving images [26]. The neurons of these networks are designed to imitate the human neurons, whose job is to transmit the visual information, and to do so they exploit the capabilities of the convolution operation. By linking blocks made by *Convolutional layers* (whose job is to perceive the elements of the image) and *Pooling layers* (which reduce the dimensionality of the data), images can be fed into Deep Neural Networks, and classification can be made by adding a regular neural network at the end of the structure, usually referred to as a *classification layer*. A comprehensive diagram of the structure of a Convolutional Neural Network can be seen in Figure 2.7.

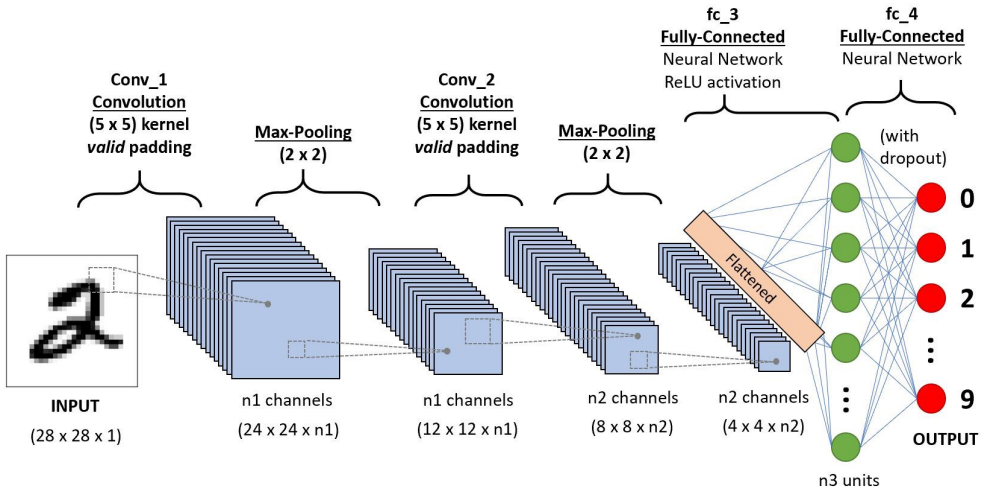


Fig. 2.7. Diagram of the structure of a Convolutional Neural Network [27].

The combination of the time sense that Recurrent Neural Networks gave to AI and the ability to compute images that Convolutional Neural Networks brought made possible that videos could be used as input on Deep Neural Networks.

2.3.2 Training and Libraries

These progresses could not have been possible if there was not a democratizing process in the way that users and researchers train these mathematical models. Back propagation and the Gradient Descent algorithm [28], backbone of the training process of Deep Neural Networks, can be easily implemented using Python libraries like Pytorch [12], TensorFlow [29] and Keras [30]. They also make GPU training accessible. A comparison between the Google search trends of PyTorch and TensorFlow (Keras is usually implemented alongside TensorFlow) can be seen in Figure 2.8, where it can be observed that, while TensorFlow used to gather the majority of the attention, Pytorch has outgrown it in the last years.

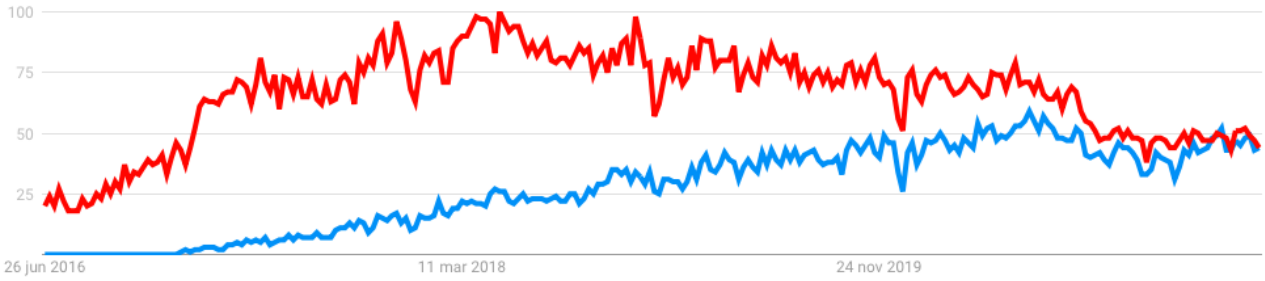


Fig. 2.8. Evolution of search trends of Pytorch (blue) and TensorFlow (red) over the last 5 years.

The preferred language for AI research shifted in the 2010s from R to Python, which has grown to be the most prominent programming language due to its easy learning process and library support [21].

2.4 Deep Learning & NLP

Even though Natural Language Processing (NLP) was conceived long before the popularization of neural networks (the first NLP models to obtain good scores on common NLP tasks were based on N-grams [31]), the current shift towards Deep Neural Networks and big, data-hungry models supposed a new frontier of possibilities and performance on the field.

2.4.1 Word embeddings

Codification of written text is made through pre-trained representations of words based on distribution, which is a process called *word embeddings*. With this technique, capable of encoding words and preserving information about their similarity, and with systems to obtain these encoded vectors such as word2vec [32], context2vec [33], GloVe [34], and ELMo [35] , architectures which learn upon that information of word distribution and closeness emerged. Nowadays, NLP models are able to process word relations (like noun-pronoun), sentence structure and word ambiguity; and new, disruptive results were obtained on tasks like question answering, textual entailment, Named Entity Recognition (NER) and Sentiment Analysis [36].

2.4.2 Big language models

Current trends on larger architectures and quantities of data popularized Transformer models [37], [38], which established themselves as the standard for NLP tasks thanks to their reusability and capability of handling long-range dependencies between words. Therefore, the focus of AI researchers shifted towards creating big, reliable Transformer models which could be sequentially fine-tuned over a specific task, and making them available to the general public. This approach reduces drastically the time and resources needed for researchers to train a state-of-the-art NLP model, as they only need to train the final part of the pipeline.

In Table 2.1, a comparison between the most used large language models can be seen, showing their number of parameters and size of the training dataset. Their development through the years has continuously raised a number of parameters and training dataset sizes, obtaining even larger models with more capabilities. In the following section, we will cover the impact of this trend on sustainability, but it is expected to see two lines

Year	Model	Number of parameters	Training data size
2019	BERT [38]	3.4E+08	16GB
2019	DistilBERT [39]	6.6E+07	16GB
2019	ALBERT [40]	2.2E+08	16GB
2019	RoBERTa [41]	3.55E+08	161GB
2020	MegatronLM [42]	8.3E+09	174GB
2020	BETO [43]	3.4E+08	4GB
2020	GPT-3 [44]	1.75E+11	570GB
2021	Switch-C [45]	1.57E+12	745GB

TABLE 2.1. OVERVIEW OF LATEST LARGE LANGUAGE MODELS [46].

of development over the course of the following years. A first line of development will deal with even larger language models, which will probably surpass the current records of number of parameters and dataset sizes; and a second line of development will aim for lighter models, balancing size with capabilites. These second types of models cannot reach similar performances on the same downstream-tasks used to benchmark larger models, but they offer faster inference time and require smaller computational impact to train [39]. Figure 2.9 offers a comparison between these two types of models until January 2020, which can be seen to diverge from January 2019 (latest GPT-3 and Switch-C are not shown due to their release date).

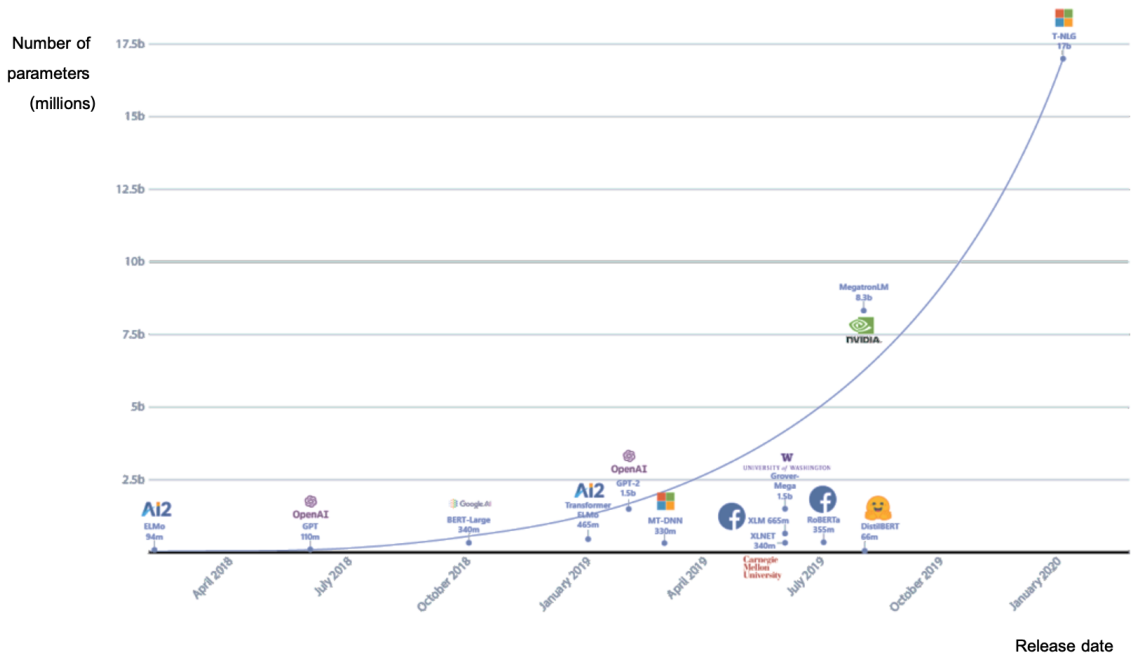


Fig. 2.9. Different language models compared by release date (x axis) and number of parameters, in millions (y axis)[47].

2.5 Pre-trained NLP models in Spanish

Most of the Transformer models available are monolingual, and trained over English corpus. Several projects aimed to reduce the gap between the amount of models designed for the English language, which will probably still be dominant in the coming years, and the rest of languages used around the world. In particular, the Spanish language is the second biggest language by native speakers [6], but finding resources to create Spanish language models is not easy [43].

One of these projects, and the one Transformer model from which the produced models of this thesis have been trained, is BETO [43]. Based on the popular BERT architecture which was initially released with English and Chinese versions, it also includes a multilingual version (usually called mBERT) pre-trained over a corpus in more than 100 languages. Fine-tuning mBERT for specific languages has allowed the creation of state-of-the-art models in many different languages, allowing the democratization of Transformer models over different communities [38]. Some examples of these models cover French [48], Portuguese [49], Italian [50], Dutch [51] and Russian [52].

BETO was designed with a similar architecture than BERT base model, with a total of 110 million parameters and 12 attention layers, 16 attention heads, and an underlying hidden size of 1024. Two versions were trained, with cased and uncased text, over a dataset gathered from Wikipedia, governmental journals, TED Talks, news ... The result was a model that outperformed most of mBERT results, and obtained two new state-of-the-art scores on GLUES benchmark, a special version of GLUE benchmark designed for the Spanish language [43].

Transformer language models improved accessibility to high-performance models, and AI researchers from different locations are adapting these architectures to provide comparable models in the different languages spoken around the world.

2.6 Ethics concerning NLP

In this section, risks associated with the state-of-the-art NLP models are shown and discussed. Environmental, financial and dataset gathering risks will be covered.

2.6.1 Sustainability

The recent progress in both methodology and hardware for training neural networks caused the current state of large networks trained on abundant data. Thanks to it, specially in the field of NLP, notable gains in accuracy and versatility have been reached. However, those improvements came also with an increase on the resources needed to train

those models, both financially and environmentally, which must be taken into account to correctly evaluate the benefits obtained with the state-of-the-art NLP models.

Over many fundamental NLP tasks, the most computationally-hungry models have been reaching the highest scores and the best results [35], [37]. Nowadays, training widely-used models like BERT or GPT-2 require a lot of computational resources which also demand energy, adding financial and environmental costs [53].

Decades ago, many NLP models could be trained using simple computers or servers, but nowadays big models require multiple GPUs or TPUs, therefore restricting the access to these resources over their price. These components, and the hardware needed to organize and control them, add a substantial cost to the environment in the form of energy consumption, which is also increased by the long training times they require. Though some of this energy may come from renewable or carbon-neutral sources, it is estimated that the majority of the energy sources are either gas or coal [53]. To support these issues, a comparison between daily events and NLP training activities has been made in Table 2.2.

Event	Estimated CO_2(kg)
NY to SF flight, 1 passenger.	900
Average of human life.	5000
Car, 1 lifetime.	57153
NLP pipeline.	18
NLP pipeline including tuning and experimentation.	35592
Transformer pipeline.	87
Transformer training with neural architecture search.	284019

TABLE 2.2. ESTIMATED CO2 EMISSIONS OF DIFFERENT EVENTS, COMPARING DAILY HUMAN ACTIVITIES AND NLP TRAINING [53].

Taking into account, for example, the BERT model [38], its training on GPU-based systems is equivalent to a flight crossing the United States, and an increase of the BLEU score of 0.1 % for the English to German machine translation task costs, at least, $150k in compute time and carbon emissions [54].

Thus, a shift into both more renewable energy consumption and more environmentally efficient NLP techniques must be achieved. To raise the awareness necessary, NLP and AI researchers are proposing different changes, such as equitable access to computation resources, prioritization of computationally efficient hardware and algorithms and reports of training time and computational resources needed on proposed models [53]. A combination of some of these practices and an improvement on our renewable energy sources are the only known ways of making NLP models as sustainable as possible.

2.6.2 Dataset Gathering and biases

The amount of data available online has enabled deep learning models to overcome the previous benchmarks on NLP. However, these bigger datasets used to train state-of-the-art models make researchers loose control over which data gets into the training, and therefore the risk that models encode gender, ethnicity, race and disability status biases is higher.

For NLP models, the usual dataset gathering techniques consist of the crawling methodology, usually over sites like Wikipedia, Reddit, Facebook or Twitter, which are user-generated sites, open to anyone, and with structural factors that make them less welcoming to marginalized population [55]. In all these cases, positions of people most likely to express an hegemonic point of view are likely to be passed to the model training, and therefore encoded into the final model [46]. Accepting large amounts of data obtained from the web as a homogeneous representation of all humanity reinforces and increases inequalities.

2.7 Misogyny and Sexual Harassment at workplace

Even though information about reported harassment or the volume of complaints per year is not of public domain, INE made available the statistic of sexual offences and misogynistic behaviours guilty sentences by sex, from 2017 to 2019 [56] (Table 2.3). From this data it is clear that men are the principal aggressors in this type of offences, and that the number of guilty sentences is increasing (around 20%).

Guilty Sentences	2017	2018	2019
Men	331	401	537
Women	1	7	11
Total	332	408	548

TABLE 2.3. GUILTY SENTENCES OF SEXUAL OFFENCES AND MISOGYNISTIC BEHAVIOURS IN SPAIN, BY SEX, FROM 2017 TO 2019.

However, from the same study the information about sexual harassment can be extracted, as shown in Table 2.4. Sexual Harassment sentences are fairly scarce, which is contradictory to the information given by organisms like FRA in Europe or UGT in Spain: 90% of victims of sexual harassment are women, 2484 women denounced these behaviours between 2008 and 2015 in Spain. And, according to *Mujeres en Igualdad*, 65% of victims of sexual harassment do not find the courage to report it to the authorities [57].

Guilty Sentences	2017	2018	2019
Men	10	1	3
Women	0	0	0
Total	10	1	3

TABLE 2.4. GUILTY SENTENCES OF SEXUAL HARASSMENT IN SPAIN, BY SEX, FROM 2017 TO 2019.

When these sexist behaviours happen at the workplace, many women do not find this courage as a consequence of the possible retaliation by their employers, that could even end in a dismissal, according to UGT. In Spain, only 4500 companies are imposed to have a equality plan, and only 276 actually have one [57]. And, if they decide to report it to the authorities, it is very likely that the aggressor will not be found guilty. Proofs, witnesses or psychological report are difficult to get, mostly because of this lack of collaboration by companies. From the 2484 sexist reports, only 49 sentences found guilty the aggressor.

Therefore, the lack of guilty sentences of sexual harassment is not due to its scarcity, but to the hardships women face to make these kind of aggressions public and be protected by the law. Equality ministries of all European Union are working towards protecting women in these processes, inside and outside the companies they work in. To achieve it, a collaboration between governments, private companies, courts and health services is vital, to protect women both physically and psychologically when these behaviours take place.

2.8 Current approach to misogyny detection

The balance between the scalability of a website or application and the human resources allocated to prevent hate speech, in all its forms, and taking into account the social and geopolitical factors of each region where it is used, is difficult to get. Nowadays, most of the online platforms follow the same report system: posts, tweets, messages, photos, videos, etc., can be manually reported by the users, and these contents get into the report system. Pipelines of report systems are private, but they are supposed to be a combination of bots and human work on those reports to flag them and take actions. However, this system implies that the hateful content is visible until it gets reported and the moderation team decides that it is hateful and breaks site's guidelines. Punishment comes after the damage is done to the victim.

Online platforms, and specially content based on written text, give an opportunity for automatic agents to get in the middle of that mechanic, in between the moment when the post is sent and the moment when the victim sees it. It would be impossible to get a human agent to check all posts prior to them made public, but an automatic system could flag dangerous posts, which will be held until a proper evaluation, and let the not

suspicious posts go by. This approach would imply a reliable automatic agent.

To obtain this agent, AI and NLP researchers work on the new AMI. This task consist in distinguishing misogynous contents from non-misogynous ones and categorize the type of misogyny found, and it is usually made over tweets, but new shared tasks are being performed with different targets [58]–[62].

2.8.1 Comparison between AMI shared tasks

These AMI shared tasks are usually defined by the way in which misogyny is classified and detected, as all the submitted models are embracing that design into their architecture to fit better the training data.

In IberEval 2018 [58], one of the most complete classification system is presented:

- **Subtask A**: Misogyny Identification. Discrimination between misogynistic texts from non-misogynistic ones, in the form of a binary classifier.
- **Subtask B**: Misogyny Behaviour and Target Classification. Recognition of the type of misogyny in the text, and its target. Five misogyny categories (*stereotype & objectification*, *dominance*, *derailing*, *sexual harassment & threats of violence* and *discredit*) and two targets (*active*, if the content of the text is aimed towards a certain woman or group of women; and *passive*, if the content is aimed to many potential receivers or women as a gender).

In this shared task, the majority of the participants exploited Support Vector Machines (SVM) and Ensemble of Classifiers (EoC) for both subtasks, while Deep Learning approaches were adopted only by a subset of participants. The winning approaches included SVM (for the second subtask) and Bag of Words (for the first and second subtasks).

SemEval 2019 [61] focused on hate speech, and included two categories: sexism and xenophobia. Both categories had data labeled into three binary categories:

- **Hateful** or **non-hateful** content.
- **Individual** or **generic** target.
- **Aggressive** or **non-aggressive** content.

This time, organizers observed that more than half of the participants investigated Deep Learning models, and supported their work with pre-trained word embeddings and fine-tuning techniques. Fewer participants exploited linguistic features and dependencies, compared to IberEval 2018. SemEval 2019 organizers thought this was probably due

to the high expectations on the ability of Deep Learning models to extract high-level features. However, winners for both Subtask A (hateful or non-hateful content) and Subtask B (target and content classification) used SVM-based architectures.

The next shared task which included AMI, EXIST [59] at IberLEF 2021, took place this year. The results are not available at the time this thesis is being developed, but, as part of the participation procedure, the data gathering and the labeling procedures has already been published. Their approach to AMI takes out target identification and focuses on this two subtasks:

- **Subtask A**: Sexism Identification. Binary classification to discriminate between misogynistic and non-misogynistic texts.
- **Subtask B**: Sexism Categorization. Aims to categorize the message according to the type of sexism between these five categories: *ideological & inequality*, *stereotyping & dominance*, *objectification*, *sexual violence* and *misogyny & non-sexual violence.*

These three shared tasks have their main differences in the choice of categories, dividing in different subsets the domain of misogyny in written text. IberEval 2018 uses different categories than IberLEF 2021, which may overlap in some areas of the domain but offer notable differences in others. For example, IberLEF 2021 differentiates between sexual and non sexual violence in its categories *sexual violence* and *misogyny & non-sexual violence* , while IberEval 2018 includes both in the category *sexual harassment & threats of violence*. One important remark is the fact that organizers in the three of the tasks tend to be the same researchers, which could show that experimentation is being made over the categorization of misogyny in AMI. SemEval 2019 offers three binary classifiers, instead of trying to classify misogyny with multiclass classifier like Subtasks B in IberLEF 2021 and glsibereval. And it is also quite important the fact that both IberEval 2018 and SemEval 2019 offered a way to classify the target of the misogyny aggression, while IberLEF 2021 did not.

2 State of the Art

3

Problem Analysis

The problem to study is the implementation Temis, acronym of *Test Español de MISoginia*, or *Spanish Misogyny Test*. It is an AMI system based on Deep Learning models, which must be capable to detect if there is misogyny in a given input text, which type is it and to whom it is targeted.

This model must be **scalable** and capable of **processing text from different sources** in the Spanish language so it can be implemented in different online environments like chats, social networks, forums, etc. Therefore, this system is expected to be **accessible** (to be implemented in as many systems as possible without the need to change the environment beneath), **agnostic** (so it could be used alongside any framework without the implementation of interfaces, toolboxes or workflows) and **accurate** (to offer the best predictions possible and reduce the workload of moderation teams).

Deep Learning is the chosen technique to develop this system. In the previous AMI shared task discussed in Chapter 2 a shift towards Deep Learning solutions could already be seen. The language models are growing exponentially bigger each year, and Transformer-based solutions are proving themselves to deliver state-of-the-art results while maintaining training cost low, thanks to fine-tuning and retraining.

3.1 System Requeriments

Several models are going to be trained, to offer different solutions to the problem based on the characteristics of the available data. Firstly, a binary classifier will be trained

with the data from IberEval 2018 [58], as it offers the most quantity of instances in the datasets for misogyny in Spanish written language. This classifier will predict, given an input text, if there is misogynistic content or not, without classifying type nor target. This model will be referred as **Binary Classification Model**

Then, a multiclass model will be trained, also with the data from IberEval 2018 [58], and with a multilabel approach. This model will be capable of predicting, given an input text, if there is misogyny in it, what its type is (over the five categories selected by the organizers of the shared task) and to whom is it targeted. More than one category can be predicted for a single input text, but only one target. If any category or target is predicted, the input text is considered to be misogynist. This model will be referred as **Multilabel Classification Model**.

Using a similar architecture as the Multiclass Classification Model, a second multiclass model will be trained, but with the data from IberLEF 2021 [59]. The objective of this model is to validate our Deep Learning approach to the problem with multilabel models, using another sub-domain of data. It will be submitted to the **AMI shared task**, and the obtained results will be discussed. This model will be referred as **Competition Model**.

At last, a final multiclass model will be produced, following a retraining approach. The **Mutlilabel Classification Model** will be retrained, including data from IberLEF 2021, manually annotated to match the same categories. This model should deliver the best results in terms in accuracy and generalization, as it will be trained in two phases, with data from two different sources. It will be referred as **Temis Model** or **Final Model**.

The **Final Model** will be accessible through an RESTful API to allow the inference process to be made using HTTP Requests.

3.2 System Limitations

Even though the objective of the experimentation of this thesis is to push the inference capabilities of the proposed model as far as possible, as stated in the Problem subsection at Chapter 1, it is not intended to be used as a fair judge of misogyny, but as a filter, a tool for moderation teams to automatically filter potentially harmful content. Current NLP and AI capabilities are far from making a system better at finding misogyny than a human being, or equally accurate, consistent and with the same critical thinking than one.

The elicitation of the Deep Learning models has also been determined by the lack of data in the Spanish language. The results in accuracy and inference capability of the models have been highly influenced by the amount of records in the training, validation and test sets available. For this reason, the AMI-related shared tasks and competitions have been crucial, and their open-source datasets are allowing many researchers to expand

this field. It is also for this reason that a retraining and fine-tuning phase is going to be carried out, alongside an annotation work to make new data compatible with the models.

3 Problem Analysis

4

Proposed Models

In this chapter, the design, elicitation and results of the different models designed for the development of this thesis will be covered, alongside the different techniques and resources used for each of the models. It will be divided into three sections, as three models were created, each with a different purpose. The final model, included in the RESTful API, was created from these proposed models, and will be explained in the next chapter of this thesis.

4.1 Binary Classification Model

This first model has the purpose of, given an input sentence, detect if there is an underlying misogynistic behaviour or not. Therefore, it is a binary classification model.

4.1.1 Data Corpus

The data used to feed this model came from the task on AMI at IberEval 2018 [58], in which many models were created with the same objective. Three subtasks were proposed, being the **Subtask A** Binary Misogyny Identification. The organization collected a corpus of 72 million instances written in Spanish language from Twitter, via three different approaches:

- Selecting sets of tweets with representative words such as *z*rra, p*ta*, etc.

- Monitoring of potential victim accounts, such as public feminist women.
- Monitoring potential identified misogynists on the social network.

Some of those collected tweets were labeled using two annotators in an initial phase, a third to resolve disagreements in a second phase, and a voting approach by external contributors from the CrowdFlower[7] platform in a third phase. After this process, the Spanish corpus offered to the contributors for this task had 3307 tweets in its training split and 831 in its test split. In 4.1 the distribution between misogynistic and non-misogynistic instances of the training and test datasets is portrayed.

	Training Split (nr. of instances)	Test Split (nr. of instances)
Misogynistic	1649	415
Non-misogynistic	1658	416

TABLE 4.1. DISTRIBUTION OF MISOGYNISTIC AND NON-MISOGYNISTIC INSTANCES OF THE TRAINING AND TEST SPLITS OF THE IBEREVAL18 SPANISH CORPUS

With this initial distribution of training and test sets, another division was carried out to obtain a validation dataset, needed to perform Hyperparameter Optimization (HPO) processes. The training set was divided into two splits: the new training set, with 85% of the instances of the original, and a validation set, with the remaining 15%. This selection was made preserving the ratio of misogynistic and non-misogynistic instances of the original training split.

4.1.2 Resources used

This system, and all that have been produced for this thesis (including the final model), have been designed and trained using biome.text[8], a practical NLP open source library based on AllenNLP[9] [11] and Pytorch[10] [12]. A Deep Neural Network architecture has been designed for the system, using the Spanish Transformer-based language model *BETO* [43] as the model of choice, which was fine-tuned to the given task. *BETO* is a *BERT* [38] model trained on a Spanish corpus, and distributed via HuggingFace's Model Hub[11] [13]. The environment where the HPO and training took place was a Google Colab session with a 12GB Tesla GPU.

[7] Now called Figure Eight: https://figure-eight.com/
[8] https://www.recogn.ai/biome-text/
[9] https://allennlp.org/
[10] https://pytorch.org/
[11] https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased

4.1.3 Hyperparameter Optimization Process

The Deep Neural Network was fine-tuned for the given task, trained with the training split of the corpus, and evaluated in the HPO with the validation split, using accuracy as the measure of choice. The HPO process consisted of three different sub-processes, with each subsequent HPO fixing some parameters and reducing the search space for others, until the best-performing neural network was obtained. All of them followed a Random Search approach with Bayesian Optimization, which tried to explore as much as possible the search space, and started batching around the best-performing areas of it, balancing exploration and exploitation. They also included ASHA trial schedulers to terminate low-performing trials [63]. HPO processes were developed using the Ray Tune Python Library [64], tightly integrated in *biome.text*.

Parameter	Search Space	Best value
Learning Rate	loguniform(1e-6, 1e-4)	0.00000398
Weight Decay	loguniform(1e-3, 0.1)	0.0707
Batch Size	24	24
Warmup Steps on Learning Rate Scheduler	choice([0, 100])	0
Pooler Type	choice([gru [65], lstm [24]])	lstm
Hidden Size of Pooler's layers	choice([32, 64, 128, 256])	64
Number of Pooler's Layers	fixed to 1	1
Bidirectionality on Pooler's Layers	choice([True, False])	True

TABLE 4.2. LIST OF TUNED HYPERPARAMETERS DURING HPO PROCESS FOR BINARY CLASSIFICATION MODEL

The tuned hyperparameters, its search spaces and the best obtained values (which forms the final trained model) are shown in 4.2, obtained after the third HPO process, and described using Ray Tune search space functions. *Choice* indicates that each run selected a random value from the ones on the list, and *loguniform* samples a value in between the upper and lower bound provided, in a logarithmic scale. In the first HPO process, an initial search over learning rate, batch sizes and weight decay was performed. Once the batch size was fixed to 24 (also taking into account the environment in which the HPO was being made, a Google Colab session with a 12GB Tesla GPU), a second, narrower HPO was performed including warmup steps on the learning rate scheduler [66]. The results of this second HPO process indicated that the models without learning rate scheduler performed better, so it was excluded from the search in the third and final HPO process. How the hyperparameters are distributed and which validation accuracy results from those distributions obtained on the third HPO are shown in Figure 4.1.

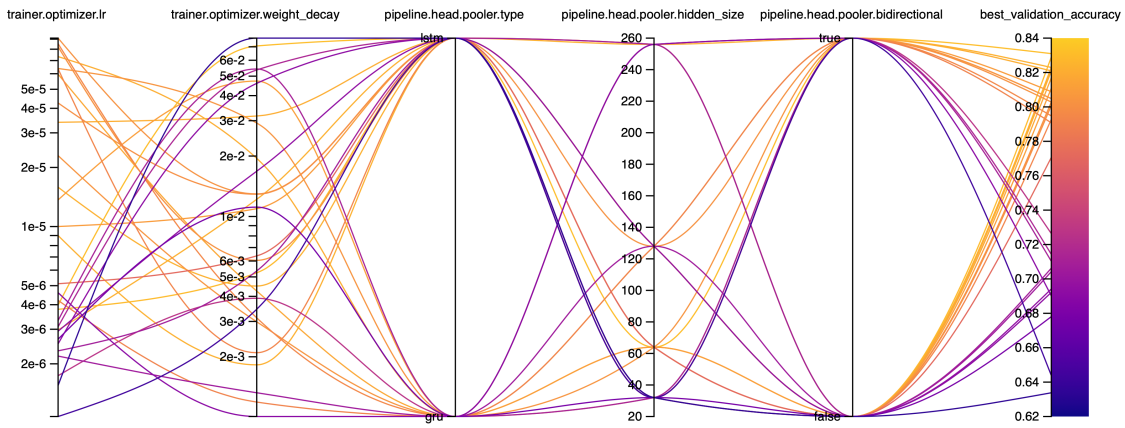


Fig. 4.1. Distribution of hyperparameters during the third HPO of the Binary Classification Model and best validation value obtained.

One important note of this HPO process and the *BETO* language model is that all experiments on this and the following models, including the final neural networks, include fine-tuning to the Transformer language model. This means that not only the final classification layer is trained, but the last layers of the Transformer model are also trained. The first layers of these models are usually kept *frozen*, weights do not change in the training process. The first layers convey the most generic parts, used to process information about the language itself, and so they are trained while the Transformer model is being trained. However, the last layers of the Transformer model are the most specialized ones, and in those places a retraining process can be made, along with the classification layer. This fine tuning adapts the weights of the last layer of the Transformer model, in our case *BETO*, to better fit the given task.

4.1.4 Obtained model

After the HPO process, the final Binary Classification Model was trained with a combination of the training and the validation datasets, and evaluated with the test set. The obtained Deep Neural Network architecture consists on a *BETO* Transformer model as the first element of the pipeline, followed by an LSTM [24] pooler layer and a classification layer. The input sentence is directly processed by the Transformer model, and embedded using the Transformer embedding system. After the Transformer model processes the input, it sends its output (a sequence of vectors) to the LSTM pooler, which converts this sequence of vectors into an standalone vector. Finally, this vector is sent to the classification layer, which computes the final output, in this case a vector with two positions (indicating the probability of misogynistic text and the probability of non-misogynistic text). A representation of this architecture can be seen in Figure 4.2. The *AdamW* algorithm [67] was used for parameter updates.

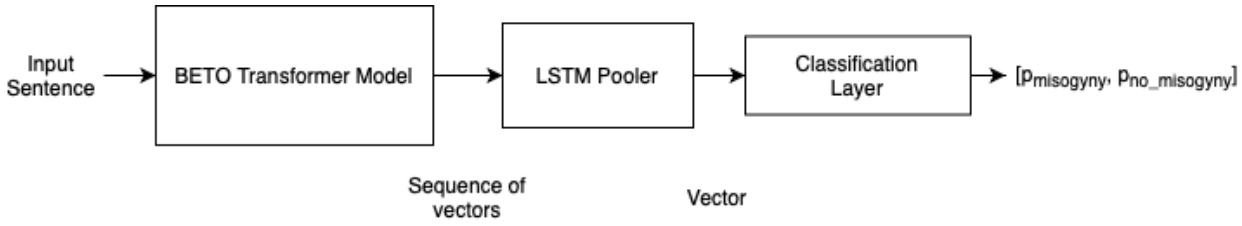


Fig. 4.2. Binary Classification Model Architecture. The output of the pipeline is a tuple with the probability of the text being and not being misogynist.

The test accuracy obtained, after 5 epochs of training, was 0.85559. Even though the comparison with the results of the participants of IberEval 2018 [58] is not fair (the language models has become greater and more precise in the time that has passed), it can be done to illustrate the jump in performance made in three years and the performance of the Binary Classification Model in particular. These results can be seen in Figure 4.3.

	SUBTASK A (Spanish)	
Rank	**Team**	**Accuracy**
1	*14-exlab.c.run3*	0.814681107
2	*JoseSebastian.c.run1*	0.814681107
3	*SB.c.run4*	0.813477738
4	*14-exlab.c.run1*	0.812274368
5	*14-exlab.c.run2*	0.812274368
6	*14-exlab.c.run4*	0.809867629
7	*SB.c.run2*	0.80866426
8	*SB.c.run5*	0.806257521
9	*_vic_.c.run1*	0.805054152
10	*SB.c.run3*	0.805054152
11	*SB.c.run1*	0.803850782
12	*AnotherOne.c.run1*	0.802647413
13	*meybelraul.c.run5*	0.796630566
14	*meybelraul.c.run2*	0.78820698
15	*meybelraul.c.run3*	0.78700361
16	*meybelraul.c.run4*	0.782190132
17	*ixaTeam.c.run1*	0.768953069
18	*AMI-BASELINE*	0.767749699
19	*meybelraul.c.run1*	0.767749699
20	*_vic_.c.run2*	0.76654633
21	*Amrita_CEN.c.run3*	0.74488568
22	*_vic_.c.run3*	0.65944645
23	*Amrita_CEN.c.run1*	0.542719615
24	*14-exlab.c.run5*	0.536702768
25	*Amrita_CEN.c.run2*	0.529482551

Fig. 4.3. Results of IberEval 18 AMI shared task on the first subtask (binary misogyny classification) in the Spanish language category, provided by IberEval organization.

Input Sentence	Prediction
Lo que una elige es ser una p*ta ignorante que es lo que tú eres diciendo semejante burrada. https://t.co/sfqZwqpD9j	Misogynistic
ESTA MUJER ES UNA Z*RRA	Misogynistic
Las 'Supernenas' de Tánger: sin velo y con minifalda. Tres jóvenes marroquíes luchan contra el acoso sexual https://t.co/oBcJ4foDhE	Non-misogynistic
el no merecía nada de esto la p*ta madre, todos merecen ser felices	Non-misogynistic
Odio el acoso callejero, acabo de darme cuenta que desde hace mucho me preocupo más de vestirme para no 'provocar' a esos asquerosos qls que para sentirme cómoda/linda	Non-misogynistic
Afirmo con enojo que su forma está haciendo tonterías y mal acoso en Japón. https://t.co/Y82vnrIUZw	Non-misogynistic

TABLE 4.3. EXAMPLES OF PREDICTIONS MADE BY THE BINARY CLASSIFICATION MODEL OVER SOME INSTANCES OF THE TEST DATASET.

Biome.text also allows to load a pretrained model and make predictions, so some tests were performed using sentences from the test dataset in Table 4.3. Specially sensitive words have been censored, while preserving readability, using the same approach as IberEval 2018 [58], and some emojis have not been transcribed due to codification incompatibilities, even tough the model has taken them into account.

Considering all the HPO processes and the final training of the model, the time needed to complete them is around 18 hours in a Google Colab environment with a 12GB Tesla GPU.

4.2 Multilabel Classification Model

The Multilabel Classification Model has the purpose of, given an input text, predict if there is misogyny in it, what is its type (over the five categories selected by the organizers of the shared task) and to whom is it targeted.

4.2.1 Data Corpus

The data corpus used for the creation of this model also came from IberEval 2018 [58], but this time using its multiclass data from subtasks A and B. Each record has the following information:

- **Text**: input text extracted from Twitter
- **Misogynous**: defines if the tweet is misogynous or not. It is the same field used in the training of the Binary Classification Model.
- **Misogyny Category**: denotes the type of misogynistic behaviour from these five categories if it has been labeled as misogynous in the previous field:
  - **Stereotype and Objectification**: widely held but fixed and oversimplified image or idea of a woman; description or comparison to narrow standards.
  - **Dominance**: superiority assertion of men over women, highlighting a gender inequality.
  - **Derailing**: justifying women abuse by rejecting male responsibility.
  - **Sexual Harassment and Threats of Violence**: describing actions as sexual advances, requesting for sexual favours, harassment of a sexual nature, intents to physically assert power through threats of violence.
  - **Discredit**: to cause people to stop respecting someone or believing in an idea or argument that comes from a woman with no other argument than gender.
- **Target**: denotes the subject of the misogynistic tweet. It can be:
  - **Active**: targeted to a specific woman or group of women.
  - **Passive**: targeted to many potential receivers, even women as a gender.

In Table 4.4 all different categories and targets are illustrated, using an example.

A record is either non-misogynous or it is misogynous and is related to one of the five categories and one of the two targets. Misogyny categories and target are complementary, they are both present when a record is predicted as misogynous.

As this data corpus is the same as the one discussed in the Binary Classification Model, the extraction techniques, number of records and distribution are the same, as we can see in Table 4.1. Furthermore, this table can be extended to Table 4.5, with the distribution between categories and targets.

Text	Misogyny Category or Target
@AgathaBribli Te lo mereces, por usar uñas de G*ARR*	Discredit
Que es una Mujer embarazada de dos nenas y un varon? Un kit de limpieza con instrucciones.	Stereotype & Objectification
@CarmenBarcelo @3gerardpique Una p*lla de 30 cm te falta para estar ocupada y no decirt tonterias	Sexual Harassment & Threats of Violence
@Otb_35 Que te calles so g*arr*.	Dominance
Las personas feministas solo se dedican a expandir mierda sobre la cara de los hombre y despues realmente lo que buscan es igualdad	Derailing
Eso y más mereces por p*rra	Active
En que se parecen las mujeres a los semaforos. -que despues de las 12 nadie las respeta.	Passive

TABLE 4.4. EXAMPLES OF INSTANCES FROM EACH CATEGORY EXTRACTED FROM THE TRAINING DATASET OF IBERLEF 2021.

4.2.2 Resources used

For this model, the Spanish Transformer-based model of choice has also been *BETO*[43], and the design has also being made on *biome.text*. The environment where the HPO and training took place was also a Google Colab session with a 12GB Tesla GPU.

4.2.3 Preprocessing & Multilabel Approach

The data provided by the organization of IberEval 2018 could be approached with three different classifiers: a binary classifier for the misogyny detection, a multiclass classifier for the misogyny category and another binary classifier for the misogyny target. This approach is a very natural one, as each record of the dataset, if misogynous, can only belong to one category and one target.

However, the approach followed to design this model was **multilabel**. Instead of designing different classifiers for each category, only one pipeline was created, which predicts the presence of each possible category independently. This means that more than one category can be predicted at the same time, it is intended to do so. In this kind of models, a threshold is fixed. The model will predict the probability that each label is present in the input text, and all probabilities surpassing that threshold compound the prediction of the model.

According to this approach, a preprocessing phase must be carried out, which was not necessary in the Binary Classification Model. For all instances of the corpus, a transformation must be made to provide the model a list of predicted labels. If the instance is not

Category or Target	Training Split (nr. of instances)	Test Split (nr. of instances)
Discredit	978	287
Sexual Harassment & Threats of Violence	198	51
Derailing	20	6
Stereotype & Objectification	151	17
Dominance	302	54
Active	1455	370
Passive	194	45

TABLE 4.5. DISTRIBUTION OF CATEGORIES AND TARGETS OF THE INSTANCES OF THE TRAINING AND TEST SPLITS OF THE IBEREVAL18 SPANISH CORPUS.

misogynous, an empty list will be provided, which is equivalent to not predicting any category at all (the model will also make the prediction backwards: if no category surpasses the threshold, no category will be predicted and it will be considered non-misogynous). If the instance is misogynous, its category and target will be appended to the label list. We can see some examples in Table 4.6.

Subtask A	Subtask B		Label List
	Category	Target	
Non-misogynous	-	-	[]
Misogynous	Derailing	Active	[derailing, active]
Misogynous	Stereotype	Passive	[stereotype, passive]

TABLE 4.6. EXAMPLE OF PREPROCESSING PROCESS FOR SEVERAL INSTANCES OF THE DATA CORPUS AND THE LABEL LIST OBTAINED FOR THE MULTILABEL CLASSIFICATION MODEL.

This label list is obtained for all records in the data corpus in order to to carry out the training. The threshold was calculated at the end of the training, making a search through the interval between 0.1 and 0.9 (usually 0.5 is the best threshold for most of the multilabel models).

The main problem with this approach comes from the data, which is not prepared for this multilabel training. We can suppose that a text can have more than one misogyny category in real life (i.e. a sentence can objectify and discredit a woman at the same time), but the organizers and annotators of IberEval 2018 did not label taking this into account, and reduced all instances to its more dominant category. Also, the target cannot be active and passive at the same time, the model will have to learn this feature through the training and not through model parameters. For this very reason, later on this thesis, the

retraining process will be introduced, in which the annotation will be made by annotators from Recognai [12], exclusively for this thesis, and with this multilabel approach.

The non-multilabel data corpus will probably make this model only predict one category and one target at a time if the instances are considered to be misogynous, which is the same behaviour expected from independent classifiers. On the other hand it will also start learning to search for characteristics of all categories in the instances; a process that will finish with the further retraining. An equally remarkable perk of this approach is the fact that misogyny identification, target prediction and categorization are trained through the same model, which can also learn high-level dependencies between those different tasks.

4.2.4 Hyperparameter Optimization Process

The Deep Neural Network produced from the original *BETO* was fine-tuned from scratch for this given task, being part of an iterative HPO process which consisted of three different sub-process that trained the neural network with the training split and evaluated its performance with the validation split. Subsequent HPOs fixed some parameters and reduced the search space for others, until the best-performing neural network was obtained. As in the HPO processes of the Binary Classifier Model, this experimentation phase followed a Random Search approach with Bayesian Optimization, which tried to explore as much as possible the search space, and started batching around the best-performing areas of it, balancing exploration and exploitation. ASHA trial scheduler was included [63], and it was developed using the Ray Tune Python Library [64].

For this model, the macro-averaged F-Score [68] was the measure of choice. Regular accuracy should not be used in systems with imbalances in their classes (which is the case), accuracy is not proper measure, as it does not distinguish between the classified instances of the different classes, which may belong to a more or less populated class. The chosen measure is usually used to evaluate the proposed models on AMI shared tasks [58] [59].

The tuned hyperparameters, their search spaces and the best obtained values (which are the ones used on the final model) are shown in Figure 4.7, described with the Ray Tune search space functions terminology. *Choice* indicates that each run selected a random value from the ones on the list, *randint* indicates that a random integer value is selected from the given interval, and *loguniform* samples a value in between the upper and lower bound provided, in a logarithmic scale. In the first HPO, an initial search is proposed, with greater search spaces and also without fixing the batch size. After this first iteration, the batch size is fixed to 16, and all search spaces are reduced. It is also in the second HPO where the learning rate scheduler with warm up steps [66] is introduced. The third HPO process serves as a consolidation of the best values obtained in second HPO, with

[12] https://www.recogn.ai/es/

very close intervals. We can see its results on Figure 4.4.

Parameter	Search Space	Best value
Learning Rate	loguniform(1e-8, 1e-3)	0.00002852
Weight Decay	loguniform(1e-3, 0.1)	0.00421
Batch Size	choice([4,8,16,24,32])	16
Warmup Steps on Learning Rate Scheduler	randint([0, 200])	100
Pooler Type	choice([gru [65], lstm [24]])	lstm
Hidden Size of Pooler's layers	choice([32, 64, 128, 256])	256
Number of Pooler's Layers	choice([1,2,3])	1
Bidirectionality on Pooler's Layers	choice([True, False])	True

TABLE 4.7. LIST OF TUNED HYPERPARAMETERS DURING HPO PROCESS FOR MULTILABEL CLASSIFICATION MODEL

As in the case of the Binary Classification Model HPO, an important note on this process and the *BETO* language model is that all experiments on this and the following models, including the final neural networks, include fine-tuning to the Transformer language model. This means that not only the final classification layer is trained, but the last layers of the Transformer model are also trained.

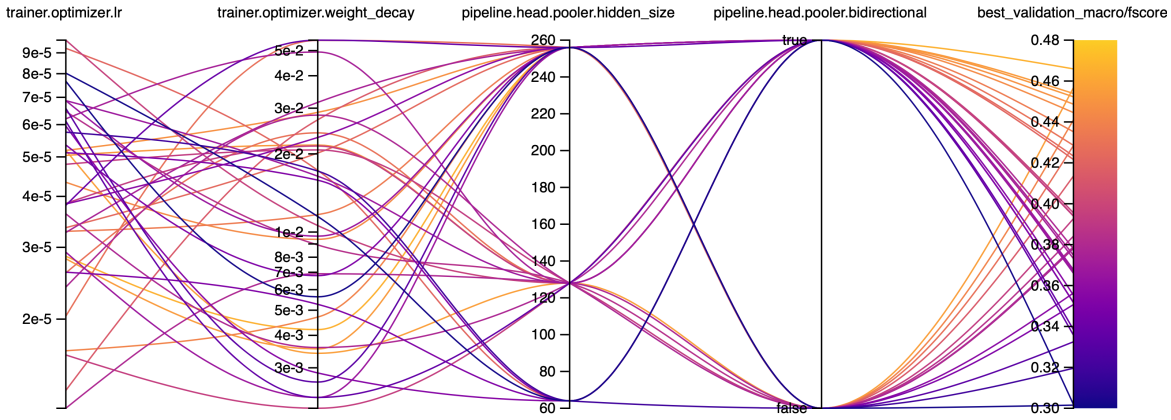


Fig. 4.4. Distribution of hyperparameters during the third HPO of the Multilabel Classification Model and best validation value obtained.

Finally, a threshold search was developed, testing different threshold values from 0.1 to 0.9, and a threshold of 0.5 was chosen to be the best performing. This threshold search was repeated for the training of the final model.

4.2.5 Obtained model

At the end of the HPO process, the parameters of the best-performing model were obtained and the Multilabel Classification Model was trained with a combination of the

training and the validation datasets, and evaluated with the test set. The obtained pipeline is also similar similar to the one obtained for the Binary Classification Model: The *BETO* model, an LSTM [24] pooler and a classification layer.

Given an input text, the model computes the probability of each category being present, and if they surpass 0.5, they are considered predictions. Predictions are, then, lists of variable sizes, and a prediction in the form of an empty list is considered to be a non-misogynous one.

The macro-averaged test F-Score obtained by this model was, after 5 epochs, 0.4412 for the average of both tasks, 0.5496 for the target predictions and 0.3327 for the category predictions. In Figure 4.5, the results for the original IberEval 2018 shared tasks are presented. The winning results are almost reached, since the difference is one of 0.0012 points of macro-averaged F-Score. The winning model was not based Deep Learning, but still one could see that the language models from 2018 to nowadays have seen a upward trend in NLP.

In Table 4.8, some predictions over instances from the training set can be seen. The model seems to be performing well on the different categories, and specially on target detection. The first input sentence could be open to debate, since the discredit is not clear by the lack of context, but the model is detecting correctly that it is an active aggression.

Input Sentence	**Prediction**
@BeeluArzola Hija de re mil puta mereces un Tiro en la concha puta de mierda!!!!!	Misogynistic, Discredit, Active
esta mujer es de las pocas q levantan mi atencion y mi p*lla https://t.co/DZFBowWwjA	Misogynistic, Sexual Harassment, Active
@winterson553 Eso es acoso y provocación	Non-misogynistic
@vanesatrz94 Las mujeres son pendejas aquí y en Francia. Una cosa es el ligue y otra el acoso sexual salu2	Misogynistic, Discredit, Passive
Estoy tratando de enseñarle a mi hermano escribir su nombre, pero es duro duro. Como la hermana. La puta madre.	Non-misogynistic

TABLE 4.8. EXAMPLES OF PREDICTIONS MADE BY THE MULTILABEL CLASSIFICATION MODEL OVER SOME INSTANCES OF THE TEST DATASET.

Adding the HPO processes and the final training of the model, the time needed to carry out this Multilabel Classification Model has been 21 hours in a Google Colab environment with a 12GB Tesla GPU.

English				
Rank	**Team**	**Macro Average F-Measure**	**Macro F-Measure (misogyny_category)**	**Macro F-Measure (target)**
1	SB.u.run3	0.442483	0.292499	0.592467
2	SB.u.run1	0.437201	0.274798	0.599603
3	SB.u.run2	0.431865	0.265948	0.597781
4	SB.c.run5	0.408758	0.222102	0.595414
5	SB.c.run4	0.401897	0.215547	0.588247
6	14-exlab.c.run5	0.369819	0.158329	0.581310
7	resham.c.run1.txt	0.351468	0.148219	0.554718
8	14-exlab.c.run3	0.351380	0.177154	0.525606
9	meybelraul.c.run3	0.349342	0.153617	0.545066
10	14-exlab.c.run4	0.343282	0.180558	0.506006
11	meybelraul.c.run2	0.342323	0.146600	0.538045
12	14-exlab.c.run2	0.341632	0.182421	0.500842
13	_vic_.c.run4	0.339590	0.138319	0.540861
14	_vic_.c.run3	0.339141	0.137421	0.540861
15	14-exlab.c.run1	0.337913	0.175096	0.500730
16	*AMI-BASELINE*	0.337382	0.156794	0.517971
17	_vic_.c.run2	0.336434	0.132007	0.540861
18	meybelraul.c.run1	0.336143	0.159844	0.512442
19	meybelraul.c.run4	0.333332	0.121221	0.545442
20	meybelraul.c.run5	0.328451	0.130986	0.525915
21	JoseSebastian.c.run1	0.326309	0.147691	0.504927
22	ITT.c.run2	0.318026	0.179529	0.456523
23	_vic_.c.run1	0.316368	0.128582	0.504155
24	AnotherTeam.c.run1	0.305317	0.111295	0.499339
25	ITT.c.run1	0.279130	0.155886	0.402374
26	_vic_.c.run5	0.236876	0.160454	0.313297
27	GrCML2016.c.run1.txt	0.178087	0.085939	0.270234
28	GrCML2016.c.run3.txt	0.091724	0.064585	0.118864
29	GrCML2016.c.run2.txt	0.083040	0.052761	0.113318

Fig. 4.5. Results of IberEval 18 AMI shared task on the second subtask (category and target misogyny classification) in the Spanish language category, provided by IberEval organization.

4.3 Competition Model

The Competition Model has the mission of validating our multilabel approach with other data corpus and different labels, and to explore these new datasets that will later be introduced to the final model after the retraining process.

The **EXIST task** at IberLEF 2021 [59] was proposed as an AMI shared task with two subtasks: a binary classification task and a category indentification task. Those proposed categories are different from IberEval 2018, so the **Competition Model** of this section will not make equivalent predictions to the **Multilabel Classification Model**.

This shared task had a multilingual approach: instead of having a category for Spanish models, the global rankings show the obtained metrics by the models over the whole test sets, which were in English and Spanish. For this reason, the participation to this shared task was also multilingual, and three Deep Neural Network were created: one for Spanish language, based on *BETO*[43], one for English language, based on a RoBERTa

architecture tuned over Twitter datasets [69], and a multilingual, also with a RoBERTa architecture, but trained over 198 million multilingual tweets [70]. In this thesis, we will focus on the Spanish results obtained with both the Spanish and the Multilingual Models. **System descriptions for all the submitted models** can be seen in the **standalone publication written** for IberLEF 2021 conference [71].

4.3.1 Data Corpus

The data corpus used for these models was made available by IberLEF 2021 organizers, and divided for both subtasks. Given the multilingual approach, in all records of both the training and the test datasets, their language was specified. Thus, each instance of the data corpus has the following information:

- **Text**: input text extracted from Twitter or Gab[13].
- **Source**: whether the text comes from Twitter or Gab.
- **Language**: whether the text is in Spanish or in English
- **Sexism Identification** (Subtask 1): whether or not the text is considered to be misogynous or not.
- **Sexism Categorization** (Subtask 2): denotes the type of misogynistic behaviour from these five categories, designed by the organization:
  - **Ideological and Inequality**: discredits the feminist movement, rejects existing inequality.
  - **Stereotyping and Dominance**: false ideas about women that suggest they are meant for certain roles or unable to do certain tasks.
  - **Objectification**: the text presents women as object, denying their dignity and personal aspects, or assumes certain physical qualities that women are considered to have to fulfill traditional gender roles.
  - **Sexual Violence**: sexual suggestions, requests for sexual favors or harassment of a sexual nature, rape or sexual assault.
  - **Misogyny and Non-sexual Violence**: content of hatred and violence towards women.

Examples of instances from the training data set are shown on Table 4.9

[13] https://gab.com

Text	Subtask 1	Subtask 2
@rocionahle @GobiernoMX El chiste no es ser mujer u hombre.. sino estar bien preparado para ser un representante digno de México. Ahí nos la sales debiendo.,	Non-sexist	-
"@DianaAS7 Cállese perra porque la mato, pero prefiero eso que estar sudando",	Sexist	Misogyny and Non-Sexual Violence
@Scream_Spain “La rubia tonta de las tetas grandes”,	Sexist	Stereotyping and Dominance

TABLE 4.9. EXAMPLES OF INSTANCES EXTRACTED FROM THE TRAINING DATASET OF EXIST SHARED TASK.

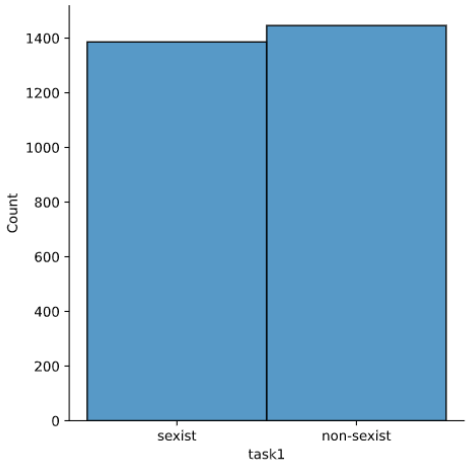


Fig. 4.6. Class distribution of instances of the Spanish train dataset for Subtask 1 of EXIST shared task.

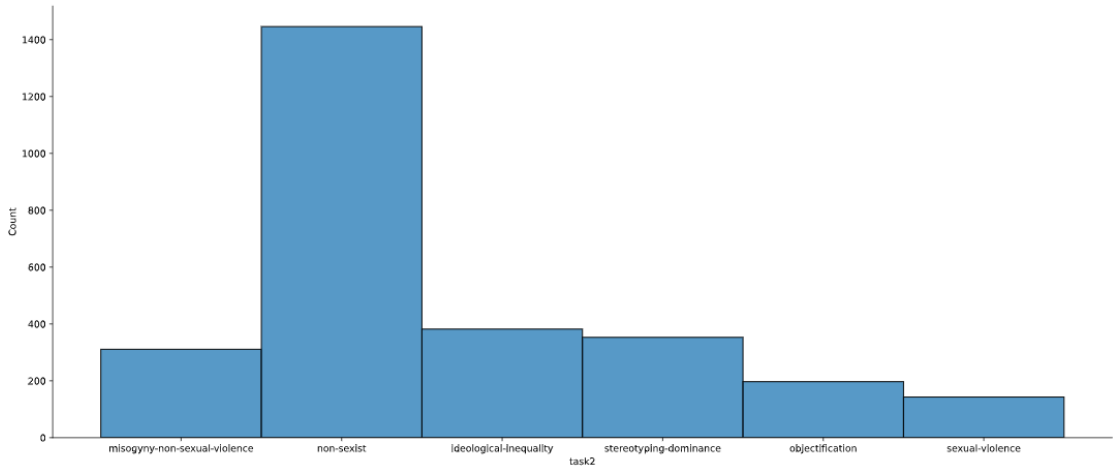


Fig. 4.7. Class distribution of instances of the Spanish train dataset for Subtask 2 of EXIST shared task.

For the data gathering process, the organization collected a number of popular expressions and terms, both in English and Spanish, considered to be misogynous. These terms were analyzed by Trinidad Donoso and Miriam Comet, experts in gender issues, and they were used to created a final set of more than 200 expressions that can be used in sexist contexts. Afterwards, a crawling process was developed to extract more than 800000 instances, and from that corpus random instances were selected, ensuring temporal separation between them. Each instance was then annotated by 5 crowdsourcing annotators under the supervision of Trinidad and Miriam. Final labels were selected according to the majority vote, and tweets with a high rate of discrepancy were manually reviewed. The final EXIST data set consists of 6977 tweets for training and 3386 for testing, with additional 492 "gabs" for English and 490 "gabs" for Spanish. The dataset generation was made ensuring class balance according to Subtask 1 [72].

Focusing on the Spanish part of the data corpus, an initial distribution analysis was carried out to see class distributions only for the Spanish language, dividing the analysis by subtasks. Training class distribution of Subtask 1 can be seen in Figure 4.6, and training class distribution of Subtask 2 in Figure 4.7. The distribution between sexist and non-sexist instances of the dataset is correctly balanced. The distribution of Task 2, focusing only on the sexist labels (non-sexist label of Subtask 2 must have the same frequency than the non-sexist label of Subtask 1), is more imbalanced.

4.3.2 Resources used

For this model, the Spanish Transformer-based language model used has also been *BETO* [43]. For the multilingual model, capable of predicting both in Spanish and in English (the further analysis of this model will only consider Spanish instances), a RoBERTa-based multilingual Transformer model was selected [70].

4.3.3 Preprocessing & Multilabel Approach

As a validation of the previous multilabel approach, these models were also designed as multilabel ones. Two pipelines were created, one only for Spanish language and a multilingual pipeline. Those were submitted to the competition alongside an English model, in a system that used one neural network when the prediction needed to be made in Spanish and the other when the prediction was made over an English tweet. Both systems were capable of making the inference for both subtasks.

Subtask 1	Subtask 2	Label List
Non-sexist	-	[]
Sexist	Objectification	[objectification]
Sexist	Sexual Violence	[sexual_violence]

TABLE 4.10. EXAMPLE OF PREPROCESSING PROCESS FOR SEVERAL INSTANCES OF THE DATA CORPUS AND THE LABEL LIST OBTAINED FOR THE COMPETITION MODEL

The preprocessing phase was similar to the one followed for the **Multilabel Classification Model**, but changing the IberEval 2018 categories for these five: **ideological & inequality**, **stereotyping & dominance**, **objectification**, **sexual violence** and **misogyny & non sexual violence**. Example of the obtained label list, with which the systems are fed, can be seen in Table 4.10.

4.3.4 Hyperparameter Optimization

In order to produce these two models, two independent HPO processes were carried. The multilingual system was fed with the whole training dataset, with instances in both English and Spanish, while the Spanish system was only trained with the Spanish data. Nevertheless, the same distribution of training and validation datasets division was followed: 85% for the training split and 15% for the validation split. The rest of the characteristics of the HPO process from the **Multilabel Classification Model** were the same, with three HPO subprocesses for each subsystem, using Random Search, Bayesian Optimization and the ASHA trial scheduler [63] with the Ray Tune Python Library [64]. The macro-averaged F-Score [68] was the metric chosen by the organization to calculate the results, so it was also used in this thesis for unification purposes. Both systems also included fine-tuning to their Transformer language models and, at the end, they both had a threshold search performed independently.

The final models were trained with both the training and the validation splits using the values obtained in Table 4.11, in 3 epochs for the Spanish neural network, and in 4 epochs for the Multilingual neural network (these values were obtained making a previous training with the training set over the validation dataset, and marking the point of the training where the overfitting began).

Finally, a threshold search was developed, testing different thresholds values from 0.1 to 0.9. The results of these searches are shown in Figures 4.8 and 4.9. For both cases, a threshold value of 0.5 was selected

Parameter	Search Space	Best Value	
		Spanish subsystem	Multilingual subsystem
Learning Rate	loguniform(1e-8, 1e-3)	0.00001734	0.00001511
Weight Decay	loguniform(1e-3, 0.1)	0.004972	0.07449
Batch Size	choice([4,8,16,24,32])	8	16
Warmup Steps on Learning Rate Scheduler	randint([0, 200])	12	14
Pooler Type	choice([gru [65],lstm [24]])	gru	gru
Hidden Size of Pooler's Layers	choice([32, 64, 128, 256])	128	64
Number of Pooler's Layers	choice([1,2,3])	1	1
Bidirectionality on Pooler's Layers	choice([True, False])	True	True

TABLE 4.11. LIST OF TUNED HYPERPARAMETERS DURING HPO PROCESS FOR SPANISH AND MULTILINGUAL SYSTEMS OF THE COMPETITION MODEL

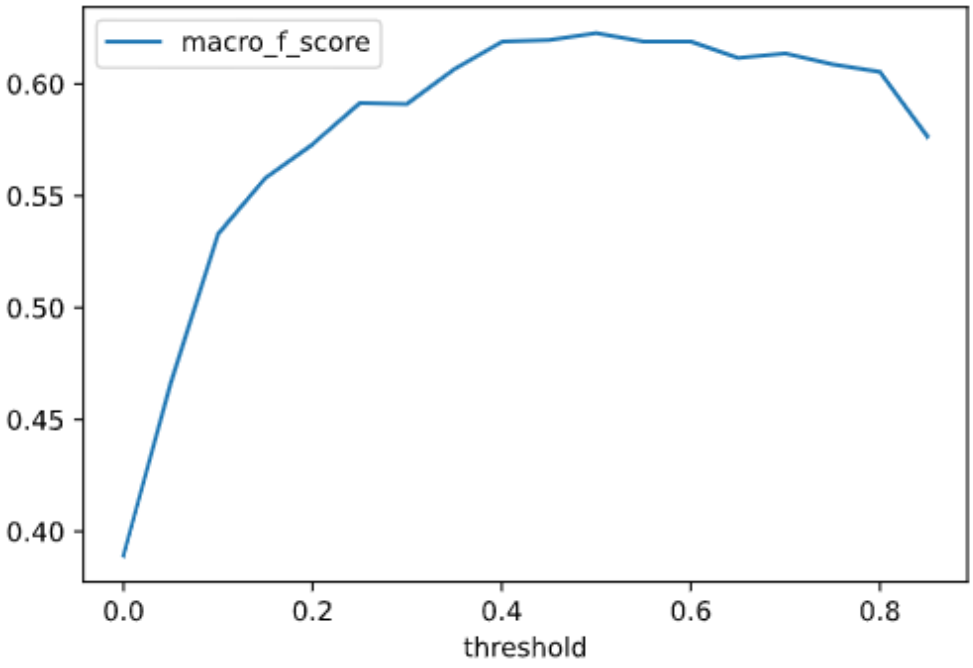


Fig. 4.8. Threshold search for the Spanish system of the Competition Model

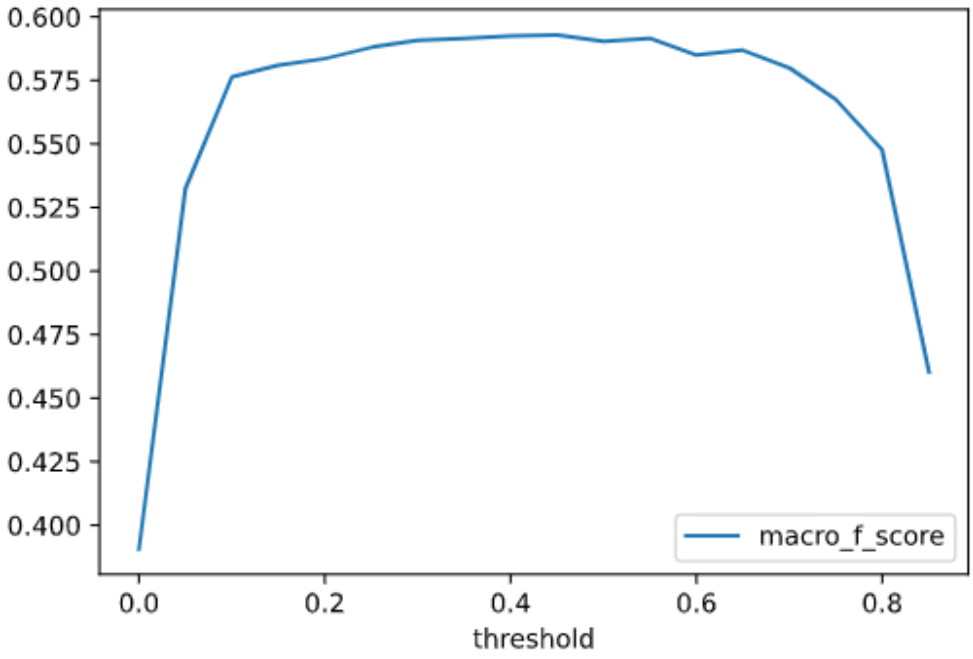


Fig. 4.9. Threshold search for the Multilingual system of the Competition Model

4.3.5 Obtained model & Competition results

When the HPO processes were performed, two pipelines were created. Following the jargon of the EXIST organization, they are called runs. Under the name of the Recognai team[14], the following runs were submitted to the shared task:

- **Run 1**: a system composed of two neural networks, one for the Spanish predictions and the other for the English predictions.
- **Run 2**: a system composed of the Multilingual neural network, which makes the predictions for both Spanish and English instances of the test set.

Given an input text, both runs compute the probability of each category being present and, if they surpass 0.5, they are considered predictions for Subtask 2, and misogyny is predicted for Subtask 1. Even though a multilabel approach was followed, the obtained prediction list was truncated to the most-likely category, because the competition only accepted as predictions one category per instance. All the training, however, was multilabel.

For this competition, a training server was available for us, so the environment included 2 *Tesla V100-PCIE* with 16GB of VRAM. Taking into consideration the Spanish and Multilingual pipelines, the models needed 38 hours to be trained, including all HPO processes.

[14] https://www.recogn.ai/

Position	Run name	Accuracy
1	task1_AI-UPV_1	0.7944
2	task1_AIT_FHSTP_2	0.7917
38	task1_recognai_1	0.7134
52	task1_recognai_2	0.6917

TABLE 4.12. RESULTS OF EXIST SHARED TASK FOR SUBTASK 1, SPANISH RANKING

Position	Run name	Accuracy
1	task2_AI-UPV_1	0.687
2	task2_SINAI_TL_2	0.6815
11	task2_recognai_1	0.6366
24	task2_recognai_2	0.6222

TABLE 4.13. RESULTS OF EXIST SHARED TASK FOR SUBTASK 2, SPANISH RANKING

The models were submitted to the EXIST organization teams on the 6th, April, 2021, and the results were made public on the 14th, April, 2021. In this thesis, we will only discuss Spanish-related results. Tables 4.12 and 4.13 show the results of the Spanish ranking for both subtasks by the two best runs and by the Recognai team (runs are named as *taskX_teamname_runY*). Also, in Table 4.14, a comparison between the results in the validation and test datasets for each model and their number of parameters can be seen.

Models	Task 1 Valid. (accuracy)	Task 2 Valid. (f-measure)	Task 1 Test (accuracy)	Task 2 Test (f-measure)	Model size (nr of params)
Spanish	0.7517	0.6227	0.7134	0.6366	$1.1 \cdot 10^8$
Multilingual	0.7621	0.5903	0.6917	0.6222	$2.8 \cdot 10^8$

TABLE 4.14. COMPETITION RESULTS OBTAINED AND MODEL SIZE, DIVIDED BY RUNS

The best obtained model was the run 1, which was the eleventh classified for Subtask 2 and thirty eighth for Subtask 1. Run 2 underperformed run 1, being the twenty fourth classified for Subtask 2, and the fifty second for Subtask 1. Both results obtained on Task 2 were close to the best ones of the competition, being 0.05 and 0.06 F-Score points away from the winner, respectively. However, the obtained results for Subtask 1 were not that close to the winning position. Binary prediction model would obtain better results on Subtask 1, as they are only trained to predict if there is misogyny or not, but the training process would be significantly more complex, as more Deep Neural Networks would have to be trained.

The mutilingual approach used for run 2 simplified the training process while obtaining good inference results. It did not reach the top performing models of the competition for the second task, but it is established as a valid alternative to classic monolingual training. It also gives researchers the advantage of the multilingual prediction, which could be useful in some domains.

The exploitation of the transfer capabilities of a pretrained language model and its optimized fine tuning to the target domain provides a conceptually easy system architecture and obtains competitive performance, especially for tasks where training data is scarce. With these systems, capable of detecting misogyny with good results over different data and categories, we have concluded that the initial multilabel approach is justified and suitable.

4 Proposed Models

5

Retraining & Final Model

The next objective of this thesis is to perform a retraining phase by labelling new data, including it into the training split and performing a new training of the model, seeking an improvement in its performance. The obtained model will be referred to as **Final Model** or **Temis Model**, and will be the main model served to perform the AMI task. How the model is served will be covered in the next chapter.

5.1 Labelling data from EXIST

At this point, there are two Deep Learning models produced in this thesis, with similar capabilities, but trained over different data and with different categories: the **Multilabel Classification Model** and the **Competition Model**. The labelling process has the purpose of overcoming this incompatibility and allowing both data corpus to be merged, so a model can be retrained. This can be done in two ways:

1. Labelling the datasets of IberLEF 2021 into IberEval 2018's format. This is: labelling the data used to train the **Competition Model** into the same format as the data that feeds the **Multilabel Classification Model**.

2. Labelling the datasets of IberEval 2018 into the IberLEF 2021's format. This would mean to label the data used to train the **Multilabel Classification Model** into the same format as the data feeding the **Competition Model**

5 Retraining & Final Model

Both data formats present meaningful differences, so the choice will influence the behaviour of the **Final Model**.

The first option was chosen, as it presented the advantage that the data from IberEval 2018 predicted not only the misogyny category, but the misogyny target too, a piece of information very important to detect aggressions towards particular women. Therefore, IberLEF 2021's data has been labelled into IberEval 2018's style, and the retraining phase is going to extend the capabilities of the **Multilabel Classification Model**.

An annotation team was arranged, including five volunteers from the Recognai team. After an group study of the labelling techniques of both IberEval 2018 and IberLEF 2021 and a discussion about several examples of each category, provided by the organizers of both shared tasks, annotation guidelines were discussed, to make uniform annotations throughout the process. General aspects of these guidelines were:

- Non-misogynous records are labelled with a non-misogyny category, to distinguish them from discarded records. Those annotations will be changed to an empty label list before feeding the model with them.

- If there is misogyny in a given instance, it can contain several misogyny categories, but only one target. A text cannot be *active* and *passive* at the same time.

- The annotation of more than one category per instance was encouraged, as the annotation team wanted to exploit the multilabel capabilities of this approach. They found that it was common for misogynous text to present a predominant misogyny category, and one or more secondary categories. For example, finding an instance which had mainly *sexual harassment and threats of violence* and also contained *objectification*.

- If there is ambiguity on an instance, or there is non-judgible data (the annotation team found some instances only containing links, or instances in languages like Catalan or Portuguese in the IberLEF 2021 training dataset), the instance is discarded.

- In some rare cases, the annotation team found that none of the five categories fitted the text, but it is considered misogynous. Then, it was only annotated as *active* or *passive*, without a category.

An Inter-Annotator agreement (IAA) system was established to merge the different annotations into one dataset. Inspired by the procedures of IberLEF 2021, and embracing the volunteering condition of the team, corner cases were avoided on purpose, and only records with the majority of the consensus were placed into the final dataset. These were the IAA rules:

1. If there is an instance in which an annotator didn't find sexism, and more than one annotator found sexism, a manual review is made by those annotators that didn't participate in the annotation of that record. If all annotators participated in the initial labelling, or there is still discrepancy in the manual review, the instance is discarded.

2. For a category to be annotated there must be consensus on, at least, two annotators. All categories labelled by two or more annotators are included in the final dataset, and there could be more than one category annotated.

3. For a target to be annotated there must be full consensus, i.e. all annotators that labelled that instance must have found the same target. If there is discrepancy, the record is discarded.

4. For an instance to be annotated as non-sexist there must be consensus between, at least, two annotators.

5. If an annotator discards one record, the record is automatically discarded for all annotators.

The annotation platform of choice was *Rubrix*, developed by Recognai. It is integrated with biome.text and pandas, which were used to produce the proposed models, and it offers an UI to make the annotations. An example of an annotation made in Rubrix can be seen in Figure 5.1

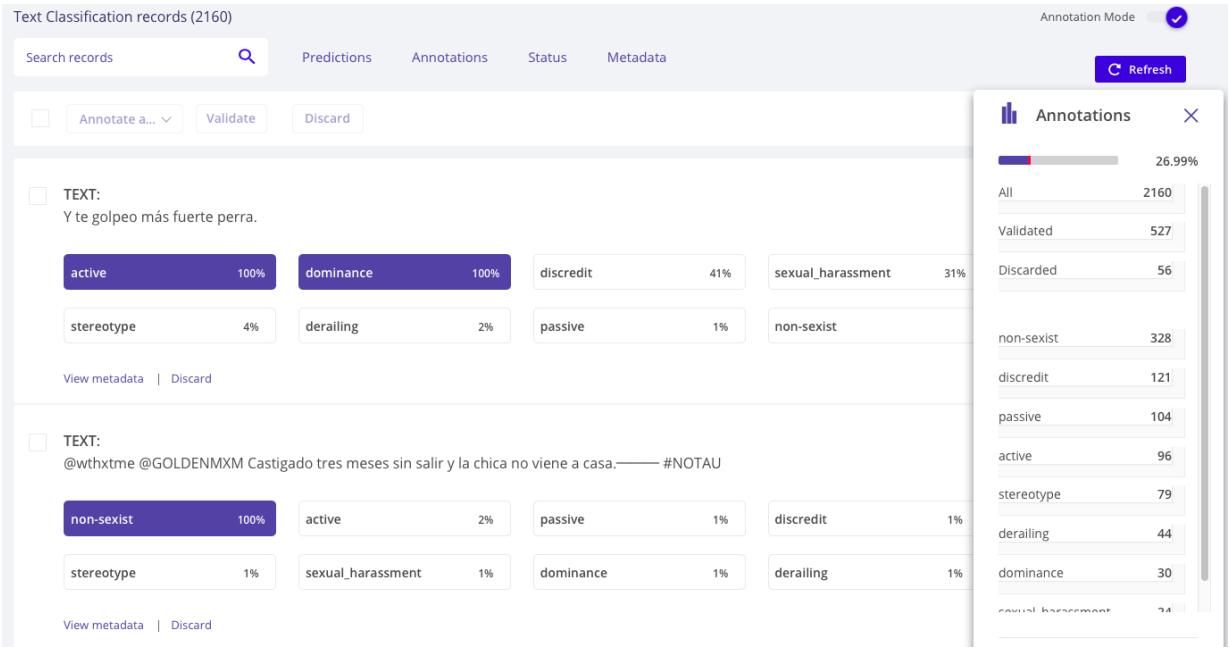


Fig. 5.1. Screenshot of the annotation platform Rubrix during the annotation procedure for the retraining and fine-tuning phase.

The annotation process took place over the week from the 31st of May, 2021, to the 4th of June, 2021. At its end, all the annotations were merged into a final dataset following the inter-annotator agreement rules. 517 instances were finally added to the training dataset, with the following distribution:

- Number of **sexist** instances: 185
- Number of **non-sexist** instances: 332
  - Number of **sexual harassment** instances: 23
  - Number of **dominance** instances: 21
  - Number of **derailing** instances: 35
  - Number of **stereotype** instances: 77
  - Number of **passive** instances: 95
  - Number of **active** instances: 89

An important fact is that there can be more than one category per instance, so the sum of all categories does not match the amount of sexist records. However, the sum of passive and active records does match the amount of sexist records, and the sum of sexist and non-sexist instances also matches the total amount of instances, 517.

5.2 Comparison between retraining and fine-tuning

Once the new labelled instances were added to the data corpus, the next step consisted in training a new model that exploited those new instances. Several approaches could be taken from this point, but they were narrowed into two and compared with each other:

- Adding the new instances to the training dataset and making a full-retraining of the **Multilabel Classification Model**, obtaining a *retrained model*.
- Making a subsequent training to the **Multilabel Classification Model**, obtaining a *fine-tuned model*.

5.2.1 Retraining

The retraining model was performed under the same HPO techniques seen in Chapter 4. Three search processes were performed to explore the search space defined in Table 5.1. As in the elicitation of the previously obtained models, this search followed a Random Search approach with Bayesian Optimization, an ASHA trial scheduler was included

[63], and it was developed using the Ray Tune Python Library [64]. To measure the performance of this model, and to compare it with the **Multilabel Classification Model**, the test split of IberEval 2018 was left unchanged, and another validation split was generated with a 15% of the total instances. Therefore, the HPO process was carried out with a new validation dataset to tune the hyperparameters, but the final evaluation was made over the same test dataset. The same procedure was followed for the fine-tuned model, to stablish a comparison between the three models: the **Multilabel Classification Model**, its *retrained* version and its *fine-tuned* version. The macro-averaged F-Score[68] was used for all of the measures and comparisons.

Parameter	Search Space	Best value
Learning Rate	loguniform(1e-8, 1e-3)	0.00002544
Weight Decay	loguniform(1e-3, 0.1)	0.04739
Batch Size	choice([4,8,16,24,32])	16
Warmup Steps on Learning Rate Scheduler	randint([0, 200])	15
Pooler Type	choice([gru [65], gru [24]])	lstm
Hidden Size of Pooler's layers	choice([32, 64, 128, 256])	256
Number of Pooler's Layers	choice([1,2,3])	1
Bidirectionality on Pooler's Layers	choice([True, False])	False

TABLE 5.1. LIST OF TUNED HYPERPARAMETERS DURING THE RETRAINING PROCESS OF THE FINAL MODEL

5.2.2 Fine-tuning

When using Transformer models as the base for NLP pipelines, all training processes could be considered fine-tuning, as the weights of the Transformer are being slightly changed from the ones used in its training for our subsequent training on our desired task. This fine-tuning phase performed the same process, but over the entire **Multilabel Classification Model**. A dataset was made from the 517 new instances, producing training (85%) and validation (15%) data splits. As this data corpus is considerably smaller than the one used for the initial training, and the **Multilabel Classification Model** already performed well for the given task, the hyperparameters obtained for this model were left unchanged on this fine-tuning except for the learning rate, which was significantly reduced. A set of learning rates from $1 \cdot 10^{-6}$ to $9 \cdot 10^{-8}$ was tested; this experimentation is showed in Figure 5.2. The best learning rate obtained was $7 \cdot 10^{-6}$, with which the fine-tuned model was trained.

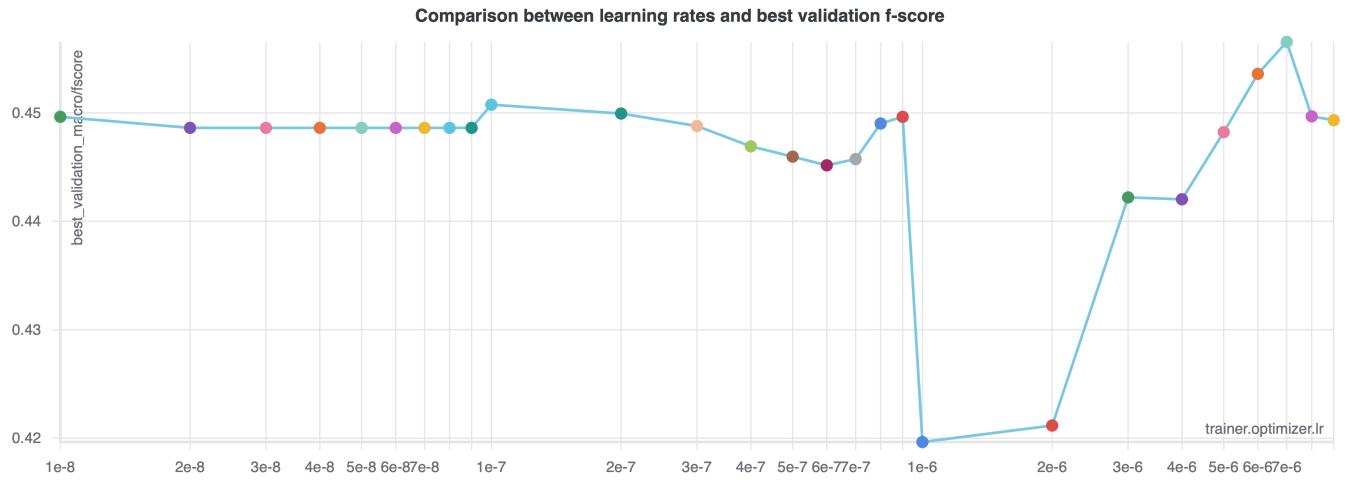


Fig. 5.2. Distribution of learning rates (logarithmic scale, x axis) and obtained validation macro F-Score (y axis) obtained in the fine-tuning process.

5.2.3 Results

In order to compare the performance of these models with the **Multilabel Classification Model** and between them, the three of them were tested with the test split from IberEval 2018. Even though this testing process is not comparing the multilabel capabilities of the new-obtained models, it is an effective way to choose the best-performing model from the three of them. Both the fine-tuned model and the retrained model were optimized with multilabel validation splits, so we are considering that the best multilabel capabilities have already been reached.

The comparison can be seen in Table 5.2. The best-performing model and, thus, the final model, is the *retrained model*.

Model	Average F1-Score
Multilabel Classification Model	0.4412
Retrained Model	0.4689
Fine-tuned Model	0.4467

TABLE 5.2. RESULTS IN AVERAGE F1-SCORE FOR IBEREVAL 2018 SUBTASK 2 AND 3 OF THE OBTAINED MODELS

5.3 Final model

Once the final model was obtained, a threshold search was performed over the test split from IberEval 2018. This result is not binding: the model will offer predictions with a list of labels and the probability of each label, the users will decide which threshold they want to apply. However, after a threshold search with values from 0.1 to 0.9, the best obtained

value was 0.5 (as it is shown on Figure 5.3). This threshold studied has also revealed that the model performs also well with small thresholds, and starts underperfoming around a threshold value of 0.7. It can be seen as a sign of good classifying capabilities: all non-sexist text are assigned with significantly lower values, so thresholds in between 0.1 and 0.6 are able to consider those instances as non-sexist correctly (normal distributions are expected in these kind of threshold searches). Nevertheless, a threshold of **0.5** is recommended for the **Final Model**.

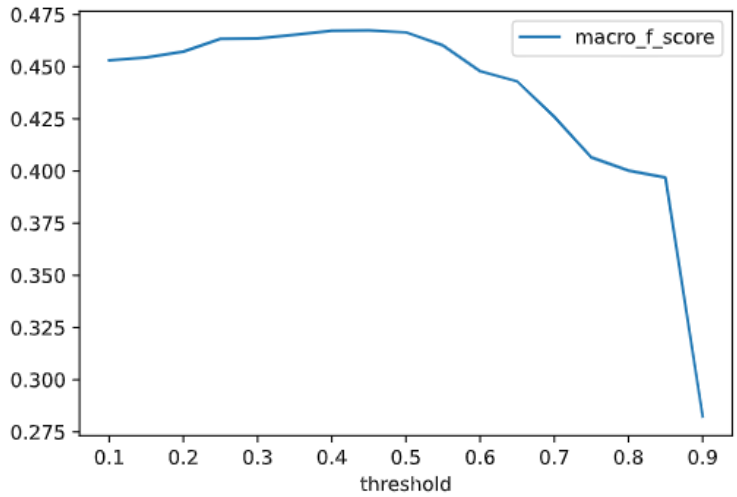


Fig. 5.3. Distribution of the obtained F-1 Score with the Final Model over IberEval 2018 test dataset with different thresholds

This final model has obtained an average F1-score of 0.4689 on the only test set available that is compatible with its format. This means, as seen in Figure 4.5, that it would have been the winner model (this is, obviously, not a fair comparison, a retraining phase has been performed and three years passed, with their subsequent advances on language models, but it is illustrative of the improved performance). Using the recommended threshold of 0.5, a run similar to IberEval 2018's first task was performed: if any category surpassed the threshold, misogyny is considered to be present in the text. Running the model over the test split, an accuracy of 0.8615 was obtained. Results of this same task (Subtask A) were presented on 4.3. Binary misogyny detection has also been improved.

Some predictions of the Final Model made over text from test splits of IberEval 2018 and IberLEF 2021 can be seen in Table 5.3.

5 Retraining & Final Model

Text	Final Model predictions
@mj_vasquezv Jajajaja callate z*rra	Dominance, Active
cabeza, cállate de una p*ta vez y deja de pensar	Non-misogynous
@perraysex Tu eres la p*ta más p*ta con cualquier tanga.... te uso el culo como depósito de semen	Sexual Harrasment & Threats of Violence, Active
@melnicksergio Cállate viejo CTM, escoria de la política!!!!!	Non-misogynous
@OrnePrete Callate que cuando yo me valla vos vas a estat igual z*rra de mierda	Discredit, Active

TABLE 5.3. FINAL MODEL'S PREDICTIONS OVER INSTANCES OF IBEREVAL 2018 AND IBERLEF 2021 FROM THEIR TEST DATASET.

6

Serving the model

This chapter covers the design and creation of the RESTful API that serves the **Final Model**. Firstly, the design process will be covered, explaining which calls were integrated in the API and why. This RESTful API is the interface with which the users will communicate with the NLP model, so it has to be intuitive, easy and convenient to use. The following section will illustrate how the RESTful API was designed, what is included in it, how to access it and make calls and integrate those calls into another environment.

6.1 Design of the API

6.1.1 Value Proposition

The models created in this thesis had the goal of improving work safety and reducing precariousness at work related to misogyny. Prior to the proper API design, the research focused on how this API and the classifier beneath could offer value to a user or a working environment, and in these context a Value Proposition analysis was carried out. It illustrates what the system offers, which of the features provide an added value and which solve preexisting problems. It focuses on three different user profiles, an *app designer* (who is theoretically including Temis in his or her application or website), a *worker* (or a final user, who would be more protected against misogynistic behaviours if Temis was included in the applications or systems that she or he uses on her or his work) and a *common* profile, including both of them. This value proposition can be seen in Figure 6.1, and a description of each of the items in the Value Proposition is shown in Table 6.1.

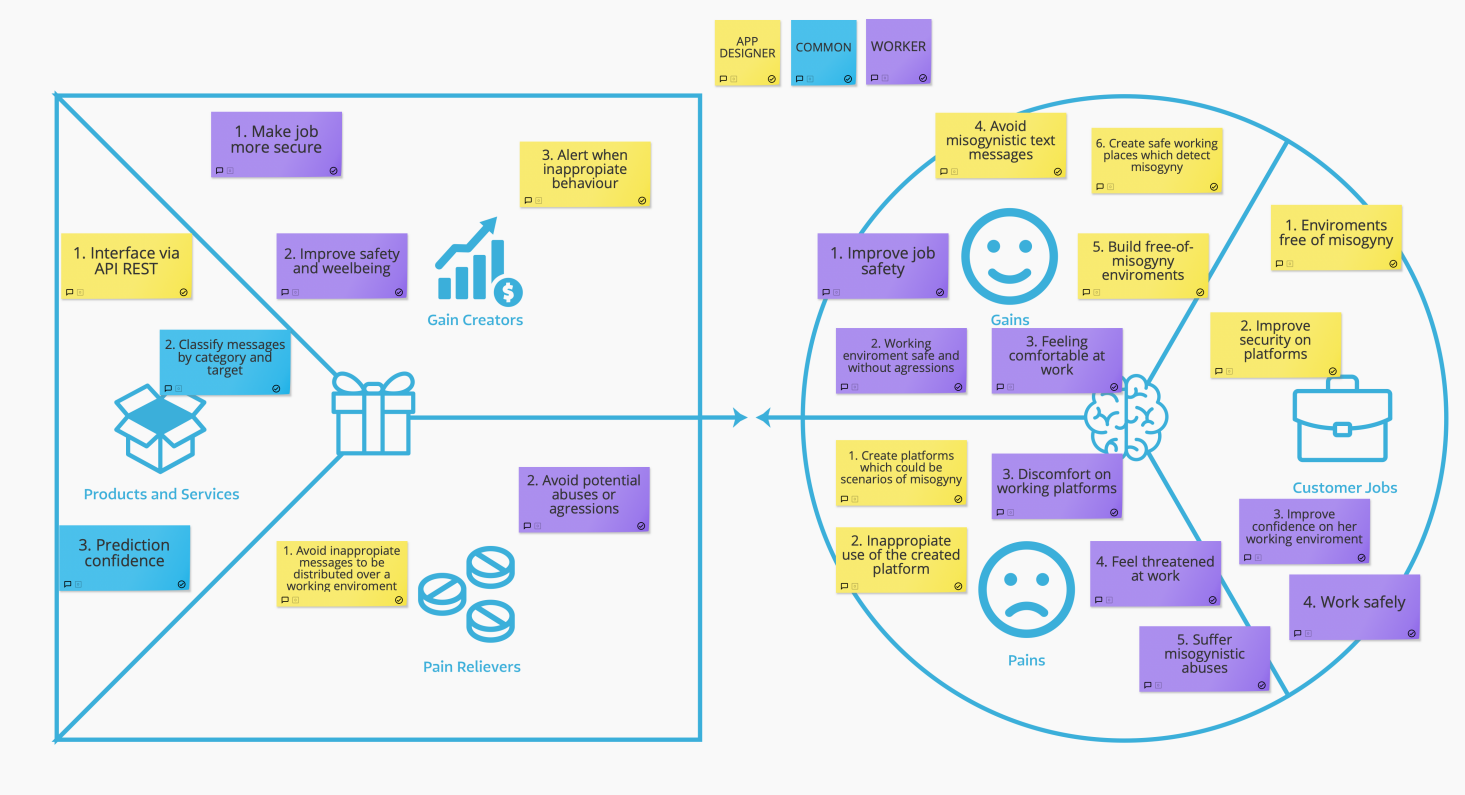


Fig. 6.1. Value Proposition for the Temis system

Some of the main hardships found were related to the few options app designers have to protect their digital environments, and therefore users, specially women, can feel unprotected. The addition of a system like Temis would allow the designers to add security layers, alert systems and prevent misogynous content to be posted without a proper moderation, reducing drastically the number of misogynous content on their platforms.

6.1.2 User Story Map

After the Value Proposition analysis, a User Story Map was designed as the previous step to implement the HTTP Request. In the User Story Map (Figure 6.2), the different ways users can interact with the RESTful API are described. This analysis lead to three main GET requests:

- **Multilabel prediction**: given an input text, receive the label list and the probability of each label (both categories and targets) to be present in the text.
- **Multilabel prediction with threshold**: given an input text and a threshold value, receive the label list whose probability surpasses the threshold. If no threshold is included, 0.5 will be used as default
- **Binary prediction with threshold**: given an input text and a threshold value, classify the text into misogynous or non-misogynous. For a text to be classified as

ID	Target	Description
Products and Services		
1	App Designer	Let the app developers communicate with the model via REST API.
2	Common	Classify texts or messages by misogyny category and target.
3	Common	Classify predictions by confidence.
Gain Creators		
1	Worker	Get a safer job environment.
2	Worker	Improve safety at work and well-being for people.
3	App Designer	Make an alert system to prevent inappropriate behaviour.
Pain Relievers		
1	App Designer	Avoid inappropriate messages to be distributed over the working environment.
2	Worker	Avoid potential abuses or aggressions by preventing the content to go online.
Gains		
1	Worker	Improve safety at job.
2	Worker	Make a working environment safe and without aggressions.
3	Worker	Making people feel comfortable at work.
4	App Designer	Avoid misogynistic text messages to be distributed on the environment.
5	App Designer	Build free-of-misogyny environment.
6	App Designer	Create safe working places which can detect and avoid misogyny.
Pains		
1	App Designer	Create platforms which could be scenarios of misogyny.
2	App Designer	Find inappropriate uses of the created platform.
3	Worker	Discomfort on working platforms.
4	Worker	Feel threatened at work.
5	Worker	Suffer misogynistic abuses.
Customer Jobs		
1	App Designer	Make environments free of misogyny.
2	App Designer	Improve security on environments and platforms.
3	Worker	Improve the confidence on her or his working environment.
4	Worker	Be able to work and communicate safely.

TABLE 6.1. DESCRIPTION OF ITEMS IN THE VALUE PROPOSITION

misogynous, at least one category or target must have a probability greater than the threshold. If no threshold is included, 0.5 will be used as default.

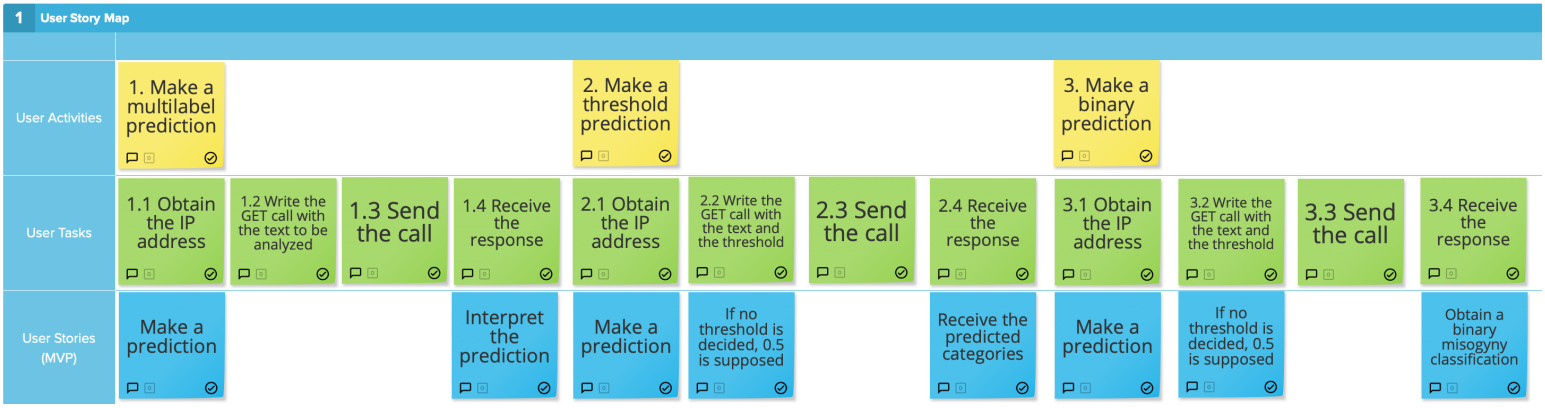


Fig. 6.2. User Story Map for Temis system

6.1.3 GET calls

In this subsection, a brief description of each call will be shown. How to make the GET requests in different frameworks will be described in the next section:

1. **/info**: offers some information about Temis and its creation, in English.
2. **/info_spanish**: offers some information about Temis and its creation, in Spanish.
3. **/label_info**: returns information about the content of the prediction, the categories, the targets and how to interpret them, in English.
4. **/label_info_spanish**: returns information about the content of the prediction, the categories, the targets and how to interpret them, in Spanish.
5. **/predict**: receives as input a parameter named text with the input text to be analyzed. Returns a list with each label and a list with the probability of each label.
6. **/predict_categories**: receives as input a parameter named text with the input text to be analyzed, and a parameter named threshold with the value of the threshold, between 0 and 1. Returns a list with each label that surpasses the given threshold. If no threshold is passed, 0.5 is used as default.
7. **/predict_binary**: receives as input a parameter named text with the input text to be analyzed, and a parameter named threshold with the value of the threshold, between 0 and 1. Returns whether the input text is misogynous or not, according to the threshold given. If no threshold is passed, 0.5 is used as default.

6.1.4 GET responses

The GET requests of Temis can be divided into two types, regarding the type of answer they return:

- **Information calls**, which are */info*, */info_spanish*, */label_info* and */label_info_spanish* returns an string with information about the system
- **Prediction calls**, which are */predict*, */predict_categories* and */predict_binary* returns a JSON-like dictionary. In Figures 6.3, 6.4 and 6.5, the answers of each call are showed.


```
{
  "predictions": {
    "labels": [
      "passive",
      "discredit",
      "derailing",
      "dominance",
      "sexual_harassment",
      "stereotype",
      "active"
    ],
    "probabilities": [
      0.9547768235206604,
      0.7898175716400146,
      0.5529518723487854,
      0.11361064016819,
      0.06852493435144424,
      0.05151188001036644,
      0.024306418374180794
    ]
  },
  "text": "Las mujeres no deberían poder ir a votar"
}
```


Fig. 6.3. Example response from the */predict* call, where the label list is shown, alongside a list with the probability of each label.


```
{
  "predictions": [
    "passive",
    "discredit"
  ],
  "text": "Las mujeres no deberían poder ir a votar",
  "treshold": 0.6
}
```


Fig. 6.4. Example response from the */predict_categories* call, where the list of labels which surpass the given threshold is returned

```
{
  "prediction": "misogynistic",
  "text": "Las mujeres no deberían poder ir a votar",
  "threshold": 0.6
}
```


Fig. 6.5. Example response from the */predict_binary* call, with sexism prediction of the model.

6.2 Accessing the API

The provided domain for the RESTful API is `temis.freemyip.com`. The HTTP Requests explained above can be made to that url to obtain the corresponding answers.

There are many different ways to make HTTP Requests from different environments and programming languages. In this section, the code to make these calls from the Terminal and from Python is explained, but app developers and programmers should search the most suitable method for their environments and frameworks. The following example should illustrate which shape these calls have, not as the only or the preferred way to perform them.

6.2.1 Shell

Curl is a utility command that is present in Linux, MacOS and Windows, and it represents the easiest way to make a HTTP Request. In the following examples, three types of calls will be showed, to show the three different scenarios (no parameters, *text* parameter and *text and threshold* parameter).

```
curl -X GET https://temis.freemyip.com/info
```

```
curl -X GET "https://temis.freemyip.com/predict?text=Input%20text"
```

```
curl -X GET "https://temis.freemyip.com/predict_categories?text=Input%20text&threshold=0.7"
```

Take into consideration that the URL encoding requires the string to be in ASCII character-set, and for that reason spaces are represented as "%20". Note that, depending on how the GET request is made, spaces may be required to be changed to this format.

6.2.2 Python

One of the most popular languages, specially for scripting and Data Science. The requests library from the standard library of Python is needed to make GET requests.

```
# Import the request library
import requests

#Save the API endpoint and the call. In this example, the predict call is used
URL = "https://temis.freemyip.com/predict"

# If the call needs it, create a Python dictionary with the parammeters
PARAMS = {"text": "input text"}

# Send the request and save the response into a variable
r = requests.get(url=URL, params=PARAMS)

# Extract the json response
data = r.json()

# Take into consideration that only prediction calls return a JSON
# Info calls return an string, which can be accessed through r.content
```

6.3 Open-source project

As stated in the motivation of this thesis (see Chapter 1), this project was intended to be, from its beginning, an open source project. Therefore, all the used datasets, the created models, instructions to access the RESTful API and a demo application to test the system can be found at Temis Github repository[15] [73]. The process to run the RESTful API locally is also explained in the documentation of the Github page.

[15] https://github.com/ignacioct/Temis

6 Serving the model

7

Socio-economic Environment

In this section, the costs of elaborating the project of this thesis will be estimated. These costs have been divided in human and technological resources. Because of the special situation produced by COVID-19, all the development has been made online, so no costs derived from transportation or physical spaces will be added.

7.1 Budget

7.1.1 Human Resources

The human resources needed for this thesis will consist of a Computer Science student in the role of a Data Scientist, his tutor and an expert tutor that will help in the development. For the retraining and fine-tuning phase, an annotation team is also needed, with experts in the matter.

The hours needed to carry out this project are estimated by week, taking into account a six-month duration on this project, from January to June. In this period, an average of 15 hours peer week have been calculated for the Data Scientist role, and a one-hour meeting with the tutor. Adding them for the duration of the project, it having lasted 384 hours. 10 hours have been added, as one must meet with the expert tutor, and another 5 hours for meeting with the annotation team, making a total of 399 hours. Around these estimations, the hours of both the tutor and the expert tutor are also estimated.

Resource	Total Hours	Hourly Cost	Total Cost
Data Scientist, Junior	399	20€	7980€
Tutor	50	40€	2000€
Expert tutor	10	40€	400€
Annotator 1	5	15€	75€
Annotator 2	5	15€	75€
Annotator 3	5	15€	75€
Annotator 4	5	15€	75€
Annotator 5	5	15€	75€
Total			10755€

TABLE 7.1. COSTS OF HUMAN RESOURCES FOR THIS PROJECT

The annotation phase took place during one week, with an average of 1.5 hours of annotation time per member. This makes a total of 7.5 hours per member, and 37.5 in total for all members.

An average salary in Spain for a junior Computer Scientist/Data Scientist is around 20€ per hour. On the other hand, an average salary for a senior Computer Scientist around 40€ per hour. Annotation experts are difficult to measure in terms of common jobs whose salary can be looked up, as they can belong to a variety of professions. Taking into account the duration of the job, a salary of 15€ per hour has been decided.

In Table 7.1 all this information is shown, and total costs are calculated. The total of the project would be 10755€

7.1.2 Technological Resources

The technology involved in this thesis can be divided into software and hardware, both of which will be added together to get the final cost of technological resources.

The hardware resources needed have been, mainly, the computer system of the student, which consists of a personal laptop, used for six months and with an expected lifespan of five years.

Most of the software used are open-source, free tools, so they do not add an additional expense. However, a GPU server was used on the final parts of the training (around two months, or 120 hours). From the several options available in the market, Amazon Web Services servers has been chosen. The price of an Amazon Web Services Elastic instance with a 16GB of VRAM located on Ireland ascends to 3.06$ per hour, or 2.53€ per hour. Table 7.2 shows the total cost of the technological resources, 413.6€.

Resource	Cost	Quantity	Duration	Real Cost
Macbook Air 2015	1100€	1	6 months	110€
AWS Elastic Instance	2.53€/hour	1	120 hours	303.60€
Total				413.6€

TABLE 7.2. COSTS OF TECHNOLOGICAL RESOURCES FOR THIS PROJECT

7.1.3 Cost Summary

In Table 7.3 technological and human costs are added up, and a 10% of the total estimated cost is added as risk and contingency reserve to obtain the total cost of the project.

Resource Type	Total Cost
Human Resources	10755€
Technological Resources	413.6€
Contigency Reserve	1116.86€
Total	12285.46€

TABLE 7.3. TOTAL COST FOR THIS PROJECT

7.2 Socio-economic Impact

The socio-economic impact reached if the Temis system is embedded to applications and websites has a lot of potential, as almost every digital environment has text-based interaction. The cost of adding the RESTful API to an already established application is minimum, and the software engineers can build their moderation system around this AMI tool, and therefore reducing its cost and friction.

From the user's perspective, this added layer of protection will help well-being and avoid multiple verbal aggressions that take place on these environments, as it has been showed in this thesis. Not only women, but all users can improve their experience on a website or application free from misogynous content.

Temis is a scalable system, as long as the system in which the RESTful API is upgraded at the same rate as its usage. It is also agnostic, it can be implemented in any framework or programming system as long as it has a HTTP Requests interface (which almost all of them have, they are a core part of online environments). And, lastly, it can be upgraded with new data corpus. The annotation process to adapt almost any related dataset to IberEval 2018 [58] has been described in Chapter 5. So, several *retraining* and *fine-tuning* processes could help the model to be even more accurate

7.2.1 Social Impact

Temis results are focused on making a social impact by reducing misogynistic behaviours in work environments. Therefore, its implementation would suppose a reduction in the spread of misogyny in text, as potentially harmful contents would be flagged and held until the moderation team decides if it is really offensive or sexist.

7.2.2 Economic Impact

Temis is offered without any cost to users or app developers. By doing so, costs on moderation architectures are reduced, and companies or governments can focus its assets on human resources to support this system. Due to its scalability, the potential benefit of its implementation surpasses the budget of the project.

7.2.3 Environmental Impact

As seen in Chapter 2, the environmentally-critical part of this NLP project is the training phase of the Transformer model at the beginning of the pipeline, which, in this case, was made by *Universidad de Chile* [43]. All the models created in Chapter 4 and the subsequent retraining and fine-tuning phases in Chapter 5 have an smaller impact on the carbon footprint of this project. Further HPO processes and final model trainings were made on dedicated servers for AI, with a total of 97 estimated hours of training (a significant less amount of time and resources than the one needed to train BETO).

Even thought the initial costs of building a Transformer-based NLP model are not yet solved, and it is still critical (as it is illustrated in Figure 2.2), all the subsequent training and optimization is significantly less harmful for the environment, even future adaptions of this projects (as BETO only needed to be trained once, when it was released).

8

Regulatory Framework

In this chapter, the regulatory framework that involved this thesis will be analyzed. First of all, the application of Temis into a work environment will be discussed, from the point of view of the occupational hazards, ethics, security and privacy. Lastly, a study of the intellectual property of this model and the tools used to develop it will be carried out.

8.1 Applicable Legislation

The elicitation of the Deep Learning models and the RESTful API did not take into consideration its implementation into a working environment, as it is supposed to be an open-source tool to help companies and governments implement their own misogyny moderation system with the collaboration of an AMI agent. However, in this section the application of Temis into a digital application is assumed.

The most relevant laws concerning this thesis in Spain are the following:

- **Law of data protection**: This regulation is included in "*Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de derechos digitales*". It recognizes the fundamental right of personal data protection for the users of a digital service. Privacy and transparency in data is explained in the **Security & Privacy** subsection.
- **Law of intellectual property**: Regulation included in "*Real Decreto Legislativo 1/1996, de 12 de abril*", which legislates the legal right to own intellectual property

and regulate over its distribution. **Intellectual Property** is covered in its subsection, on this chapter.

8.1.1 Risks

As stated in previous chapters, this AMI tool is not intended to be the only moderation agent involved. Automatic agents cannot take the responsibility of the entire workflow, as their accuracy is not guaranteed for all the possible cases, and they lack critical thinking. At the end of every misogyny abuse detected by Temis there must be a human team capable of making the final decision. However, this and other AMI systems can be used to detect possible misogynistic behaviours, which can be held and not posted or sent until the moderation team ratifies.

Therefore, a serious risk of misuse and possible misogyny behaviours passing through the AMI undetected or non-misogynous text being detected as misogynous is present if the system is used independently, and without the support of a human moderation team. Therefore, the professional responsibility of the companies or governments implementing Temis is to build their moderation systems around human teams, with AMI tools as their support, and not contrariwise.

8.1.2 Ethical Responsibilities

The ethical responsibilities attached to the use of Temis, a system compromised with the fight against gender inequalities, is to use the system to penalize sexist behaviours and to protect the possible victims of those abuses. It is also an ethical responsibility, when building a moderation team to work alongside this or any other AMI system, to ensure diversity in gender and cultural background, to provide the best critical thinking possible related to misogynous content. This will also help the application, website or digital environment to be more protected against misogynous aggressions.

8.1.3 Security & Privacy

The RESTful API of Temis has been built from a Flask application, which is accessible at Temis Github page [73]. Thanks to this transparency in data, it can be assured that the RESTful API is not collecting personal data of any kind, neither from the predicted text nor information about the users. Predictions are made *on the fly*, data is only retained to make the inference process using biome.text.

Biome.text is also a full open-source project. No data is being collected at any point of the inference process, which is based on Pytorch [12].

8.2 Intellectual Property

This project is published under the Attribution, Non-Commercial, No Derivatives 4.0 International Creative Commons license[16]. This means that third parties must credit this thesis, indicate if changes were made and make no endorsement to the licensor or express endorsement by the licensor while doing so. The project cannot be used with commercial purpose. If a transformation or remix is made to this material, it cannot be distributed again.

All the libraries involved in the development of this project have open-sources licenses. These are the specific licenses of the main ones:

- **Biome.text, Rubrix, HuggingFace Transformers and AllenNLP**: Apache License 2.0[17]. A permissive license whose main conditions require preservation of copyright and license notices. Users have the permission to make distributions, modifications and comercial, patent and private uses. There are limitations in trademark use, liability and warranty.
- **Pytorch**: BSD-style license, with minimal restrictions. The main one is the condition that further redistributions of source code must retain the copyright notice, both of source code and binary distributions.
- **Python**: Zero-Clause BSD. Permission to use, copy, distribute and modify the software for any purpose with or without fee.

The produced models, their weights, the datasets gathered during the experimentation, and in-depth documentation, alongside this document, are available at the Github page of the project[18].

[16] https://creativecommons.org/licenses/by-nc-nd/4.0/
[17] https://www.apache.org/licenses/LICENSE-2.0
[18] https://github.com/ignacioct/Temis

8 Regulatory Framework

9

Conclusions

Deep Neural Networks have proven themselves to be one of the most suitable computational models to solve complex, real-world problems, where the classic rule-based systems are not enough to encode all the underlying parameters. NLP researches have obtained outstanding results on many different domains, and AMI shared tasks, with annual editions on many important languages, are the proof that all these tools are being used to make a positive impact on digital environments and on society.

In this section, a summary of the achieved objectives and routes for further work will be discussed.

9.1 Achieved Objectives

In this thesis, the creation of the Temis model has been described, from the problem analysis, to the creation of several Deep Learning models that serve as its foundation, the final retraining and fine-tuning phase, and the serve phase, to make available the inference capabilities obtained for free, open use.

Three foundational Deep Learning models were created using the data corpus from IberEval 2018 and IberLEF 2021 shared tasks, which also offered all their data to allow further research. They were the **Binary Classification Model** (which obtained an outstanding test accuracy of 0.8556 on IberEval 2018 test set), the **Multilabel Classification Model** (with the best results of all IberEval 2018 Deep Learning approaches, and which served as the standard in multilabel classification for the Final Model), and the **Compe-**

tition Model (which obtained a twelfth position on the IberLEF 2021 shared task, and offered the data corpus to perform the retraining phase).

Then, the theorical bases for a retraining and fine-tuning base were established, and 517 new instances were added to the data corpus to feed the **Final Model**. A comparison between fine-tuning and retraining techniques was carried out, and the best-performing of those (retraining) was used to create a **Temis Model** that scored 0.4689 of macro-averaged F-Score on the multilabel test dataset 0.8615 on the accuracy test for binary classification (the best results on both categories).

After the creation of the **Temis Model**, a RESTful API was designed and create to make these inference capabilities available to the app designers and general public, so it can be embedded to moderation systems.

Alongside the work on the main thread of this thesis, a **participation** on **IberLEF 2021** was submitted, scoring 0.7134 accuracy points on the binary subtask (**0.08 points away from the winner**), and 0.6366 macro-averaged F-Score points on the multilabel subtask **(0.05 points away from the winner**).

9.2 Further work

To increase the inference capabilities of posterior Deep Learning-based AMI models, it is crucial to also increase the data available to train them. Shared task are doing an excellent job at uploading new data corpuses, but they are intended to push model architectures beyond, not to create models for the general usage. Therefore, the publication of big, structured databases must be done, ensuring the quality of this ground-truth data by experts in this domain.

With an increment in training data, the search for better Neural Network topologies would be more complete, and better inference would be achieved, thus obtaining better models with better generalization capabilities. More accurate predictions would help moderation teams make a better job and, therefore, obtain safer environments on the Internet.

These better models would help building more advanced moderation pipelines, with which as many potentially sexist or harmful content as possible can be detected before going public, and as many misogynist behaviours and aggression can be denied as possible.

There is an important work to be made towards scope: models that can understand different dialects of a language, texts with different social and cultural characteristics, and, in the case of the Spanish language, models that can understand its different versions used on Latin American countries. Magnifying the types of texts that these types of model can understand and accurately classify would provide a harder layer of protection towards

sexism online.

9 Conclusions

Acronyms

AI Artificial Intelligence. 5, 7–9, 14, 16, 17, 20, 24, 68

AMI Automatic Misogyny Identification. 4, 5, 20, 21, 23, 24, 27, 36, 39, 49, 67, 69, 70, 73, 74, *Glossary:* Automatic Misogyny Identification

BERT Bidirectional Encoder Representations from Transformers. 16, 17, *Glossary:* Bidirectional Encoder Representations from Transformers

BLEU BiLingual Evaluation Understudy. 17, *Glossary:* Bilingual Evaluation Understudy

DDoS Distributed Denial-of-Service. 2, *Glossary:* DDoS attack

EoC Ensemble of Classifiers. 20, *Glossary:* Ensemble of Classifiers

EXIST sEXism Identification in Social neTworks. 21, 42, 45, 46, *Glossary:* Sexism Identification in Social Networks

FRA European Union Agency for Fundamental Rights. 18

GAFAM Google, Apple, Facebook, Amazon and Microsoft. 7

GPT-2 Generative Pre-trained Transformer 2. 17, *Glossary:* Generative Pre -trained Transformer 2

GPT-3 Generative Pre-trained Transformer 3. 15, *Glossary:* Generative Pre-trained Transformer 3

GPU Graphics Processing Unit. 11, 13, 17, 32, 38, *Glossary:* Graphics Processing Unit

HPO HyperParameter Optimization. 28–30, 32, 34, 36–38, 43, 45, 52, 53, 68, *Glossary:* Hyperparameter Optimization

IAA Inter-Annotator Agreement. 50, *Glossary:* Inter-Annotator Agreement

INE Spanish National Institute of Statistics (Instituto Nacional de Estadística). 18

LSTM Long-Short Term Memory. 12, 30, 38, *Glossary:* Long-Short Term Memory

NER Named Entity Recognition. 14, *Glossary:* Named Entity Recognition

NLP Natural Language Processing. 2–5, 7, 12, 14, 16–18, 20, 24, 28, 38, 57, 68, 73, *Glossary:* Natural Language Processing

RESTful API RESTful Application Programming Interface. 3–5, 24, 27, 57, 58, 62, 63, 67, 69, 70, 74, *Glossary:* RESTful Application Programming Interface

SVM Support Vector Machines. 20, 21, *Glossary:* Support Vector Machines

Temis Test Español de MISoginia. 23, 24, 49, 57, 58, 60, 61, 63, 67–70, 73, *Glossary:* Temis

TPU Tensor Processing Unit. 11, 17, *Glossary:* Tensor Processing Unit

UGT Union General de Trabajadores (Spanish Syndicate). 18, 19

UI User Interface. 51, *Glossary:* User Interface

VRAM Virtual RAM. 45, 66

Glossary

annotator People which classify data into different categories to create a labelled corpus that could feed an AI model. 28, 51

Automatic Misogyny Identification Technique which consists on designing automatic agents capable of distinguishing between misogynous and non-misogynous content without human supervision. 4

Bag of Words NLP technique which consists on representing texts by a set of unordered words and basing the decisions on multiplicity. 20

BETO BERT-based Spanish language model. 16, 30, 68

Bidirectional Encoder Representations from Transformers Architecture of NLP models which consist on applying a bidirectional training to Transformer models. 16

Bilingual Evaluation Understudy Benchmark to evaluate machine translations from one natural language to another. 17

biome.text Practical NLP open source library created by Recognai and based in Pytorch, AllenNLP and HuggingFace Transformers. 4, 28, 29, 32, 34, 51, 70, 71

corpus Also known as data corpus, a collection of examples used in AI. 4, 16, 27–29, 33–36, 39, 40, 42, 52, 53, 67, 73, 74

crawling Also known as web-crawling, technique to obtain datasets by extracting information of different websites, following a certain criteria. Agents that automatically perform this task are called crawlers.. 18, 42

DDoS attack A cyber-attack in which the attacker seeks to make an online service unavailable, usually by flooding the target with traffic from a distributed network of sources. 2

Deep Learning Field of Artificial Intelligence which uses multi-label neural networks to model high-level abstractions with non-linear transformations. It is a Machine Learning method. 4, 7, 9, 11, 12, 20, 21, 23, 24, 38, 69, 73, 74

Deep Neural Network Multi-label neural networks used in Deep Learning. 11–14, 28–30, 36, 39, 73

Ensemble of Classifiers Machine Learning techinque which consists on producing different classifiers and making predictions by averaging their results. 20

epoch An entire learning process iteration. In an epoch a neural network goes through all the training data and evaluates the results. 31, 38

F-Score Also known as F-Measure or F_1 score is the harmonic mean of the precision and the recall of an experiment. Very used in classification experiments, specially for non-binary classification (multiclass). If it is macro-averaged, the metric will be computed independently for each class, and the averaged, whereas if it is micro-averaged, the contribution of all classes will be aggregated before computing the average metric. Macro-average treats all classes equally, micro-average is more suitable for class imbalances.. X, 36, 38, 43, 46, 53, 54, 74

Generative Pre -trained Transformer 2 Generative NLP model developed by OpenAI in 2019. 17

Generative Pre-trained Transformer 3 Generative NLP model developed by OpenAI in 2020. 15

GET request An HTTP method designed to retrieve data from a server. A query string is sent, consisting on tuples (name, value). 58, 60–62

Google Colab Python environment created by Google, based on the Jupyter Notebook architecture, that allows users connect to remote servers, and offers high-end computers with GPUs and TPUs. 28, 32, 38

Graphics Processing Unit Highly parallelized processing structures used on computer graphics, videogames and AI. 11

HTTP Request Hypertext Transfer Protocol, used to communicate between endpoints in communication networks. 24, 58, 62, 67

hyperparameter Parameters of an AI model used to control the learning process.. 29, 36, 53

Hyperparameter Optimization Process in which an AI model is explored to obtain its best hyperparameters. It usually involves search algorithms. 28

IberEval 2018 Conference on NLP, specialized in Spanish, Portuguese, Catalan, Basque and Galician languages. It featured an AMI shared-task. 5, 20, 21, 24, 27, 31–35, 38, 39, 43, 49, 50, 53–55, 67, 73

IberLEF 2021 Conference on NLP, specialized in Spanish and other Iberian languages. 5, 21, 24, 39, 40, 49, 50, 55, 73, 74

Inter-Annotator Agreement Technique to merge annotations from different annotators over the same label into a single set of annotations, trying to preserve the consensus. 50

Long-Short Term Memory Neural network architecture capable of handling sequences of data. 12

Machine Learning Branch of Artificial Intelligence which studies algorithms capable of improving themselves through the experience and the consumption of data. 7

N-grams Contiguous sequence of elements from a given text. They usually describe syllables, phonemes, letter or words. 14

Named Entity Recognition NLP task which consists on locating and classifying named entities in unstructured texts. 14

Natural Language Processing Branch of Artificial Intelligence which studies the interaction between computers and human language, creating tools to analyze natural language data and mechanisms to establish communication between people and computers using human languages. 2

overfitting Phenomena of AI models in which the model is not able to keep learning the underlying logic of the task it is being trained to solve, and it starts memorizing the training data. 43

pooler Part of Deep Neural Networks which transforms a sequence of vector into an standalone vector. It is usually done to obtain a vector with the same dimension as the number of output categories of the model. 30, 38

RESTful Application Programming Interface Programming interface which follows the REST architecture. User communicates with REST APIs by sending them calls, and REST APIs answer these calls with responses according to the call. 3

Rubrix Open source tool for tracking and iteration data of AI projects created by Recognai. 5, 51, 71

SemEval 2019 International workshop on Semantic Organization.. 20, 21

Sentiment Analysis Also known as opinion mining is a NLP task that seeks to wether a given input has a given sentiment, i.e. positive, negative, neutral, or a more complex variety. 14

Sexism Identification in Social Networks First shared task on sexism Identification in Social networks at IberLEF 2021. 21

split Division made over a data corpus to feed an AI model. Usually they are three: a training split to tune the neural network, a validation split to test different hyperparameters and a test split to measure the performance of the model. 28, 29, 36, 43, 53–55

Support Vector Machines Machine Learning algorithms specialized on classification and regression tasks. 20

Switch-C NLP language model developed by Google in 2021. 15

Temis Model proposed and produced in this thesis, capable of detecting sexist content in Spanish-written text. 23

Tensor Processing Unit Processing structure specialized on tensors, developed by Google to improve Machine Learning experiments. 11

Transformer Architecture of NLP models based on attention mechanisms that weight the influence of the different parts of the text. Usually trained over big data corpus and fine-tuned for particular tasks, thank to its reusability. 14, 16, 17, 23, 28, 30, 34, 37, 42, 43

Bibliography

[1] M. Duggan, *Online Harassment 2017*, en-US, Jul. 2017. [Online]. Available: https://www.pewresearch.org/internet/2017/07/11/online-harassment-2017/ (visited on 02/16/2021).

[2] M. Anzovino, E. Fersini, and P. Rosso, "Automatic Identification and Classification of Misogynistic Language on Twitter", en, in *Natural Language Processing and Information Systems*, M. Silberztein, F. Atigui, E. Kornyshova, E. Métais, and F. Meziane, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2018, pp. 57–64, ISBN: 978-3-319-91947-8. DOI: 10.1007/978-3-319-91947-8_6.

[3] B. Poland, *Haters: Harassment, Abuse, and Violence Online*. University of Nebraska Press, 2016, ISBN: 978-1-61234-766-0. DOI: 10.2307/j.ctt1fq9wdp. [Online]. Available: https://www.jstor.org/stable/j.ctt1fq9wdp (visited on 02/17/2021).

[4] A. Hutchinson, *Twitter Will Increase Its Use of Automation Tools as It Looks to Ensure Accuracy in COVID-19 Discussion | Social Media Today*, Mar. 2020. [Online]. Available: https://www.socialmediatoday.com/news/twitter-will-increase-its-use-of-automation-tools-as-it-looks-to-ensure-acc/574263/ (visited on 02/18/2021).

[5] *What happens when I report something to Facebook? Does the person I report get notified? | Facebook Help Centre*. [Online]. Available: https://www.facebook.com/help/103796063044734/?helpref=uf_permalink&parent_cms_id=263149623790594 (visited on 02/18/2021).

[6] D. M. Eberhard, G. F. Simons, and C. D. Fennig, Eds., *Ethnologue: Languages of the World, Twenty-Third Edition.* English, Illustrated edition. Dallas: Sil International, Global Publishing, Oct. 2020, ISBN: 978-1-55671-458-0.

[7] *Mujeres y mercado laboral en la actualidad, un análisis desde la perspectiva de género: Genéricamente empobrecidas, patriarcalmente desiguales*, es. [Online]. Available: https://eduso.net/res/revista/21/el-tema-colaboraciones/mujeres-y-mercado-laboral-en-la-actualidad-un-analisis-desde-la-

perspectiva-de-genero-genericamente-empobrecidas-patriarcalmente-desiguales (visited on 06/04/2021).

[8] *Mujer, salud mental y empleo: ¿y si fueran ellas?...* es, Section: Experiencias habitadas, Aug. 2019. [Online]. Available: https://www.grupo5.net/mujer-salud-mental-y-empleo-y-si-fueran-ellas/ (visited on 06/04/2021).

[9] M. C. Fernández Felipe, M. L. d. l. Cruz Cantos, M. Gayoso Doldan, and S. Rodríguez Tupayachi, "Carga mental en la mujer trabajadora: desigualdad de género y prevalencia", es, *Medicina y Seguridad del Trabajo*, vol. 61, no. 238, pp. 18–33, Mar. 2015, ISSN: 0465-546X. DOI: 10.4321/S0465-546X2015000100003. [Online]. Available: http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S0465-546X2015000100003&lng=en&nrm=iso&tlng=en (visited on 06/04/2021).

[10] *Violencia de género en el ámbito laboral*, es-ES. [Online]. Available: https://www2.cruzroja.es/-/violencia-de-g-c3-a9nero-en-el-c3-a1mbito-laboral (visited on 06/04/2021).

[11] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. Liu, M. Peters, M. Schmitz, and L. Zettlemoyer, "AllenNLP: A Deep Semantic Natural Language Processing Platform", *arXiv:1803.07640 [cs]*, May 2018, arXiv: 1803.07640. [Online]. Available: http://arxiv.org/abs/1803.07640 (visited on 05/25/2021).

[12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library", *arXiv:1912.01703 [cs, stat]*, Dec. 2019, arXiv: 1912.01703. [Online]. Available: http://arxiv.org/abs/1912.01703 (visited on 05/25/2021).

[13] 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. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "HuggingFace's Transformers: State-of-the-art Natural Language Processing", *arXiv:1910.03771 [cs]*, Jul. 2020, arXiv: 1910.03771. [Online]. Available: http://arxiv.org/abs/1910.03771 (visited on 05/25/2021).

[14] R. S.L, *Rubrix*, en. [Online]. Available: https://www.rubrix.ml/ (visited on 06/17/2021).

[15] N. B. a. I. Hogarth, *State of AI Report 2020*. [Online]. Available: https://www.stateof.ai/ (visited on 06/22/2021).

[16] B. Mondal, "Artificial Intelligence: State of the Art", in, Jan. 2020, pp. 389–425, ISBN: 978-3-030-32643-2. DOI: 10.1007/978-3-030-32644-9_32.

[17] *ImageNet*. [Online]. Available: https://www.image-net.org/ (visited on 06/22/2021).

[18] *Gartner forecasts the future of augmented intelligence in business*, en-US, Aug. 2019. [Online]. Available: https://techwireasia.com/2019/08/gartner-forecasts-the-future-of-augmented-intelligence-in-business/ (visited on 06/22/2021).

[19] *What are Neural Networks?*, en-us, Apr. 2021. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks (visited on 06/23/2021).

[20] *A Quick Introduction to Neural Networks*, en-US. [Online]. Available: https://www.kdnuggets.com/a-quick-introduction-to-neural-networks.html/ (visited on 06/23/2021).

[21] I. Rowan, *The State of AI in 2020*, en, Jul. 2020. [Online]. Available: https://towardsdatascience.com/the-state-of-ai-in-2020-1f95df336eb0 (visited on 06/22/2021).

[22] *Cloud Tensor Processing Unit (TPU) | Cloud TPU*, es-419-x-mtfrom-en. [Online]. Available: https://cloud.google.com/tpu/docs/tpus?hl=es (visited on 06/22/2021).

[23] N. Reddy, *A Survey on Specialised Hardware for Machine Learning*. Jul. 2019. DOI: 10.13140/RG.2.2.20697.26725.

[24] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory", *Neural computation*, vol. 9, pp. 1735–80, Dec. 1997. DOI: 10.1162/neco.1997.9.8.1735.

[25] *Cómo usar redes neuronales (LSTM) en la predicción de averías en las máquinas*, es, Nov. 2018. [Online]. Available: https://blog.gft.com/es/2018/11/06/como-usar-redes-neuronales-lstm-en-la-prediccion-de-averias-en-las-maquinas/ (visited on 06/22/2021).

[26] Y. Bengio and Y. Lecun, "Convolutional Networks for Images, Speech, and Time-Series", Nov. 1997.

[27] S. Saha, *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, en, Dec. 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (visited on 06/22/2021).

[28] S. Ruder, "An overview of gradient descent optimization algorithms", *arXiv:1609.04747 [cs]*, Jun. 2017, arXiv: 1609.04747. [Online]. Available: http://arxiv.org/abs/1609.04747 (visited on 06/22/2021).

[29] *TensorFlow*. [Online]. Available: https://www.tensorflow.org/ (visited on 06/22/2021).

[30] *Keras: the Python deep learning API*. [Online]. Available: https://keras.io/ (visited on 06/22/2021).

[31] T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean, "Large Language Models in Machine Translation", in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 858–867. [Online]. Available: https://www.aclweb.org/anthology/D07-1090 (visited on 03/29/2021).

[32] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality", *arXiv:1310.4546 [cs, stat]*, Oct. 2013, arXiv: 1310.4546. [Online]. Available: http://arxiv.org/abs/1310.4546 (visited on 03/29/2021).

[33] O. Melamud, J. Goldberger, and I. Dagan, "context2vec: Learning Generic Context Embedding with Bidirectional LSTM", in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 51–61. DOI: 10.18653/v1/K16-1006. [Online]. Available: https://www.aclweb.org/anthology/K16-1006 (visited on 03/29/2021).

[34] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation", in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162. [Online]. Available: https://www.aclweb.org/anthology/D14-1162 (visited on 04/05/2021).

[35] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations", *arXiv:1802.05365 [cs]*, Mar. 2018, arXiv: 1802.05365. [Online]. Available: http://arxiv.org/abs/1802.05365 (visited on 03/29/2021).

[36] E. M. Bender and A. Koller, "Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data", en, in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online: Association for Computational Linguistics, 2020, pp. 5185–5198. DOI: 10.18653/v1/2020.acl-main.463. [Online]. Available: https://www.aclweb.org/anthology/2020.acl-main.463 (visited on 03/29/2021).

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need", *arXiv:1706.03762 [cs]*, Dec. 2017, arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762 (visited on 03/29/2021).

[38] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun.

2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. [Online]. Available: https://www.aclweb.org/anthology/N19-1423 (visited on 03/29/2021).

[39] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", *arXiv:1910.01108 [cs]*, Feb. 2020, arXiv: 1910.01108. [Online]. Available: http://arxiv.org/abs/1910.01108 (visited on 03/30/2021).

[40] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations", *arXiv:1909.11942 [cs]*, Feb. 2020, arXiv: 1909.11942. [Online]. Available: http://arxiv.org/abs/1909.11942 (visited on 03/30/2021).

[41] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", *arXiv:1907.11692 [cs]*, Jul. 2019, arXiv: 1907.11692. [Online]. Available: http://arxiv.org/abs/1907.11692 (visited on 03/30/2021).

[42] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism", *arXiv:1909.08053 [cs]*, Mar. 2020, arXiv: 1909.08053. [Online]. Available: http://arxiv.org/abs/1909.08053 (visited on 03/30/2021).

[43] J. Canete, G. Chaperon, R. Fuentes, and J. Pérez, "Spanish pre-trained bert model and evaluation data", *PML4DC at ICLR*, vol. 2020, 2020.

[44] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language Models are Few-Shot Learners", *arXiv:2005.14165 [cs]*, Jul. 2020, arXiv: 2005.14165. [Online]. Available: http://arxiv.org/abs/2005.14165 (visited on 04/05/2021).

[45] W. Fedus, B. Zoph, and N. Shazeer, "Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity", *arXiv:2101.03961 [cs]*, Jan. 2021, arXiv: 2101.03961. [Online]. Available: http://arxiv.org/abs/2101.03961 (visited on 03/30/2021).

[46] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? ", en, in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Virtual Event Canada: ACM, Mar. 2021, pp. 610–623, ISBN: 978-1-4503-8309-7. DOI: 10.1145/3442188.3445922. [Online]. Available: https://dl.acm.org/doi/10.1145/3442188.3445922 (visited on 03/29/2021).

[47] Corby Rosset, *Turing-NLG: A 17-billion-parameter language model by Microsoft*, en-US, Feb. 2020. [Online]. Available: https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/ (visited on 04/05/2021).

[48] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, É. V. de la Clergerie, D. Seddah, and B. Sagot, "CamemBERT: a Tasty French Language Model", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7203–7219, 2020, arXiv: 1911.03894. DOI: 10.18653/v1/2020.acl-main.645. [Online]. Available: http://arxiv.org/abs/1911.03894 (visited on 03/30/2021).

[49] F. Souza, R. Nogueira, and R. Lotufo, *Portuguese Named Entity Recognition using BERT-CRF*. Sep. 2019.

[50] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile, "ALBERTO: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets", Nov. 2019.

[51] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim, "BERTje: A Dutch BERT Model", *arXiv:1912.09582 [cs]*, Dec. 2019, arXiv: 1912.09582. [Online]. Available: http://arxiv.org/abs/1912.09582 (visited on 03/30/2021).

[52] Y. Kuratov and M. Arkhipov, "Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language", *arXiv:1905.07213 [cs]*, May 2019, arXiv: 1905.07213. [Online]. Available: http://arxiv.org/abs/1905.07213 (visited on 03/30/2021).

[53] E. Strubell, A. Ganesh, and A. McCallum, "Energy and Policy Considerations for Deep Learning in NLP", *arXiv:1906.02243 [cs]*, Jun. 2019, arXiv: 1906.02243. [Online]. Available: http://arxiv.org/abs/1906.02243 (visited on 03/29/2021).

[54] D. R. So, C. Liang, and Q. V. Le, "The Evolved Transformer", *arXiv:1901.11117 [cs, stat]*, May 2019, arXiv: 1901.11117. [Online]. Available: http://arxiv.org/abs/1901.11117 (visited on 03/29/2021).

[55] L. K. Jones, *Twitter wants you to know that you're still SOL if you get a death threat — unless you're...* en, Oct. 2020. [Online]. Available: https://medium.com/@agua.carbonica/twitter-wants-you-to-know-that-youre-still-sol-if-you-get-a-death-threat-unless-you-re-a5cce316b706 (visited on 04/04/2021).

[56] *Delitos sexuales según sexo(28750)*, es. [Online]. Available: https://www.ine.es/jaxiT3/Tabla.htm?t=28750 (visited on 06/19/2021).

[57] S. R. Moreno, "El acoso sexual en el trabajo: se denuncia poco, se condena menos y las empresas no responden", es, *El País*, Mar. 2019, ISSN: 1134-6582. [Online]. Available: https://elpais.com/retina/2019/03/07/talento/1551974512_453267.html (visited on 06/19/2021).

[58] E. Fersini, P. Rosso, and M. Anzovino, "Overview of the Task on Automatic Misogyny Identification at IberEval 2018", en, in *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), co-located with 34th Conf. of the Spanish Society for Natural Language Processing (SEPLN 2018)*, vol. 2150, Seville, Spain, Sep. 2018, pp. 214–228.

[59] F. Ródriguez-Sánchez, J. Carrillo-de-Albornoz, L. Plaza, J. Gonzalo, P. Rosso, M. Comet, and T. Donoso, "Overview of EXIST 2021: sEXism Identification in Social neTworks", *Procesamiento del Lenguaje Natural*, vol. 67, no. 0, 2021, ISSN: 1989-7553.

[60] E. Fersini, D. Nozza, and P. Rosso, "Overview of the Evalita 2018 Task on Automatic Misogyny Identification (AMI)", en, in *EVALITA Evaluation of NLP and Speech Tools for Italian : Proceedings of the Final Workshop 12-13 December 2018, Naples*, ser. Collana dell'Associazione Italiana di Linguistica Computazionale, T. Caselli, N. Novielli, and V. Patti, Eds., Code: EVALITA Evaluation of NLP and Speech Tools for Italian : Proceedings of the Final Workshop 12-13 December 2018, Naples, Torino: Accademia University Press, Jun. 2019, pp. 59–66, ISBN: 978-88-319-7869-9. [Online]. Available: http://books.openedition.org/aaccademia/4497 (visited on 04/02/2021).

[61] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, and M. Sanguinetti, "SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter", in *Proceedings of the 13th International Workshop on Semantic Evaluation*, Minneapolis, Minnesota, USA: Association for Computational Linguistics, Jun. 2019, pp. 54–63. DOI: 10.18653/v1/S19-2007. [Online]. Available: https://www.aclweb.org/anthology/S19-2007 (visited on 01/01/2021).

[62] M. A. Carmona, E. Guzmán-Falcón, M. Montes, H. J. Escalante, L. Villaseñor-Pineda, V. Reyes-Meza, and A. Rico-Sulayes, "Overview of MEX-A3T at IberEval 2018: Authorship and aggressiveness analysis in Mexican Spanish tweets", Aug. 2018.

[63] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, "A System for Massively Parallel Hyperparameter Tuning", *arXiv:1810.05934 [cs, stat]*, Mar. 2020, arXiv: 1810.05934. [Online]. Available: http://arxiv.org/abs/1810.05934 (visited on 05/25/2021).

[64] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, "Tune: A Research Platform for Distributed Model Selection and Training", *arXiv:1807.05118 [cs, stat]*, Jul. 2018, arXiv: 1807.05118. [Online]. Available: http://arxiv.org/abs/1807.05118 (visited on 05/25/2021).

[65] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation", *arXiv:1406.1078 [cs, stat]*, Sep. 2014, arXiv:

1406.1078. [Online]. Available: http://arxiv.org/abs/1406.1078 (visited on 05/31/2021).

[66] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour", *arXiv:1706.02677 [cs]*, Apr. 2018, arXiv: 1706.02677. [Online]. Available: http://arxiv.org/abs/1706.02677 (visited on 06/04/2021).

[67] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization", *arXiv:1711.05101 [cs, math]*, Jan. 2019, arXiv: 1711.05101. [Online]. Available: http://arxiv.org/abs/1711.05101 (visited on 06/02/2021).

[68] Y. Sasaki, "The truth of the F-measure", en, p. 5,

[69] *cardiffnlp/twitter-roberta-base · Hugging Face*. [Online]. Available: https://huggingface.co/cardiffnlp/twitter-roberta-base (visited on 06/06/2021).

[70] F. Barbieri, L. E. Anke, and J. Camacho-Collados, "XLM-T: A Multilingual Language Model Toolkit for Twitter", *arXiv:2104.12250 [cs]*, Apr. 2021, arXiv: 2104.12250. [Online]. Available: http://arxiv.org/abs/2104.12250 (visited on 06/06/2021).

[71] I. Talavera, D. Fidalgo, and D. Vila-Suero, "System Description for EXIST shared task at IberLEF 2021: Automatic Misogyny Identification using pretrained Transformers", in *Proceedings of the Iberian Languages Evaluation Forum co-located with 37th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2021, Málaga, Spain, September 2021*, M. Á. G. Cumbreras, J. Gonzalo, E. M. Cámara, R. Martínez-Unanue, P. Rosso, J. Carrillo-de-Albornoz, S. Montalvo, L. Chiruzzo, S. Collovini, Y. Gutiérrez, S. M. J. Zafra, M. Krallinger, M. Montes-y-Gómez, R. Ortega-Bueno, and A. Rosá, Eds., ser. CEUR Workshop Proceedings, CEUR-WS.org, Sep. 2021.

[72] *EXIST*, en-us. [Online]. Available: http://nlp.uned.es/exist2021/ (visited on 06/06/2021).

[73] I. Talavera, *ignacioct/Temis*, original-date: 2021-06-13T10:00:18Z, Jun. 2021. [Online]. Available: https://github.com/ignacioct/Temis (visited on 06/13/2021).