Detection of toxicity in social media

A study on semantic orientation and linguistic structure

Trabajo de Fin de Máster

Jordina Nogués Graell

Trabajo de investigación para el

Máster en Tecnologías del Lenguaje

Universidad Nacional de Educación a Distancia

Dirigido por:

Prof. Dr. D. Víctor Fresno Fernández

Septiembre 2022

Abstract

Social networks' astonishing increase in popularity allowed users to be connected to their friends and family, in addition to being able to make new connections, either in the personal, in the academic or in the professional area. It is not to doubt the benefits that social media had during the COVID-19 pandemic, as it allowed a virtual environment for meetings and social interactions. However, social networks have a dark side which can be appreciated in forms of toxic content. This toxicity present in all kinds of social media platforms raised a warning for users, researchers, and companies, and that is why there has been an increase of studies and works related to detection and prevention of toxicity in social networks. Although the term *toxic* is not an easy one to describe, the research community worked based on their understanding or needs to define what we understand by toxic, what forms of toxic content are present online, and how to detect it using several Machine Learning approaches.

This work is based on toxicity detection on social media and focuses on the use of semantic orientation bias and linguistic structure of the messages to detect toxic content. More particularly, it is based on the term anisotropy, meaning that the word vectors are distributed through the multidimensional space oriented in a particular direction. For this reason, we are using Static Word Embeddings, as they maintain the semantic properties of the meaning of the words they represent. We performed experiments on vector proximity and orientation proximity, which allowed us to check if we could predict new toxic messages using these factors.

The second foundation of this work is to explore if linguistic structure influences in detecting toxic content. As say, if there are some words, categories or linguistic structure that have more impact in the process of toxicity detection, and how can we compound sentence vectors to address this same issue in sentence level. We performed several experiments that illustrated which linguistic content was more relevant to consider. At word level, we selected Nouns and excluded stopwords (as they present some inherent semantic orientation bias), and at sentence level we performed the Composition process in a linear way using a simple global average composition function, which calculated the average of all the vectors that compound the sentence to obtain a sentence vector.

The results allowed us to confirm that toxic content indeed shows orientation direction bias towards the same semantic space and that linguistic structure plays a role in such content.

Content disclaimer. Some of the data used for the experiments of this work, in addition to some illustrative examples, may be sensitive to the reader. Sensitive data may include offensive language, insults, profanity, threatening content, and mental health-related topics such as anorexia and suicide, which could be perceived as triggers or distressing. Do not hesitate to ask for **help** to your family or friends and contact your closest medical service or hospital if you or someone you know is a victim of abuse, online abuse, or struggle with intrusive or suicidal thoughts.

Resumen

Las redes sociales han crecido mucho en popularidad recientemente y esto ha permitido a los usuarios estar conectados con sus amigos y familiares, además de permitirles encontrar nuevos contactos tanto en el área personal como en la académica o personal. No cabe duda de que los beneficios de las redes sociales durante la pandemia de COVID-19 fueron increíbles, ya que permitieron un entorno virtual para encuentros sociales. Sin embargo, las redes sociales tienen una cara oscura que se muestra en forma de contenido tóxico. Esta toxicidad que está presente en muchas redes sociales ha alarmado tanto a los usuarios como a los investigadores y las compañías y por ello se ha dado un incremento en los estudios y trabajos relacionados con la detección y prevención de la toxicidad en las redes sociales. Aunque no es fácil de describir qué se entiendo por *tóxico*, la comunidad científica ha trabajado según su propio entendimiento de lo que abarca el término o según sus necesidades a cubrir para definir lo que se entiende por tóxico y qué formas de contenido tóxico existen en línea, además de cómo detectar la toxicidad utilizando diferentes aproximaciones de *machine learning*.

Este trabajo se basa en la detección de toxicidad en las redes sociales y se centra en el uso del sesgo en la orientación semántica y la estructura lingüística de los mensajes para detectar contenido tóxico. En concreto, nos basamos en el término anisotropía: el significado de los vectores de palabras se distribuye según una orientación particular en el espacio semántico. Por ello, utilizamos embeddings estáticos (*Static Word Emgeddings*), ya que permiten mantener las propiedades semánticas del significado de las palabras que representan. Siguiendo esta idea hemos realizado varios experimentos en proximidad vectorial y proximidad de orientación para determinar y predecir contenido tóxico.

El segundo pilar de este trabajo se basa en explorar si la estructura lingüística influencia la detección de contenido tóxico. Es decir, si hay categorías gramaticales o estructuras lingüísticas que tienen un impacto en la detección de la toxicidad y cómo se pueden crear vectores de frase mediante composición para abordar este mismo proceso a nivel de frases. Para ello, hemos realizado

diferentes experimentos para demostrar qué estructura lingüística era más relevante para tener en cuenta. A nivel de palabra, hemos seleccionado los sustantivos, además de excluir las *stopwords* (ya que presentan sesgos inherentes en la orientación semántica); a nivel de frase, hemos compuesto los vectores de las palabras de manera linear utilizando una función de promedio general (*global average*), que nos ha permitido calcular el vector promedio de los vectores de las palabras que componen la frase para obtener el vector de la frase.

Los resultados obtenidos de esta investigación nos han permitido afirmar que el contenido tóxico presenta una orientación direccional en el espacio semántico, además de permitirnos demostrar que la estructura lingüística también juega un papel relevante en este tipo de contenido.

Advertencia de contenido. Algunos de los datos utilizados para los experimentos de este trabajo, además de los ejemplos ilustrativos en algunas de las secciones del mismo pueden considerarse susceptibles o sensibles al lector. La información contiene lenguaje ofensivo, insultos, obscenidades o amenazas además de mostrar tópicos relacionados con la salud mental como la anorexia y el suicidio, que podrían considerarse inquietantes o podrían fomentar estas conductas. No dudes en pedir **ayuda** a tus familiares o amigos y contacta los servicios sanitarios o hospitales más cercanos si tú o alguien que conoces es víctima de abuso, abuso en línea o tiene pensamiento intrusivos o suicidas.

Contents

Index of tables

1. Toxic behaviour on social media

In this section we introduced the problem of toxicity in social media as well as the problem of definition and the challenges of detecting this behaviour online. We also presented our objectives and hypothesis and the contributions of our research. Finally, we illustrate the chapters' division.

1.1. Introduction

Currently, social networks are more popular than ever and with their increased popularity, the incidences of negative behaviour among their users are rising. Freedom of communication leads sometimes to abusive and undesired behaviour such as hate speech, racism, abusive language, doxing, or offensive speech (Paraschiv 2020). Toxicity detection is a task that many researchers and industries are focusing on. Achieving a secure online place where everyone is respected is among the goals and, for that reason, there has been a huge increase in toxicity (and other forms of abusive language) papers, research, tasks, and challenges. On the other side of the spectrum, there has been also an increase of other kinds of more concerning behaviours online, such as suicide and anorexia related topics. Those cases are equally important to toxic behaviour detection to make social and online platforms a safe and enjoyable place, not an environment of hatred or somewhere to be lost for people who experiment mental health disorders.

In this thesis we present a study that considers the potential use of *semantic orientation* in vector representations of words and the *linguistic structure* of the messages for detecting toxic information. The main reason is that techniques that work best nowadays are based on models that are not interpretable (mainly based on Deep Learning, DL). Our hypothesis is that if we are capable of understanding if those factors influence (and how do they do it), we will be able to improve the systems based on DL because we will be able to make them explicitly take this information in account.

1.2. Hypothesis and Objectives

We formulate the hypothesis in relation to *semantic orientation* and *syntactic structure*. Thus, we explore our hypothesis following two research questions.

RQ1: *Does semantic representation in toxicity have any kind of orientation bias in the semantic representation space*?

In relation to that we also ask ourselves: Do toxic terms or messages tend to have a semantic orientation bias towards a specific direction in the semantic representation space? If the answer to those question is "yes" then we would wonder: Could we use such semantic orientation to detect new toxic messages?

RQ2: *Does toxicity have any kind of inherent syntactic structure? If so, could we use such structure to detect new toxic messages?*

As a general objective to this thesis, we present the study of the problem of what kind of information is the most relevant to detect toxic messages in social media. For that, we develop different systems able to detect toxic messages where we evaluate the different studied variables.

1.3. Contributions

In this thesis we contributed to the research community in the context of semantic orientation and linguistic structure in toxicity detection tasks. Regarding the first one, we showed that toxic words and messages have a biased semantic orientation in the semantic vector space which could be used as a feature to detect other toxic messages (even though it needs improvement). In the second case, we showed that there are some grammatical categories more biased to toxicity, such as Nouns, and in a second-place Adjectives and Adverbs; and that the way sentence vector composition is performed is not as relevant (at least not in our case, as we used a sequential and linear order with similar results in all cases). In this sense there is still space for further investigation as we demonstrated that there are some categories that have relevant information, and we can assume that the implicit linguistic structure in the use of those categories can contain crucial information in the task of toxicity detection as well. Such open window showed that there is room for improvement taking in consideration such categories and the linguistic structure of the messages in which they appear.

Finally, this thesis also contributed to the research community through an extensive and detailed state-of-the-art investigation. We studied tasks of toxicity detection on social media but, in addition we developed parallel investigations in other related tasks to toxicity: *abusive language*, *anorexia*, *anxiety*, *emotion detection*, *hate speech*, *misogyny*, *morbidity*, *offensive language*, *sentiment analysis*, and *suicide*. Throughout the literature review we compiled and showed several methods used by different authors in the each of the tasks. We also presented the used datasets, and we compared the different approximations and methodologies that were used, which included classical Machine Learning (ML) methods, DL and even the most recent Transformers-based algorithms. Thus, this literature review is a strong and concise summary and comparison that could be used for further studies and investigators of those areas as well as an overview of the approximations used in the presented tasks. We also wanted to note that, due to the extension and number of approximations and collections used along the state-of-the-art, in certain places of the study we presented absolute values of the different metrics (F1-Score, Precision, Recall…) and different collections (which we described along the state-of-the-art revision) to indicate the degree of advance in the studied task. That aimed to show if an approximation in a particular collection has high values in a certain score, but not to compare different approximations that use different collections: in that sense we compared *results* with absolute values. However, we also illustrated different systems' approximations in distinct corpora in related tasks: in that sense we talked about *performance*, as we illustrated how a model performed, what algorithms were better, which dataset characteristics were more relevant or what information was best to consider in the tasks.

1.4. Structure of the document

The rest of the work is divided as follows. *Chapter 2* offers a literature review of toxicity tasks and other related tasks, and a contextualization of text representations in such tasks. *Chapter 3* discusses the concepts of semantic space vectors and semantic orientation, in addition to present the methodology and design of the experiments, we also describe the proposed algorithm based on semantic composition and linguistic structure. *Chapter 4* focuses on the three experiments based on semantic proximity: vector proximity, orientation proximity, and nearest neighbours. Finally, *Chapter 5* concludes the thesis and discusses future work.

2. State of the art

The problem of detection of toxicity has been studied in detail recently due to the increasing use of social networks and their easy accessibility by everyone. Both companies and users are concerned by the increasing presence of such behaviours, that is why it is so important to develop systems that allow the users to be as comfortable as possible while online. The term *toxicity* or *toxic behaviour* is complex and polysemic, and can include many forms of bad behaviours, even in the literature authors do not understand it or use it in the same way. That is why in addition to purely toxicity studies, it is important to focus on other forms of bad behaviours such as tasks related to detection of abusive, offensive, or aggressive language; and forms of attack to targeted groups, such as misogyny, racism, or homophobia. Finally, similarly related to toxicity but on the other side of the problem, it is important to pay especial attention to concerning behaviours that happen in social media, such as tasks oriented to detect anorexia or bulimia, and to suicide prevention, to prevent future harm to the user himself or herself.

In this section we explored each of the tasks related to toxicity detection to observe how different studies and researchers have studied these problems. From more basic forms of traditional ML classifiers to more complex and newer Transformer models, the interest of these kind of tasks has been increasing greatly. This can even be appreciated in the numerous Challenges, such as the challenges on Kaggle[1]: “Toxic Comment Classification Challenge[2]”. There are different Evaluation Campaigns as well: SemEval[3], including tasks on idiomatic detection[4] (task 2), sarcasm detection[5] (task 6); or Language-oriented tasks: IberEval[6], for Iberian languages, or GermEval[7], for German.

[1] https://www.kaggle.com/

[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

[3] https://semeval.github.io/

[4] https://sites.google.com/view/semeval2022task2-idiomaticity

[5] https://sites.google.com/view/semeval2022-isarcasmeval

[6] https://sites.google.com/view/ibereval-2018

[7] https://germeval.github.io/

2.1. Tasks related with social media toxicity detection

In this section we reviewed different approaches and studies that have been made in relation to different kinds of toxic or concerning behaviours online. There is a subsection for every kind of concerning behaviour: *Abusive Language*, *Anorexia*, *Anxiety*, *Emotion Detection*, *Hate Speech*, *Misogyny*, *Morbidity*, *Offensive Language*, *Sentiment Analysis*, *Suicide*, and *Toxicity*.

2.1.1. Abusive Language

In the task of detecting abusive language, Founta et al. (2018) implemented a unified DL architecture (Recurrent Neural Network, RNN) (Pavlopoulos et al. 2017; Gao and Huang 2017; Pitsilis et al. 2018; Zhang et al. 2018) able to digest and combine any available attribute for the task of recognize various types of abusive behaviour in Twitter by capturing subtle, hidden commonalities and differences between the various abusive behaviours with the same model. They used pre-trained Word Embeddings (WE) (GloVe; Pennington et al. 2014) for each attribute (text, user, network, tweet). They used four datasets: A cyberbullying dataset (Chatzakou et al. 2017) that contains 6091 tweets divided into 8.5% *bully*, 5.5% *aggressive* and 86% *normal*; an offensive dataset (Waseem and Hovy 2016) consisting of 16059 tweets divided into 12% *racism*, 20% *sexism* and 68% *none*; a hate dataset (Davidson et al. 2017) that contains 24783 tweets that have 6% *hate*, 77% *offensive* and 17% *neither* class; and a sarcasm dataset (Rajadesingan, Zafarani and Liu 2015) with 61075 tweets divided as 10.5% *sarcastic* and 89.5% *normal*. The results showed that on the cyberbullying dataset they observe that using a single set of attributes (text or metadata) they achieve better ROC AUC[8] (0.92 or 0.93, respectively) but worse Accuracy (0.89 or 0.88, respectively); however, the interleaved[9] training

[8] For a detailed explanation on evaluation metrics, such as ROC AUC, see Chapter 3 (section 3.2.2.). We would like to note that in the task of toxicity (and other forms of toxic or concerning behaviours online) Recall would be the best metric to evaluate the systems with. A high Recall on the toxic category, means that we are not missing any of the messages labelled as such, thus, Recall should be the main metric of evaluation in these tasks.

[9] As they take in consideration two sets of attributes (text and metadata), they approached a training for each of the sets of attributes separately and then concatenating the two-pretrained

substantially outperforms the baseline (ROC AUC 0.96, Accuracy 0.92); in terms of Recall they also achieve the best results with the interleaved learning models with a value of 0.92 in contrast to the 0.88 from the baseline. On the offensive and sarcasm datasets, they largely outperformed the previous results, which did not consider metadata. In sarcasm they reach a ROC AUC of 0.98 (compared to the 0.7 from the previous results), and a Recall that increases from 0.81 (baseline) to 0.87 (interleaved model); in the case of offensive dataset, there was no ROC AUC reported in the previous results, but the combination of text and metadata (interleaved model) reached a Precision and Recall of 0.89 (compared to 0.87 of the baseline). Finally, in the hate dataset the interleaved model reached a ROC AUC of 0.98 (in the baseline it was 0.66), while the Precision (0.96) and the Recall (0.97) increase since the baseline (0.89 and 0.90 respectively). They concluded that the best performance was reached when all attributes were used (both text and metadata), and this also demonstrated that the metadata did not overlap the information from the text. As it could be appreciated, the interleaved system outperformed all the other systems in all metrics (see the ROC AUC metric as a general overview of the performance), however, the substantial benefits in terms of Recall could be observed in the sarcasm dataset (0.97 Recall) and in fewer terms in the offensive dataset (0.87 Recall). Also, it was interesting to note that the datasets were not balanced, as most of the messages were *not toxic*, then these results suggest an interesting improvement as most of them reach values of 0.80 and above.

On the other hand, Karan & Snajder (2018) used a Linear Support Vector Machine (LSVM) (Xu et al. 2012; Dadvar et al. 2013; Schofield and Davidson 2017) in the task of investigating to what extent abusive language detection could benefit from combining training sets and sharing information between them through domain adaptation techniques. They relied on the simplest text representation with unigram counts, and their results showed that having in-domain data was crucial for achieving good performance. In this study they used 9 publicly available datasets in English: Kol (Kolhatkar et al. 2018), Gao (Gao and Huang 2017), TRAC (Kumar et al. 2018a, b), Was2 (Waseem 2016), Was1

models together. That allows them to *train the full network simultaneously while mitigating the drawbacks* (Founta et al. 2018).

(Waseem and Hovy 2016), Wul1, Wul2, Wul3 (Wulczyn et al. 2017) and Kaggle[10]. They finally concluded that for most datasets, their proposal (FEDA, Frustratingly simplE Domain Adaptation) leaded to performance improvements, and for six out of nine datasets there was at least one augmentation dataset which gave a statistically significant performance improvement. Thus, the domain adaptation technique improved results on smaller datasets, when the augmentation was made with a larger dataset from a different domain.

Finally, Murgado et al. (2021) created a prototype of social monitor aimed at detecting inappropriate behaviour on social networks (such as Twitter and YouTube). The datasets were extracted by themselves using the Twitter official API[11] and the YouTube's official API[12]. This tool included a database where the user could store the posts from the social networks to later analyse them with previously trained NLP systems. In particular, the tool integrated two systems based on document classification to identify anorexia and offensive language. It relied on traditional ML systems such as SVM (Noble 2006) and other state-of-the-art methods based on Transformer models such as BERT (Devlin et al. 2018). Even though they did not present results or conclude with any relevant information about the performance of the system (their work is mostly a description of the prototype), it could be a useful tool for monitoring inappropriate behaviour online and a beneficial support to models and systems.

2.1.2. Anorexia

In the case of anorexia, López-Úbeda et al. (2019) used classical ML approaches for automatically detecting anorexia symptoms (binary classification) with the TF-IDF technique of text representation. They compiled their own corpus, SAD (Spanish Anorexia Dataset, López-Úbeda et al. 2019), from Twitter. This corpus has 5707 Spanish tweets divided in 2707 tweets referring to *anorexia* and 3000 tweets of *control* (not referring to anorexia), which showed that they created a highly balanced corpus. Their results showed that the performance was very

[10] Toxic Comment Classification Challenge: https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data

[11] https://developer.twitter.com/en/docs/twitter-api

[12] https://developers.google.com/youtube/v3

similar in all systems, although SVM and Multilayer Perceptron (MLP) (Hornik et al. 1989) were the only ones that obtained Recall values above 0.9 (0.93 for both systems) in contrast to the 0.82-0.89 values of the other models such as Random Forest (RF), Naïve Bayes (NB), Logistic Regression (LR) and Decision Tree (DT). It was interesting to see that, in their case, the Recall was slightly higher than the Precision (which oscillated between 0.79-0.89) however, as we could appreciate earlier, in tasks such as anorexia or toxicity detection, Recall would be a better metric to consider in the moment of evaluating systems' performances. As we could observe, the study was performed in a balanced corpus, so one of the conclusions they extracted from the error analysis was that in the cases where textual information was poor or where rhetorical figures such as irony and sarcasm were used, it would be interesting to explore other techniques to check if that would improve the performance of the models.

In a second study, López-Úbeda et al. (2021) decided to apply transfer learning techniques using Transformer-based models: BERT (Devlin et al. 2018) and XLM (Lample et al. 2019). They were interested in comparing the performance of multilingual and monolingual approaches, that is why they decided to use BETO (a BERT model trained on Spanish texts, Cañete et al. 2020) on one side, and M-BERT (Devlin et al. 2018, Pires et al. 2019) and XLM on the other. In this study they used the same corpus as earlier, the SAD. Their results showed that BETO trained on Spanish texts outperformed multilingual models in the Spanish anorexia task with a Recall and F1-Score of 0.94. However, the other systems showed results not far from that value: LSTM achieved 0.93 Recall and 0.91 F1-Score, meanwhile BiLSTM, CNN, XLM and BERT achieve a Recall of 0.93 and F1-Scores of 0.91-0.93. That illustrated the importance of task- and language-targeted models as there were words not found in the multilingual models that were important for the task.

As it could be appreciated in Founta et al. (2018), they used an interleaved training method that showed the best overall performance in the sarcasm dataset than in the other datasets and models illustrated by López-Úbeda et al. (2019, 2021) who, in the first one, used classical models (SVM and MLP), and in the second case BETO. It could be appreciated that in the tasks of anorexia detection performed by López-Úbeda (2019, 2021) the results did not differ in high terms between classical and Transformer-based models, although they were both achieving slightly worse performance than the interleaved model by Founta et al. (2018) in the sarcasm dataset. This illustrated that the presence of language-specific models in the case of anorexia detection in Spanish had better results than classical models and even better performance than the interleaved learning in English datasets of related tasks such as cyberbulling, hate speech and offensive language (taking also in consideration that those datasets were not balanced). However, the performance in the sarcasm dataset was improved in the case of the interleaved learning model by Founta et al. (2018). Sarcasm is more different from the other toxic behaviours we are focusing our attention to as it can present more implicit forms of toxicity and a wider target group. This could be the reason why an interleaved learning method boosted the performance in the sarcasm dataset but may not be as useful in the other datasets. As appreciated, we could affirm that BETO achieved the best performance so far in a balanced Spanish dataset, followed by many other DL systems (BERT, XLM, CNN and BiLSTM) or ML models (SVM and MLP), which boosted the outcome by its language-specific characteristics.

2.1.3. Anxiety

From studies on anxiety, Shen & Rudzicz (2017) explored the detection of anxiety through personal narratives. In their study they implemented N-gram language modelling, vector embeddings, topic analysis, and emotional norms. They collected a dataset from Reddit using the Reddit API, which consisted of 22808 posts over three months that included 9971 anxiety-related posts ("*Anxiety*") and 12837 general posts ("*Control*"). During their study, they showed the effectiveness of vector-space representations and LDA (Latent Dirichlet Allocation, Resnik et al. 2015) features, which correlated anxiety and specific LDA topics, such as school and alcohol (and drug) consumption. The best results were achieved with the combination of a Neural Network (NN) classifier and N-grams, where they achieved an 0.91 Accuracy and 0.92 Precision. When combined with lexicon-based features, such as LIWC (Linguistic Inquiry and Word Count, Pennebaker et al. 2015) features with N-gram probabilities, they achieved 0.98 Accuracy and 0.99 Precision. They also scored an Accuracy value over 0.90 using WE (Word2Vec). In terms of Recall, although they did not illustrate it in the tables of results, they commented that the NN achieved Recall values over 0.90, while the SVM classifier produced the lowest Recall, between 0.79 and 0.90. The authors concluded that the use of LIWC, LDA and N-gram (uni- and bigram) features elevated the Accuracy to levels of 0.98 (note that it is a balanced dataset, so that value showed relevant results). In addition, LDA topics could help in identifying which topics people with anxiety or other mental illnesses discuss online; while the N-gram features allowed to find lexicons related to feelings and first-person references (singular pronouns represented in the anxiety group); finally, the authors also pointed out the study of collocations, as the writers of anxiety-related posts were looking to find other people sharing similar experiences.

In this case, it could be appreciated that the Recall values oscillate between 0.79-0.90 with the SVM classifiers and above 0.90 when using NN. Even though they did not specify the precise values, we could still claim that the best performance achieved so far have been the one by Founta et al. (2018) in the sarcasm dataset. Taking in consideration more related datasets and tasks such as anorexia detection, we could reckon that the ML systems by López-Úbeda (2019, 2021) and, in even higher terms, the BETO model (López-Úbeda 2021) still outperformed the presented model -being both datasets balanced. Thus, although Shen & Rudzicz (2017) used a NN model and considered interesting additional data such as LDA topics and LIWC features, they achieved the lowest performance in comparison with the other authors.

2.1.4. Emotion Detection

In the case of emotion detection, Plaza-del-Arco et al. (2021d) detailed the overview of the emotion classification of tweets task (EmoEvalEs[13]) using the EmoEvent corpus (Plaza-del-Arco et al. 2020). This corpus is a multilingual emotion corpus of around 6000 tweets related to domains such as *entertainment*,

[13] https://competitions.codalab.org/competitions/28682

catastrophe, *political*, *global commemoration*, and *global strike*. Each instance is labelled with the main emotion expressed according to the following categories: *anger*, *disgust*, *fear*, *joy*, *sadness*, *surprise* and *neutral or no emotion*. Moreover, the tweets were annotated as *offensive* or *non-offensive*, thus this corpus allowed experimentation in multi-class emotion classification and in binary offensive classification. In the corpus we could observe 706 *offensive* tweets in Spanish and 518 in English, which meant that only 20% of the corpus contained offensive content. Most of the teams used NN, in particular Transformer-based models, in two ways: as encoders to obtain contextualized sentence embeddings features from the text, and finetuning the pre-trained models on the task of emotion detection. The Macro-Recall values oscillated between 0.50 the lowest and 0.73 the best one. The best team, GSI-UPM (Vera, Araque and Iglesias 2021) studied the combination of different features (TF-IDF, N-grams, sentiment values, and the provided *event* and *offensiveness* column from the dataset) with a fine-tuned XLM-RoBERTa. The authors commented that even though the best values were achieved by the RoBERTa model they also presented results with a LR model which was not much lower than the RoBERTa's.

The results showed that the combination of linguistic information confirmed the benefits of opting for hybrid solutions, which improved the results achieved by the three teams that considered *offensive* and *event* information: GSI-UPM, haha (Li 2021) and WSSC (Vitiugin and Barnabò 2021). As a final remark, the authors pointed out an improvement of performance in terms of Macro-F1 of the best system from that year (0.72) in comparison with the winner from the previous year (0.45 Macro-F1). There was no further description on the system from the previous year, but this value improvement was a significant result in the research community and in this kind of tasks as it showed an advance in the state-of-the-art research every year that passed, in addition to allowing more relevant work to be done in these tasks.

2.1.5. Hate Speech

Next, the case of hate speech has been widely studied and approached by both classical ML techniques (Davidson et al. 2017, MacAvney et al. 2019) and DL techniques (Kovács et al. 2021, Pham et al. 2020, Zimmerman et al. 2018). Mostly the task consisted of automatic detection of online hate speech.

Regarding classical approaches, it was possible to appreciate that Davidson et al. (2017) created a hate speech lexicon to collect tweets containing hate speech keywords to train a multi-class classifier to distinguish between these different categories: *hate speech*, *offensive language* and *neither*. From this corpus (85.4 million tweets in total) they took a sample of 25000 labelled tweets. They used unigram, bigram, and trigram features, each weighted by its TF-IDF; and as classifiers they tested LR, NB, DT, RF, and LSVM (see Aggarwal & Zhai 2012 for a detailed survey of text classification algorithms). Their results showed that LR and LSVM tended to perform significantly better than the other models. Further, LR with L2 regularization seemed to be the one that more readily allowed to examine the predicted probabilities of each class. The authors stated that the best model performed an overall Recall of 0.90, however they saw that almost 40% of the *hate speech* was misclassified (as the Recall from the hate speech class was 0.61). This suggested that the model was biased towards classifying tweets as less hateful or offensive than the human coders. However, the Recall from the *offensive* class was 0.91. In this case, it could be appreciated that the performance for the *hate* category was one of the lowest seen until now, although the *offensive* class was high and competitive in comparison with the performances from previous authors. As a final remark, the authors stated the usefulness of terms when distinguishing between *hate* and *offensive language* but, while there were terms that could be used in both cases, most of them were usually associated with *hate*. This was an interesting remark given that the Recall for *hate* was lower than the *offensive* one, which the authors justified saying that hate could be used in more different ways (to a person or group, targeted, in a conversation, etc.) and, thus, is a more complex behaviour to classify.

In the case of MacAvney et al. (2019), they proposed a multi-view LSVM for the classification of hate speech. They used 4 publicly available datasets for their experiment: Stormfront: 10568 sentences labelled as 11% *hate*, 86% *not* *hate*, 2% *relation* and 1% *skip* (de Gibert et al. 2018); TRAC: 15869 Facebook comments labelled 69% as *non-aggressive*, 16% *overtly aggressive* and 16% as *covertly aggressive* (Kumar et al. 2018a,b); HateEval[14]: 19600 tweets distributed as 43% *hate* and 57% *not hate*; and HatebaseTwitter: 24802 tweets divided as 5% *hate*, 76% *offensive* and 17% *neither* (Davidson et al. 2017). For their study, each type of feature was fitted with an individual LSVM classifier, creating a view-classifier for those features. They focused their attention on the interpretability problem, so they decided to use a Multi-view SVM approach, which showed near-state-of-the-art performance while being simpler and producing more easily interpretable decisions than NN. Their approach was keyword-based, using an ontology or dictionary, that contained potentially hateful words that were identified. Their results oscillated between 0.50-0.80 Macro-F1, being the mSVM the best ranked in most of the cases. That was in the line as seen in Plaza-del-Arco (2021d) when they described the best systems from that year and the previous year, seeing that the performance increased by 60% in 2021 in comparison with 2020. In this case, the performance from MacAvney et al. (2019), even though they commented the Macro-F1 oscillation of their proposed model, it outperformed the models' performances from Plaza-del-Arco (this result was achieved by the mSVM classifier in the Stormfront dataset). In addition, they described that the best working layers seemed to be the unigram word-level view and the 3- to 5-gram character level view. Also, their error analysis showed that the misclassified posts were based on mutual linguistic features (surrounding context), semantic features (implicit hate), and length (short posts).

Kovács et al. (2021) also noted issues with NN regarding the explainability and transparency, even so, they used transfer learning using DL models: Long Short-Term Memory (LSTM) (Badjatiya et al. 2017), Convolutional Neural Network (CNN)-LSTM (Van Huynh et al. 2019) and RoBERTa (Liu et al. 2019); and classical models: MATLAB implementation of K-nearest neighbours (K-NN), AdaBoost, Linear Discriminant, LR, RF, and SVM. They used three corpora for the study: HASOC (Mandl et al. 2019), OLID (Zampieri et al. 2019, 2020) and HateBase (Davidson et al. 2017). HASOC (Hate

[14] https://competitions.codalab.org/competitions/19935

Speech and Offensive Content) was created in three languages (English, German and Hindi) from Twitter and Facebook. The English data consisted of 7551 instances. OLID contained 13240 tweets, 4400 (33%) labelled as *hateful* or *offensive*. Finally, HateBase contained 24783 tweets annotated into three classes: *hate speech* (1430 tweets), *offensive language* (19190 tweets) and *neither* (4163 tweets). The results showed that with a combined model of RoBERTa trained on HASOC data with pre-trained embeddings from FastText (Mikolov et al. 2017, Bojanowski et al. 2016, Joulin et al. 2016) on all datasets, they achieved the best performance: 0.79 Macro-F1. This result was also observed in MacAvney et al. (2019) with this same dataset, even though they did not use Transformer models, they achieved 0.80 Macro-F1. This demonstrated that there was still place of improvement for both classical (or ensemble of classic) and Transformer-based models in the task of hate speech detection as it could be observed that two systems so different in architecture still achieved very similar results.

Also, Pham et al. (2020) adapted the general-purpose RoBERTa language model to a specific text classification task by fine-tuning a pre-trained model on a smaller dataset. The dataset used was the HSD (Vu et al. 2020) that included 25431 samples (posts or comments from Facebook) annotated as *hate speech* (709 tweets labelled as "hate"), *offensive but not hate speech* (1022 tweets labelled as "offensive") and *neither offensive nor hate speech* (18614 tweets labelled as "clean"). They tuned PhoBERT on a domain-, language-specific (Vietnamese) dataset by re-training the model on the Masked Language Model (MLM) task. In terms of Macro-F1, we could observe that the results oscillated between 0.68 and 0.69. As we could appreciate, the performance was lower than the one in MacAvney et al. (2019). However, it is important to note that in this case, Pahm et al. (2021) were working on Vietnamese language, thus, even though the performance was worse it was a competitive result taking in consideration that we were talking in terms of minority languages. As a final remark, the results showed a boost on performance, achieving a new-state-of-the-art on Vietnamese Hate Speech Detection campaign with 0.72 F1-Score.

Finally, Zimmerman et al. (2018) proposed an ensemble model with NN to better classify hate speech. The results showed a 5-point improvement in F-measure when compared to the original work on a publicly available hate speech evaluation dataset. A benefit of CNN classifiers and WE was the ability to consume sequential tokens through concatenation of token embeddings into a matrix, in contrast to N-gram features which lose the notion of position in a text. Plus, CNN classifiers, in theory, can consume variable length documents. Their results showed a significant improvement in the ensemble methods over single models, also they showed that ensemble models performed better with high variance. The best ensemble model achieved 0.72 F1-Score, which showed an improvement of nearly 2% over the best individual model. Thus, they demonstrated the importance of ensemble models in hate speech detection as they improve the results by benefiting from the positive points of every individual classifier. This performance was parallel to the one achieved by Pham et al. (2020), in their task of hate speech detection in Vietnamese. That could tell us two relevant conclusions: in one hand, that Pham et al. (2021) achieved competitive results in their language-specific task, which was a great improvement in other languages than English. On the other hand, in the case of Zimmerman et al. (2018), considering that the article was from 2018, that there was still plenty of path to cover in hate speech research. As we have seen that hate speech was a difficult category to classify, it needed further research with as many classifiers and methods as possible.

2.1.6. Misogyny

Another category that increasingly interested the research community was the task of detection of misogyny. In some cases, classical models seemed to perform better (Frenda et al. 2018, Shushkevich & Cardiff 2019, Pamungkas et al. 2020), while in other cases NN outperformed them (Plaza-del-Arco et al. 2021b, Samghabadi et al. 2020, Mina et al. 2022). It was also interesting to note that in some studies both classical and NN were either combined (Aldana-Bobadilla et al. 2021, Ahluwalia et al. 2018, García-Díaz et al. 2021) or had similar performances (Guest et al. 2021).

First, it was fascinating to check the cases where classical models outperformed NN, as in Frenda et al. (2018), where they focused on the detection of misogyny based on stylistics (character N-grams), semantics (sentiment information), and specific topic information and context of the text, in addition to the use of lexicons (meaningful words) in both Spanish and English tweets. The corpus was the one used in the AMI (Automatic Misogyny Identification, Fersini et al. 2018b) campaign. The dataset was composed of 3307 Spanish tweets and 3251 English tweets annotated as *misogynous* or *not misogynous*. In this study, the tweets were represented by a vector composed of all specific features (set of lexicons), pondered with Information Gain and character N-grams, and weighted with TF-IDF measure. In addition, the data was lemmatized and stemmed with Porter stemmer. The challenge was divided in two subtasks: task 1 consisted in a binary classification between *misogynous* and *not misogynous*, and task 2 consisted in clarify the behaviour and target of the misogynistic messages. In our case, we were interested in the first subtask. Their conclusions showed that an ensemble technique gave promising results achieving 0.80 Accuracy in English and 0.79 in Spanish.

Shushkevich & Cardiff (2019) experimented with different classical ML models (SVM, NB and LR), ensembles of classical models, and DL models (LSTM and CNN) in different languages (English, Spanish and Italian) to identify which features helped to recognize misogynistic tweets. For English and Spanish, they used the AMI dataset, meanwhile for English and Italian they used Evalita (Fersini et al. 2018a). The English dataset of Evalita consists of 1785 *misogynistic* tweets and 2215 *non-misogynistic*; as per the Italian one, it includes 4000 tweets (46% *misogynistic*). After this study, they concluded that classical models (and ensembles of those models) allowed to achieve higher results than the models based on NN in case of misogyny identification in Twitter.

In the case of the task from the AMI dataset, we were also just focusing on the first subtask: binary classification of misogynous content. In this case, the best team, 14-exlab (Pamungkas et al. 2018) used SVM models: with radial basis function for the English dataset and a SVM with a linear kernel for the Spanish dataset. They benefited from several lexical features: swear word count, swear word, sexist slurs and hashtags presence. They achieved 0.91 Accuracy in the English dataset -there were no results for the Spanish one. In second case, with the Evalita dataset, again we were only focusing on subtask 1 (binary classification). The best results of the English dataset were achieved by the

Hateminers team, who used a LR model, achieving an 0.70 Accuracy (again, there were no reported result for Italian neither). The results showed that classical ML models (especially ensembles of this models) allowed to achieve higher results than the models based on NN in case of misogyny identification in Twitter. The results for the English dataset in the AMI task were better than the ones presented earlier from Frenda et al. (2018) who achieved Accuracy values of 0.80 with a SVM model. In this case, the results of the classic models were outperforming the early ones: with an Accuracy value of 0.91 in English. In addition, it was unfortunate that there were no results reported for the Spanish dataset, which would also contribute greatly to the study of misogyny detection in other languages.

Pamungkas et al. (2020) wanted to focus on the most important features to detect misogyny and the issues which contribute to the difficulty of misogyny detection. For that, they studied the relationship between misogyny and other abusive language phenomena (sexism, hate speech and offensive language) by doing cross-domain classification experiments, in addition to explore multilingual environments. They proposed 3 research questions with several approaches. The first research question was: *What are the most predictive features to distinguish between misogynistic and non-misogynistic content in social media?* To face it, they investigated state-of-the-art systems on datasets from the AMI tasks. They found that most submitted approaches were using traditional ML systems. Thus, they investigated the most predictive features for the classifiers to predict misogyny. The second research question was: *How is misogyny related to other abusive phenomena, and how do they inform each other towards detection of abusive language at large?* To answer this question, they collected datasets of hateful language that are somewhat related to each other in terms of topic and target, as well as datasets different in nature. Finally, the third research question: *Is the knowledge about misogyny learned from one language informative to predict misogyny in other languages?* As hateful language and misogyny is not something particular for English, they experimented with cross-lingual environment to detect misogyny. For that they used two datasets (AMI and Evalita) in three different languages (English, Spanish and Italian), and built a system to classify misogyny in cross-lingual settings. The systems were two kind of DL architectures including two RNN-based models: LSTM and Gated Recurrent Unit (GRU) (Cho et al. 2014); and Transformer-based model: BERT; however, the best results were achieved by simple classifiers (SVM and LR) with manually engineered features (bag of words, bag of hashtags, bag of emojis, swear words, sexist slurs, women-related words and hate words lexicon).

The results of their experimentation showed that the best systems in both campaigns were simple classifiers: the best values were achieved by a SVM model with RBF kernel with 0.91 Accuracy, 0.87 Precision, 0.91 Recall and 0.89 F1-Score for English. The Accuracy results were in the same line as the ones presented in Shushkevich & Cardiff (2019) for this same dataset, who also scored 0.91 Accuracy. In this case it was interesting to appreciate the results from the Spanish dataset, as the overall best results were achieved by the SVM with linear kernel with 0.81 Accuracy and F1-Score, 0.80 Precision and 0.82 Recall. Note that, in this case, the best Recall was achieved by the BERT model with a value of 0.87. These results could be compared with the ones from Frenda et al. (2018), who scored 0.80 Accuracy in the English dataset and 0.79 in the Spanish one. As it could be appreciated, the Accuracy value of the English dataset was greatly outperformed by Pamungkas et al. (2020) as well as Shushkevich & Cardiff (2019). However, the results compiled with the Spanish dataset were similar in both Pamungkas et al. (2020) and Frenda et al. (2018): 0.81 for the first one, and 0.79 for the second one. This left the door open for further study in other languages than English with different models as, even though ML models performed quite acceptably there was language-specific information to be taken in consideration that could help improve the results and performances of the models.

In the case of the Evalita task, in the English dataset the results showed that BERT outperformed in almost all scores: 0.71 Accuracy, 0.70 Precision, 0.66 Recall, and 0.68 F1-Score. BERT was only outperformed by GRU without pre-trained embeddings, which achieved the highest Recall: 0.69. Finally, in the case of the Italian dataset, the best results were achieved by the SVM with linear kernel scoring 0.84 Accuracy, 0.86 F1-Score, 0.77 Precision and 0.97 Recall. The results showed that both SVM models (with RBF or linear kernel) achieved very competitive results, even showing that in the case of the Italian dataset the SVM with linear kernel achieved the highest Recall: 0.97. These results could be compared with the ones by Shushkevich & Cardiff (2019) who scored 0.70 Accuracy in the English dataset, similarly to the presented scores. Unfortunately, these authors did not report results in the Italian dataset, however it was captivating to observe that the results in the Evalita task were higher for the Italian dataset than in the English one (in contrast with the AMI performance where the English results were higher). This showed the potential that ML models still have in detection of misogyny tasks in language-specific datasets, as well as the importance of language-independent studies.

As a conclusion, the authors pointed out that in the English dataset the importance of sexist slurs and the presence of women words were the most predictive features. That confirmed the claim that sexist slurs were found to be mainly used in misogynistic instances. For Spanish the best system was a classic SVM that included features such as bags of words (1-gram to 3-grams), bags of hashtags, bags of emojis, sexist slurs and woman words presence, and the presence of words related to female genitalia, prostitution, cognitive disabilities, physical disabilities, diversity and male genitalia. As seen, the use of classic ML was beneficial in terms of transparency, in contrast to the opaquer results form the best model in the English dataset in Evalita: BERT. These results illustrated the importance of experimenting in datasets in other languages as well as the beneficial use of classical models in the tasks of detection of misogyny online.

Secondly, it was possible to appreciate the studies where NN outperformed classical models. One of the studies was Plaza-del-Arco et al. (2021b), where they used a multi-task learning approach: multiple tasks related to sexism identification were learned in parallel while using a shared representation. They used the EXIST dataset (sEXism, Identification in Social neTworks, Rodríguez-Sánchez et al. 2021), which consisted of almost 7000 tweets (in both Spanish and English). In addition, they also used other corpora for their experiments: InterTass (for polarity classification, Martínez-García et al. 2017), EmoEvent (for emotion classification, Plaza-del-Arco et al. 2020), HatEval (for hate speech identification, Basile et al. 2019) and MEX-A3T (for aggressiveness detection, Aragón et al. 2020). For their study, they used two

Transformer-based models (BERT-based model trained in English and BETO trained in Spanish, Devlin et al. 2018, Cañete et al. 2020). The challenge was divided in two subtasks: task 1 aimed to sexism identification, and task 2 aimed to sexism categorization. We paid special attention to task 1 for the objectives of our study. Their results showed Recall values of 0.78 on the EXIST test set, and 0.78 Accuracy, in comparison with the Accuracy of the winning team: 0.79. They remarked that the best results were achieved when combining sexism identification and polarity classification, and the combination of sexism identification and offensive language also showed promising results. In comparison with other studies where BETO was used, we could appreciate López-Úbeda et al. (2021) where they achieved a higher performance. One of the main differences was that while López-Úbeda et al. (2021) used BETO in a purely Spanish dataset, Plaza-del-Arco et al. (2021b) were testing in a bilingual dataset (English and Spanish), which added complexity to the detection. However, even their results were not as good as the ones from the previous study, their performance is still high enough to be competitive in a multilingual corpus.

Samghabadi et al. (2020) also used a multi-task learning approach using BERT. The model used attention mechanisms over BERT to get relative importance of words, followed by Fully Connected (FC) layers, and a final classification layer for each sub-task, which predicted the class. They used a multilingual (English, Hindi and Bengali) dataset (Bhattacharya et al. 2020) labelled as *not aggressive*, *covertly aggressive*, and *overtly aggressive* for subtask 1; and *gendered* and *non-gendered* for subtask 2. For the first task (detection of *sarcastic aggression*, *explicit aggression*, and *no aggression*) the systems performed slightly worse than in the second task (*gendered* vs. *non gendered* messages), where we found a classification that included sexism, misogyny, and other gender-targeted offensiveness. The results from the first task in Macro-F1 were 0.71 for the English and Hindi dataset, and 0.73 in the Bengali; in the second task, we could appreciate a Macro-F1 of 0.80 for Hindi, 0.85 for English, and 0.92 for Bengali. The authors commented some of the error analysis due to the balancing of the datasets, for example in Hindi most of the training messages were tagged as *non-aggressive*, while in the test set the majority of the messages were from the *aggressive* category. They concluded that the sarcastic aggressiveness was the most challenging to detect across all languages. However, we could appreciate in Founta et el. (2018) that the sarcastic performance was the best among the categories they tested. In addition, in the second task they remarked that *non-gendered* messages were easier to detect in Hindi and Bengali than in English due to more presence of examples in the training sets. That was relevant information regarding minority languages in the sense that it is highly beneficial to have further examples of toxic categories in languages other than English for further research. They finally commented that their English values were ranked in third place (out of 15 teams) with small differences of 0.0136 and 0.0159.

Mina et al. (2022) also showed that NN outperformed classical models in a fine-grained classification task. The dataset used was the EXIST dataset (Rodríguez-Sánchez et al. 2021). Thus, as it is a bilingual dataset, they used two multilingual Transformer models: one based on multilingual BERT (mBERT; Devlin et al. 2019), and one based on XLM-R (Conneau et al. 2019); and two different strategies: unsupervised pre-training with additional data, and supervised fine-tuning with additional and augmented data. The most promising results were achieved with the unsupervised pre-training strategy of the XLM-R model with additional extra data, h which they achieved 0.77 F1-Score in task 1 (binary classification), and 0.56 Macro-F1 and 0.64 Accuracy in task 2 (multi-class classification). These results were similar to the ones achieved by other studies on this same dataset and task, such as Plaza-del-Arco et al. (2021b), where they achieved 0.78 Accuracy in task 1, and 0.57 in task 2 using BETO. As it could be appreciated, in task 1 both studies reached similar results (0.77 and 0.78), but the main difference was the performance of the models in the second task, where the XLM-R model from Mina et al. (2022) went up to 0.64 in contrast to the 0.57 Accuracy from Plaza-del-Arco et al. (2021b). The differences could be due to the model, as it was proved that XLM-R improved cross-lingual language understanding (Conneau et al. 2019) and outperformed mBERT on several NLP tasks. Also, the additional external data added to the XLM-R model: external datasets or translations. This confirmed the beneficial use of domain adaptation and domain addition techniques to improve the performance of models in tasks such as misogyny and sexism detection.

Third, the case where authors used combinations of both classical models and NN. This was the case of García-Díaz et al. (2021), where they had two goals: 1) Sentiment Analysis and Social Computing technologies for detecting misogynous messages in Twitter, and 2) the compilation of the Spanish MisoCorpus-2020. The MisoCorpus-2020 (García-Díaz et al. 2021) was composed of 7682 tweets related to misogyny written in Spanish. They combined Average Word Embeddings (AWE) (Arora et al. 2017) using Spanish FastText, and linguistic features (figures of speech, pragmatics, morphological features, grammar and spelling mistakes, part of speech tagging, punctuation and symbols, twitter features, sociolinguistics, topics, sentiment lexicons, and stylometry) to understand which phenomena principally contributed to the identification of misogyny. For the classifier, they tested three ML classifiers: RF (Ho 1995), Sequential Minimal Optimization (SMO, a SVM classifier, Platt et al. 1998) and LSVM (Fan et al. 2008). The results of their models on the MisoCorpus-2020 scored up to 0.85 Accuracy (in the best of the cases) while using SMO with AWE and Linguistic Features scored lower (RF and LSVM are behind with 0.79 and 0.82 Accuracy respectively). The combination of AWE and LF outperformed the baseline model (BoW), and the models using AWE and LF separately. We could appreciate that the performance was higher in this case than in other Spanish tasks on misogyny detection, such as Plaza-del-Arco et al. (2021b) or Mina et al. (2022). However, we must note that the MisoCorpus-2020 consisted only of Spanish messages, while the EXIST corpus (the one used in the other studies) was a multilingual corpus (Spanish and English), which added complexity to the task due to the bilingual environment. We could also compare it to the performances of other studies in this same task, such as the studies from García-Díaz et al. (2021) who made a great improvement in misogyny detection in Spanish achieving an incredibly high performance. They also opened the door to further studies made with AWE. Another relevant highlight from their study was the combination of AWE with classical models. In addition, they compiled and shared a new corpus for detecting misogyny in Spanish that was a great tool to perform further experiments.

Aldana-Bobadilla et al. (2021) also proposed an ensemble technique, where they used RNN to overcome the lack of data, BERT for transfer learning, and, finally, once they determined the semantic features, they used LR because it was fast, easily understandable, and appropriate for a dichotomous dependent variable. They created a pipeline to collect, filter, tag, and generate documents' features for training a recognition model. The proposal was divided in three main stages: 1) Gathering (obtaining an appropriate set of documents for their purpose); 2) Feature Extraction (encoding information into a model able to recognize that misogyny could be a subtle type of violence); 3) Modelling (determining the semantic features and obtaining a reliable model). They tested their approach in their own test set and in the HatEval dataset (Basile et al. 2019). It was a bilingual (Spanish and English) dataset, but they only kept the Spanish tweets, and they focused on the task of hate speech detection against women. Their results showed a 0.93 Recall, 0.90 F1-Score, and 0.88 Accuracy for their test set; and 0.95 Recall, and F1-Score, and 0.93 Accuracy in the HatEval test set. Those results were outstanding in comparison with other studies that used the same dataset or other hate speech or misogyny datasets such as the performances by Pham et al. (2020) and Kovács et al. (2021). Also, in terms of misogyny detection in Spanish, the presented study achieved higher performance than the other studies presented earlier such as Pamungkas et al. (2020) or García-Díaz et al. (2021). Finally, the authors concluded with a remark on the benefit of adding extra data for covering the more sources they could find that deal with misogynistic attitudes. Their proposal was a good addition to the studies in Spanish as they added the Latin American varieties to the research. The authors themselves noted the negative performance in sentences of passive misogyny by their model, which were not used in the dataset, but they justified future work towards that path.

A similar case was studied by Ahluwalia et al. (2018) who relied upon both word N-grams and character N-grams as the semantic units for their model, and partially relied upon unclassified tweets to build an embedding layer. They used the English AMI (Fersini et al. 2018b) dataset, which was composed of 3251 tweets divided as 52% *non-misogynous* and 48% *misogynous*. For their approach, they used WE and an ensemble of 5 classifiers: LR, SVM, RF, Gradient Boosting (GB) (Friedman 2001), and Stochastic Gradient Descent (SGD) (Nemirovski et al. 2009; Zhang 2004); where the ensemble selected the class that had the highest-class probability averaged over all the individual classifiers. Their best results were scored by the BoW (unigrams and bigrams) and the Ensemble Model (ensemble of 5 classifiers: LR, SVM, RF, GB and SGD) with 0.79 Accuracy (it was a balanced dataset, where the calculated Accuracy of the baseline was 0.52) in the binary classification task. In this case, the authors pointed out the best results by the Ensemble. However, as it could be appreciated in the works form Shushkevich & Cardiff (2019) and Pamungkas et al. (2020) in the same English dataset, they achieved way higher values with SVM models (0.91 Accuracy in both cases). Those authors made use of linguistic features to help improve the detection of misogyny, which Ahluwalia et al. (2018) did not perform and was reflected in lower results. The authors concluded and left the door open to further investigation in the vector space to improve results (note that the article is from 2018, so indeed there has been improvement in such context more recent research).

Finally, the study by Guest et al. (2021) showed that both classical ML algorithms and NN performed similarly. Here they provided a hierarchical taxonomy for online misogyny (different forms of misogynistic content on Reddit) by creating a high-quality dataset and evaluating it with three different models. The corpus consisted of 6567 comments from Reddit divided into 11% *misogynistic* and 89% *non-misogynistic*. In this study, the authors evaluated the baseline (Logistic Unigram Classifier, LUC) with two uncased BERT-based models (one weighted, and the other using class weights emphasizing the minority class). The results showed that all models performed poorly on misogynistic content, with LUC scoring the highest Precision on misogyny (0.88), but very low Recall (0.07) and a very low F1-Score (0.13). In contrast, the weighted BERT model had the highest Recall (0.50) and F1-Score (0.43). In this case it was interesting to note that the preferred model for a misogyny detection task should be the weighted BERT, as it had the highest Recall (the metric to prioritize in toxicity detection contexts). Finally, the authors analysed the misclassified messages to realize their main problem was in the false positives. They found plenty of messages classified as misogynistic when, even though the messages were directed to women, they were not misogynistic or sexist at all; or referenced misogyny but they were not misogynistic themselves. Although those were not the highest performances achieved in the literature using different corpora: see for example the results from Pamungkas et al. (2020) where their Recall was above 0.80 with a ML model in misogyny detection; or in the work from López-Úbeda et al. (2021) in anorexia tasks where they achieved Recall values over 0.90 with BERT-based and other DL models.

2.1.7. Morbidity

For the case of morbidity identification in clinical notes, Dessì et al. (2020) showed that classical models outperformed DL models. The dataset used for this work was n2c2 obesity data (Henry et al. 2020) which consisted of 16 binary morbidity classes: *asthma*, *CAD* (Coronary Artery Disease), *CHF* (Congestive Heart Failure), *Depression*, *Diabetes*, *Gallstones*, *GERD* (Gastroesophageal Reflux Disease), *Gout*, *Hypercholesterolemia*, *Hypertension*, *Hypertriglyceridemia*, *OA* (Osteoarthritis), *Obesity*, *OSA* (Obstructive Sleep Apnea), *PVD* (Peripheral Vascular Disease) and *Venous Insufficiency*. Each class had around 1000 samples divided between 10-50% of positive values and 40-80% negative values. For their study, they proposed an architecture composed by a DL algorithm (LSTM) and WE: GloVe and Word2Vec (Mikolov et al. 2013a, b) and their own Word2Vec embeddings trained in the target domain. They compared the results against TF-IDF using SVM and MLP as baselines, which seemed to perform better than the DL architecture they proposed probably due to specific features that made the dataset biased in favour of traditional ML approaches. The results showed that the DL architectures (combination of WE and LSTM) had F1-Scores that oscillated between 0.56 (domain Word2Vec), 0.78 (pre-trained Word2Vec) and 0.91 (pre-trained GloVe); while the results of the ML models went between 0.97 (TF-IDF with MLP) and 0.98 (TF-IDF with SVM). The authors illustrated some conclusions from those results: first, they noted that Word2Vec performed worse than GloVe, while in the literature it was noted that GloVe enabled DL models to better recognize biased inputs. Second, they noted that although in the literature domain-specific embeddings outperformed the pre-trained ones (Dessì et al. 2017), this was not their case. They justified it explaining that their dataset was not big enough to let the model learn all the domain particularities and, thus, pre-trained embeddings (on a lot more texts) were a better option. Third, a more surprising result was that the baselines outperformed the embeddings, a reason behind that was the specific features that appeared alone for categories: in this case, the classical ML models could perform the classification with very high Precision thanks to how the feature vector was built with the TF-IDF technique. They finally concluded that their results indicated that there were specific features that made the dataset biased in favour of traditional ML approaches.

2.1.8. Offensive Language

Other related important studies were performed on offensive language. In this case, it was possible to appreciate that most of the studies made use of NN (Plaza-del-Arco et al. 2021c and Ranasinghe & Zampieri 2020), but there were also studies with classical ML algorithms such as Plaza-del-Arco et al. (2019, 2021a).

First, we could appreciate how Plaza-del-Arco et al. (2021c, 2021a) used the Spanish BERT model, BETO, in their both tasks. The dataset used was OffendEs (Plaza-del-Arco et al. 2021c) a corpus composed of 47128 Spanish comments (from Twitter, Instagram, YouTube) labelled as pre-defined categories: *Offensive, target is a person* (OFP) with 2051 comments; *Offensive, target is a group of people or collective* (OFG) with 212 comments; *Non-offensive, but with expletive language* (NOE) with 1235 comments; and *Non-offensive* (NO) with 13212 comments. The authors experimented with a multi-class classification and a binary classification, but we focused our attention to the binary classification, where they achieved a 0.78 Macro-F1 and a 0.58 Recall (offensive class). If we compare their performance with the studies we previously explored, we could observe that the metrics were in similar paths than Kovács et al. (2021) with a RoBERTa and FastText combined model in a hate speech task; and like Plaza-del-Arco et al. (2021b) with BETO for misogyny. However, we could appreciate that the performance from López-Úbeda et al. (2021), in an anorexia task, was the highest one among the studies that used

BETO. The differences between both datasets fell on the balancing, while OffendEs was highly imbalanced, the SAD corpus was balanced, which allowed a better training of both labels in a more parallel way. Finally, the authors concluded that the main characteristic of the dataset was indeed its unbalancing, which allowed stratified random sampling to best allow researchers choose what they needed for their experiments.

Another study where DL models outperformed the classical ones was the one made by Ranasinghe & Zampieri (2020). Here the authors used cross-lingual Contextual Word Embeddings (CWE) and transfer learning (inter-language and inter-task learning) to make prediction in languages with less resources. They took advantage of existing English data to project predictions in three other languages: Bengali, Hindi, and Spanish. The authors tackled both off-domain and off-task data for Bengali. They showed that not only could these methods project predictions for different languages but also for different domains (e.g., Twitter vs. Facebook) and tasks (e.g., binary vs. three-way classification). They provided important resources to the community: the code, and the English model was freely available to everyone interested in working on low-resource languages using the same methodology. For this study they used OLID (Offensive Language Identification Dataset; Zampieri et al. 2019) for English, HatEval (Basile et al. 2019) for Spanish, HASOC (Mandl et al. 2019) for Hindi, and TRAC-2 (Bhattacharya et al. 2020) for Bengali. Here they addressed the problem of data scarcity in offensive language identification by using transfer learning and a cross-lingual Transformers model (XLM-R) from a resource rich language like English to three other less-resourced languages. The results were divided by the 3 languages they studied. The best values were achieved by the XLM-R model with transfer learning (inter-task, inter-domain, and inter-language) in all cases. In particular, the Spanish model achieved 0.75 Macro-F1, the Bengali 0.85 Macro-F1, and the best value was achieved by the Hindi dataset with 0.86 Macro-F1. The authors noted that the results for Bengali needed special attention due to the use of off-domain data with respect to the English data (Facebook instead of Twitter), and it contained three labels instead of two. This performance showed similarity in comparison with the study from Mina et al. (2022), in the Spanish EXIST dataset for sexism detection. However, in terms of minority languages, such as Bengali and Hindi, we could observe that the proposed performance achieved by Samghabadi et al. (2020) in the task of misogyny detection competed with the present study for the first language, but Ranasinghe & Zampieri (2020) perform better in Hindi in comparison with the later study. As a final remark, it was interesting to note the possibility of transfer learning on off-domain data and off-task data in multilingual context to perform offensive language detection, which even opened further possibilities to extend it to other forms of toxicity language.

In the case of the studies where classical models worked better than DL models, we could appreciate the proposals by Plaza-del-Arco et al. (2019, 2021a). In the first case, the dataset used was the task's OffensEval dataset divided as 8840 *not offensive* tweets and 4400 *offensive* tweets. In the second case they used the OffendEs (Plaza-del-Arco et al. 2021c) Spanish dataset, and the OffendMEX (Plaza-del-Arco et al. 2021a) for the Mexican variety of Spanish.

In their first approximation, the authors proposed the integration of lexical features from a polarity lexicon and an offensive/profane word list in the classification using a LSVM algorithm, together with the Term Frequency (TF) considering unigrams. The lexical features were obtained with two lexicons: VaderSentiment[15] and Offensive/Profane Word List[16]. The results showed that for the *not offensive* class they achieved 0.88 F1-Score, and 0.56 for the *offensive* class; in addition, they achieved 0.44 Recall and 0.79 Macro-F1 in the case of the *offensive* class. That could be due to the unbalancing of the data: 67% of the data are considered *not offensive* in comparison with 33% *offensive* tweets.

In their second study they were identifying offensive language targeting the Mexican variant of Spanish with a contextual and non-contextual binary classification. Two approaches were taken: a Bi-GRU neural network for the non-Contextual binary classification, and a XGBoost + BETO ensemble for the Contextual binary classification. In addition, they evaluated a bag-of-unigrams-bigrams-trigrams and a LSVM classifier as baselines. They could appreciate that all DL models outperformed the baseline-SVM model, but the baseline-BOW

[15] https://www.nltk.org/_modules/nltk/sentiment/vader.html

[16] https://www.cs.cmu.edu/~biglou/resources/

outperformed the Bi-GRU model. They finally concluded that the sole inclusion of the features was not enough to improve the performance.

The results from both studies showed that the best value was achieved by the LSVM model from Plaza-del-Arco et al. (2019) in the OffensEval dataset; however, it was interesting to observe the results from the other models: in the case of subtask 1 and 3 (non-contextual multiclass classification in Spanish, and binary classification in Mexican Spanish respectively), the best results were achieved by Transformer-based models. In the first case, an XLM-RoBERTa that scored 0.73 Macro-F1, and in the second case, a BETO model that scored 0.70. In subtask 2 (contextual multiclass classification for Spanish), a MLP model scored the highest with 0.73 Macro-F1. Finally, in subtask 4 (contextual binary classification in Mexican Spanish), the best value was achieved by the baseline-DL, which was a combination of XGBoost and BETO and scored a 0.68 Macro-F1. Consequently, even if the performances were lower in the case of Plaza-del-Arco et al. (2021a), the tasks were also more complex, being multiclass classification or in the Mexican variety of Spanish. Note that the multiclass classification results from Plaza-del-Arco et al. (2021c) with BETO reached 0.64 Macro-F1, which was a lower result in comparison with the 0.73 Macro-F1 scored now (Plaza-del-Arco et al. 2021a) in the same dataset: OffendEs, This task allowed advance in the study of offensive language identification in Spanish in a more complex matter, multiclass classification, in addition to introducing studies in other varieties of Spanish, such as Mexican Spanish. This last remark illustrated the importance of dialectological and even sociolinguistic features that should be introduced in the tasks of detection of offensive language and other forms of toxicity online.

2.1.9. Sentiment Analysis

In the case of sentiment analysis, it was possible to observe the study by Cumalat Puig (2020) who aimed to detect if a text contained several types of abusive behaviour. The main goals to achieve in this project were: 1) Reproduce previous baseline results, 2) Incorporate Catalan short texts to the experiments, 3) Improve the previous baseline system, 4) Study and test several ways to convert the texts to embeddings to further use them for classification tasks, and 5) Study and test different pipelines (combinations of pre-processing and classifiers). The database used included around 200000 short texts in Catalan and Spanish divided into 7 categories: *aggression*, *violence*, *anxiety*, *depression*, *distress*, *sex*, and *substance*. In this study he explored vectorization (TF-IDF, Doc2Vec) and classification techniques (RF, SVM, BERT) for a short text classification task in Catalan and Spanish with very informal language on abusive topics. His results showed that the best model was a multilingual version of BERT with his proposed robust pre-processing without stemming, which achieved 0.75 general Accuracy. However, he also showed similar results (surpassing BERT in one of the categories) and with a much faster computing time by using BERT for extracting embeddings and then classifying them using a SVM model. The author interestingly showed the progressive improvement that went from TF-IDF models (with Accuracy around 0.40 in combination with RF, and around 0.70 in combination with SVM), Doc2Vec in combination with a SVM (that scored 0.69), BERT (0.749), and BERT combined with SVM models (0.748). He also remarked that the BERT-SVM, that used BERT tokens as input to a SVM classifier, was an interesting option if there was a need of fast results, as the differences between this model and the fine-tuned BERT in terms of results were not that remarkable. This was the only work studied on Sentiment Analysis, but we thought it was an interesting remark as it showed a great variety of model combinations and how the results increased with the most advanced models. Also, it provided an extensive study in Spanish and in another minority language, Catalan. We must not forget the existence and importance of minority languages in these tasks, that was why this well performed study was worth considering.

2.1.10. Suicide

In the case of studies about suicide detection, the proposals were quite equally divided between ML (O'Dea et al. 2015, Ramírez-Cifuentes et al. 2020, Ryu et al. 2019) and DL (Astoveza et al. 2018, Tadesse et al. 2020, Ophir et al. 2020), even some showed similar results for both (Ji et al. 2020).

First, in the studies where classical models showed better results, it was possible to appreciate the one from O’Dea et al. (2015) where they aimed to establish the feasibility of consistently detecting the level of concern (‘strongly concerning’, ‘possibly concerning’ and ‘safe to ignore’) for individuals’ Twitter tweets, which made direct or indirect textual or audio-visual references to suicidality. Using a set of instructions and categories, human coders aimed to do this using only the content of the tweet itself. Following this process, in this study they designed and implemented an automated classifier that could replicate the Accuracy of the human coders. The feasibility of this automated prediction was to be examined using Recall and Precision metrics. The study demonstrated that was possible to distinguish the level of concern among suicide-related tweets, using both human coders and an automatic machine classifier. Plus, the findings confirmed that Twitter is used by individuals to express suicidality and that such posts evoked a level of concern that warranted further investigation. In this study, they used two different data sets: Set A and Set B, and the combination of both. Set A consisted of 830 tweets divided as 152 *strongly concerning* (18%), 456 *possibly concerning* (55%) and 222 *safe to ignore* (27%); and Set B consisted of 991 tweets divided as 106 *strongly concerning* (11%), 574 *possibly concerning* (58%) and 312 *safe to ignore* (31%). Here two ML algorithms were used: SVM and LR; being the SVM with TF-IDF no-filter the best performing algorithm, which showed a gain in performance Accuracy when the two sets were combined, with an overall Accuracy value of 0.76. In the case of *strongly concerning* tweets, the best results were achieved in Set A, with 0.64 Recall and 0.74 F1-Score; in Set B the results were lower, which also made the results of the combined dataset worse. On the other hand, in the case of *possibly concerning* tweets, we could appreciate that the highest results were achieved in Set A (0.97 Recall), however the best F1-Score (0.83) was reached when the two sets were combined. Note that most of the tweets were from the *possibly concerning* category, and as we could appreciate the results for the *safe to ignore* category were low. Also, the main difference between Set A and B was the more presence of *strongly concerning* tweets in set A. The authors remarked the difficulty in the agreement of the human coders when labelling this data, in addition to the sensitive and ethical concerns, which were difficult to navigate. However, they highlighted the importance of these kind of tasks. Suicide detection could be highly related to the tasks of anorexia and anxiety detection, as they are not toxic content, but we would consider it concerning behaviour that needs to be addressed as well. In previous tasks we could appreciate how different approaches reached 0.93-0.94 Recall (López-Úbeda et al. 2019, 2021) in anorexia detection; and 0.70-0.90 (ML) and 0.90 (DL) Recall in tasks of anxiety detection (Shen & Rudzicz 2017), which entailed high levels of risk behaviour detection on social media, especially in cases based on DL models.

Next, in the study by Ramírez-Cifuentes et al. (2020), they explored behavioural, relational, and multimodal data extracted from multiple social platforms, and developed ML models to detect users at risk. They aimed to the identification of significant statistical differences between the textual and behavioural attributes of the control groups (a suicide-related vocabulary group, and a general group) compared with the suicidal ideation risk group. The dataset consisted of 1200 tweets divided into 74% *control* cases, 10% as *suicidal ideation risk* cases and a remaining 16% fell into a *doubtful* category. They experimented with 3 classical models (RF, LR and SVM) and a DL one (CNN). The combination of textual, visual, relational, and behavioural data outperformed the Accuracy of using each modality separately. However, the text-based baseline models (BOW and WE) outperformed the proposed models. The best results were achieved by the SVM model with 0.91 Precision, 0.77 Recall, 0.83 F1-Score and 0.84 Accuracy in task 1 (where they were using a focused group with suicide-related vocabulary). In task 2 (with a generic group that may or may not use suicide-related words) the results were a bit worse, but SVM was still the best classifier with 0.83 Precision, 0.82 F1-Score and 0.82 Accuracy; however, the Recall improved increasing to 0.80. This performance could be compared with the previous study on suicide: there, we saw an extremely high Recall for *possibly concerning* tweets in set A, however, we could appreciate that the dataset was imbalanced towards that class and that the *safe to ignore* class was performing poorly. Aside from that, we could appreciate that both studies performed similarity in suicide detection. Finally, the authors concluded that their findings were more positive when the control group was used, as that meant that the classifier was able to distinguish users in the risk group from the control cases. Also, as they were selecting some extra features for the models, they pointed out the importance of their interpretability as those elements that could be understood and used by the clinical professionals, being the identification of textual and behavioural elements the most important one. The addition of image-based features improved the results in comparison with the purely text-based features as well.

Finally, the study by Ryu et al. (2019) aimed to develop ML models to predict suicide behaviours through a step wise approach, from low to high risk. They worked on an application of ML algorithms to public health data and identification of individuals experiencing suicide ideation among the general population. They developed models to predict which individuals had a history of recent suicide attempts, and thus an increased suicide risk, among those who have experienced suicide ideation. The data was obtained from the Korea National Health and Nutrition Examination Survey (KNHANES, Kweon et al. 2014) and corresponded to 1324 *suicide attempters* and 1330 *non-suicide attempters*; finally, a 70% of the data was assigned to the training set. They efficiently screened individuals at high risk for suicide in the general population. The results showed 0.78-0.82 Accuracy values for the RF algorithm, outperforming DL models. They also performed a resampling (Synthetic Minority Over-sampling TEchnique, SMOTE) to balance the data, a recursive feature elimination, a 10-fold cross validation to avoid overfitting, and increased the generalization of the model. Features for physical health (days of feeling sick or in discomfort, days of walking per week), substance use (AUDIT score, amount of daily smoking), and socioeconomic status (average work week, household composition) played an important role in classifying suicide attempters and suicide ideators. Specially, features such as depressed mood, stress level, and quality of life were of greater importance. The results showed 0.95 ROC AUC, 0.89 Accuracy, and over 0.87 F1-Score. This results, especially the ones after using the SMOTE technique, showed improvement over the previous performances on suicide detection (O’Dea et al. 2015, Ramírez-Cifuentes et al. 2020). This could be due to a better prepared dataset that contained more data, and it was properly balanced, in addition to the use of features that improved the performance of the models (which was shown previously in Ramírez-Cifuentes et al. 2020 as well). Finally, the authors concluded positively towards ML algorithms in tasks of suicide detection, as it could be seen already in the three studies presented so far. However, further study is needed in this concerning and difficult topic.

In the case of DL models outperforming the classical ones, it was possible to observe how Astoveza et al. (2018) used an Artificial Neural Network (ANN), a MLP classifier. For this study they gathered 5174 tweets, which 3055 were English and 2119 were Filipino or Taglish, labelled binary as *risky* or *non-risky*. Their best results were achieved with a Learning rate Adaptive, and a Learning rate initialization of 0.001, where they scored 0.78 Accuracy with the text feature set, and 0.72 with the emoji feature set. The authors noted that it should be considered that the feature set with text performed more stably compared to the feature set with emojis. The DL model from this study performed similarly than the observed previously (using a ML model). The present model is even outperformed by a SVM (Ramírez-Cifuentes et al. 2020) and RF with SMOTE (Ryu et al. 2018). That poor performance could be due to the use of many features by the earlier authors, while the present study only considered text or emojis. These performances proved the importance of external features in addition to the only text messages in suicide detection. However, a positive point from the present study was the use of other minority languages, such as Filipino and Taglish, which added complexity to the task, as there were fewer resources for languages other than English. So, in this sense, their study added light and had to be considered a relevant work to the detection of suicidal behaviour tasks in the context of language inclusion.

Next, Tadesse et al. (2020) compared the strengths and potential of CNN and LSTM techniques and four traditional classifiers (SVM, NB, RF and XGBoost). In this study they used a Reddit dataset (Ji et al. 2018) divided as 3549 *suicide-indicative* posts and 3652 *non-suicidal* posts. They combined NN architectures with WE to achieve the best relevant classification results. They contributed to an N-gram analysis, classical features analysis (BOW, TF-IDF) and statistical feature performance (WE), and finally a comparative evaluation. Their results showed that the LSTM-CNN hybrid model, in combination to the WE from Word2Vec, considerably improved the Accuracy of the classification (as it combined the strengths of both LSTM and CNN algorithms) achieving 0.93

F1-Score, and 0.94 Accuracy and Recall. This performance was outstanding and outperformed all previous performances from the rest of the studies on suicide detection. Specially, the Recall was remarkable as the highest value -and the most important one in this kind of tasks. The authors justified the reason of this performance as the combination of both DL models compensated their individual shortcomings. The model took advantage of the LSTM to maintain context information in a long text, while the CNN extracted the patterns using richer representations of the original input text and by being able to analyse words and their combinations. Another relevant factor, previously observed as well in Ramírez-Cifuentes et al. (2020) and Ryu et al. (2019), was the use of extra features, such as frustration, hopelessness, negativity, and loneliness.

Third, Ophir et al. (2020) used CWE as input for two ANN models: Single Task Model (STM) and Multi-task Model (MTM), which included hierarchical, multi-layered risks factors (texts, personality traits, psychosocial risks, psychiatric disorders) in the task of detection of suicide risk from textual Facebook posts. The dataset used included 83292 posts generated by 1002 Facebook users of which 36% were considered *general risk of suicide* and 13% were considered *high risk of suicide*. It could be appreciated that their results showed that the MTM model improved the Accuracy substantially in contrast to the STM model (which scored 0.65 in the *general* group and 0.66 in the *high suicidal risk*). Also, the performance of the MTM model was like the one from Astoveza et al. (2018), where they also used an ANN model. The authors finally concluded that the use of MTM (which integrated theory-driven risk factors) produced improvement in the prediction of suicide risk from textual posts in social media compared with the STM and other models. The reason was that the ANN models allowed to “discover” suicide-related content even if the authors of the post did not explicitly share it. For further research they suggested the inclusion of image-related features to the posts, which was proven beneficial in Ramírez-Cifuentes et al. (2020). However, as we could appreciate in Tadesse et al. (2020), there were other powerful methods that improved the performance.

Finally, Ji et al. (2020) showed results for both classical and DL models with similar results. In this study they presented several datasets from different sources: Reddit (Ji et al. 2018), Twitter (Coppersmith et al. 2015, Vioulès et al.

2018), ReachOut Forum[17] (Milne et al. 2016), EHR[18] data from the California Emergency Department[19], and regarding mental disorders (resources for mental health disorders without effective treatment that can turn into suicidal ideation), the eRisk dataset for Early Detection of Signs of Depression (Losada et al. 2016) or the Reddit Self-reported Depression Diagnosis (RSDD, Yates et al. 2017). In the case of ML models, they paid special attention to textual context analysis (lexicon-based filtering and word cloud visualization) and feature engineering (tabular, textual, and affective features), and for the DL-based representation learning they used CNN- and LSTM-based text encoders. They finally concluded that boosted manually feature engineering techniques and DNN-based models showed the best result. Although the authors did not report numeric results, they highlighted the limitations of this kind of tasks: data efficiency, where there was an important need of annotated and balanced datasets; the presence of annotation bias due to the manual labelling; and the lack of suicidal intention understanding, as the models focused on features, but the intention of suicide remained complex even from the psychological perspective, which lacks in these methods.

As it could be appreciated along the section, one of the main problems of suicide detection tasks was the inconsistency of publicly available datasets. Each study had to compile its own dataset to study suicide detection, which were small, imbalanced, biased, or showed different contextualities in the messages. This also affected us with the inability to properly compare results, as the datasets included different kind of data, labels, and balancing (even though we could still compare performances). Also, each study considered different external data which added complexity to the task. This was a big problem in the tasks of suicide detection as it slowed down the possibility of experimenting and limited the available resources. As a final remark, Ji et al. (2020) interestingly numbered a series of available datasets for suicide detection which we considered strong and valuable knowledge for further research in the area; in addition to all the provided datasets from each study in this the section.

[17] A peer support platform provided by an Australian mental health care organization.

[18] Demographical information, admissions, diagnostic reports, and physician notes.

[19] https://data.chhs.ca.gov/group/resources

2.1.11. Toxicity

If the attention was set to tasks of toxicity detection, it was possible to appreciate that most of the latest references used DL techniques to solve their objectives (Koratana & Hu 2018, Pappie 2019, Paraschiv 2020, Pavlopoulos et al. 2020, Taulé et al. 2021, Warholm 2021, Gharbi et al. 2021), while only a few of them used ML solutions (Ožegović & Celin 2020). There was also a case where they combined both ML and DL algorithms (Van Aken et al. 2018). Finally, there were other approaches like the use of adversary examples to test the errors of the models. In particular, Google's Perspective system, where Hosseini et al. (2017) showed how harmful those examples were for toxic detector systems.

In the case of the DL models, we could appreciate that most of the authors used BERT (or language-adapted BERT systems). That was the case of Paraschiv (2020) where he used an adapted BERT model to improve fake news, propaganda, and offensive tweets detection for English and German. For English, he used the SIMAH (SocIal Media And Harassment, Cellier 2019) dataset, which contained 6374 tweets divided into 4 categories: 2713 *Harassment*, 55 *Indirect Harassment*, 76 *Physical Harassment*, and 2582 *Sexual Harassment*. For German, the Germeval[20] dataset was used, which contained 33% *offensive* tweets (consisting of 2% *profane*, 13% *insult* and 18% *abuse*) and 67% labelled as *other*. His results showed that his model outperformed the standard release of BERT with a relative high margin. For the task of offensive language classification, in the case of the English dataset, most of the highest results were achieved by BERT-P (English), which scored 0.81 Accuracy, 0.69 Recall, and 0.75 F1-Score. This model was just outperformed in terms of Accuracy by the GRU+Attention model, that scored 0.85 Accuracy. In the case of the German dataset, the best results were achieved by the BERT-H (German), which reached 0.76 Accuracy, and 0.77 Recall and F1-Score. The author finally concluded that the use of BERT models was the best choice, in addition, the addition of "specialization" pre-training boosted the performance in each task. The only drawback that the author remarked was the computationally expensiveness of the BERT models, for that, he proposed future work on

[20] https://germeval.github.io/

distilled versions of BERT. These results showed improvement in the performance of detection of offensive language tasks in comparison with previously presented studies, even if it were small steps. That was the case of the works from Pham et al. (2020) for hate speech detection in Vietnamese using RoBERTa + FastText. Other studies in Spanish showed similar (or lower) performances using BETO, such as Plaza-del-Arco et al. (2021c) for hate speech or Plaza-del-Arco et al. (2021b) for misogyny detection. As it could be appreciated, adding language-specific data in the model improved the performance, as well as adding variety and valuable information to the tasks. However, it could also be observed that ML models performed outstandingly in similar tasks: Davidson et al. (2017) in the offensive task with an SVM, or Pamungkas et al. (2020) with a SVM in a trilingual dataset (English, Spanish and Italian) in misogyny detection tasks. All models in those studies scored 0.80-0.90 Recall, which was the most relevant metric in toxicity-related tasks. Those values illustrated the positive general performance of every system in their tasks. In addition, these studies shared the addition of extra features to support the classification processes, which ended up a beneficial aspect to consider.

Taulé et al. (2021) proposed the DETOXIS (DEtection of TOXicity in comments In Spanish) task and created the NewsCom-TOX corpus (Taulé et al. 2021), which consisted of 4359 comments in response to different articles extracted from Spanish online newspapers and discussions from forums (on average, 31% of the comments were *toxic*). The results showed that BETO outperformed the multilingual models by far as well as how those models outperform ML models (TF-IDF with RF, SVM or LR). The best system was on some occasions outperformed in terms of Recall by other approaches, but at the cost of a significant Precision lost. This model scored around 0.67 Recall, while the Recall of the best model was around 0.75 in a binary classification. The authors pointed out the importance of fine-tuning on similar tasks related to sentiment and emotion analysis, that increased the training data and made more precise predictions. In addition, this was the only team in the competition that used the provided extra features (argumentation, constructiveness, stance, target, stereotype, sarcasm, mockery, insult, improper language, aggressiveness, and intolerance). This performance could be compared with other tasks and studies where BETO was used, such as López-Úbeda et al. (2021) in the task of anorexia detection, as well as Plaza-del-Arco et al. (2021b) in misogyny detection. Note that these models outperformed the current one in Recall; however, the present dataset contained comments in which subtle hidden toxic messages were included, thus the added difficulty in the task. Overall, the studies showed all the possibility for improvement left in these tasks. Even so, they accentuated the importance of different strategies to build upon or explore further in the studies on hate speech and toxicity detection.

Warholm (2021) fine-tuned a BERT model with a Norwegian dataset partly gathered by the author and partly taking comments from a dataset by Jensen (2020), which showed improvement in the results in Norwegian hate speech detection in contrast to NB-BERT, a model trained on English data used to detect Norwegian toxicity. The dataset consisted of 7078 comments where 18% were *offensive*. In his study he captured pragmatic devices, such as politeness, to detect linguistic cues that predicted a conversation's future health. His system reached levels of 0.75-0.80 ROC AUC in the Norwegian dataset. However, his best results were accomplished when the NB-BERT model learned from English datasets (with 0.90 ROC AUC) in the sentiment analysis task, which did not perform as well in the case of offensive language detection. Finally, the author pointed out the added difficulty of the few existences of Norwegian datasets of unhealthy comments online, thus, the need for more data collection in such language, as more examples would mean better results and better performance overall (as in Price et al. 2020).

In the case of Pappie (2019), he also used two types of RNN: a LSTM and a GRU; which were compared against a LR baseline model. The dataset used was the one from the "Quora Insincere Questions Classification[21]" challenge on Kaggle, which contained over 1 million rows, and it was labelled as either *sincere* and *insincere* (the dataset was highly imbalanced as only 6% of the samples are from the *insincere* class). The best results were achieved by the LSTM with 0.95 Accuracy, 0.70 Recall, and 0.67 F1-Score; but note that both the LSTM and GRU outperformed the baseline model. The Accuracy

[21] https://www.kaggle.com/c/quora-insincere-questions-classification

performance was similar to the one achieved by Tadesse et al. (2020), who was using a LSTM-CNN + Word2Vec model in the suicide detection task. However, the Recall is deeply outperformed in that study (as well as the F1-Score), and by López-Úbeda et al. (2021) with a LSTM model in the anorexia detection task. The difference in performances was due to the differences in the task and dataset, as the detection of toxicity englobed much more types of messages than detecting suicide or anorexia. That fact added difficulty to the toxicity detection task, as the term did not have a closed meaning and it was more subjective and could present higher variance. In addition, as we could appreciate, the dataset was highly imbalanced towards the *healthy* class, thus, the Accuracy was biased towards that class, which justified the incredible high Accuracy, but more average Recall and F1-Score values. In contrast, Tadesse et al. (2020) and López-Úbeda et al. (2021) used balanced datasets. These results remarked the importance to consider a balanced dataset in the studies, as we could appreciate the strong variance in performances. Finally, the authors also remarked the incapability of performing cross validation, or not being able to search for the optimal hyperparameters, which could possibly improve the final performance and results. The different values obtained by the various systems indicated that these kinds of approximations would achieve good performances with the kind of represented messages in those collections.

Next, Pavlopoulos et al. (2020) were motivated to check if the context can either amplify or mitigate the perceived toxicity of posts. For that they developed DL classifiers, both context-insensitive and context sensitive. They used 10 different datasets: CCTK and CWTK (Pavlopoulos et al. 2020) for toxicity, Davidson et al. (2017) for hate and offense, Zampieri et al. (2019) for offense, Waseem & Hovy (2016) for sexism and racism, and Gao & Huang (2017) for hate (all in English); Wiegand et al. (2018) for German insult, abuse, and profanity; Ross et al. (2017) for hate in German; Pavlopoulos et al. (2017) for rejection in Greek; and Mubarak et al. (2017) for obscene and offense in Arabic. The best results were achieved by PERSPECTIVE (context insensitive CNN-based model) and BERT-CCTK (context insensitive, fine-tuned BERT of the CCTK dataset) and their context-aware variants (CA-CONC-PERSPECTIVE and CA-CONC-BERT-CCTK) on much larger datasets. Of the four models, the lowest, BERT-CCTK, scored 0.78 ROC AUC, PERSPECTIVE scored 0.79, finally CA-CONC-BERT-CCTK and CA-CONC-PERSPECTIVE scored 0.816 and 0.818 respectively. The authors concluded illustrating how the context could both amplify (3.6%) and mitigate (1.6%) the perceived toxicity, even though they claimed that there was no significant evidence that context improved the performance of the classifiers (likely related to the small number of context-sensitive comments).

Some authors also approach other NN, such as RNN, CNN or LSTM. For example, Koratana & Hu (2018) explored various approaches of either classical (LR) and CNN and RNN based models. They remarked that if a LR or SVM model was fast and sufficiently confident enough in its prediction, it was used; otherwise, a DL model was chosen. The dataset used was the one from the "Toxic Comment Classification Challenge" on Kaggle, which included 223549 comments annotated as one of the following categories: *toxic*, *severe toxic*, *obscene*, *insult*, *threat*, *identity hate,* and *clean*; most of the samples (about 200000) were labelled as *clean*. For the DL architecture, they took two approaches: GRU RNN with attention (used specific modifications), and Very Deep Convolutional Neural Network (VDCNN, motivated by the success of Very Deep Networks in computer vision). The GRU/LSTM took as input WE, and its output was a sentence embedding, which was fed through the attention layer and then fed again into a fully connected classifier. Their results showed that the best Accuracy (0.99) was achieved by the Bi-LSTM with attention and pretrained embeddings, and the best F1-Score (0.66) was achieved by both the LSTM and GRU with attention and FastText embeddings, in contrast to the 0.44 F1-Score and 0.97 Accuracy from the LR baseline. Their performance was like the one by Pappie (2019) who approached a toxicity detection task with a LSTM, both with imbalanced datasets. However, the models were outperformed by Tadesse et al. (2020) for suicide detection with a LSTM-CNN + Word2Vec model with a balanced dataset. Once again, showing the importance of having balanced datasets in this kind of toxicity detection tasks. Finally, the authors remarked that FastText pretrained embeddings offered a significant improvement for two reasons: embeddings could calculate sub-word embeddings, and they were trained on extremely large corpus.

Another study that explored both DL and ML models is the one performed by Gharbi et al. (2021). In their study they explored the vocabulary (Tunisian) of the dataset through feature engineering approaches and performed classifications of ML models (NB and SVM) and DL models (ARBERT, MARBERT and XLM-R). Compared to other Arabic dialects which are mostly based on Modern Standard Arabic (MSA), the Tunisian dialect is a combination of many other languages like MSA, Tamazight, Italian and French. Because of its linguistic richness, dealing with NLP problems could be challenging due to the lack of large, annotated datasets. In this paper they introduced a new annotated dataset composed of approximately 10000 comments: more than 6000 *abusive* comments, 3000 *hate* comments, and less than 1000 *normal* comments. They provided an in-depth exploration of its vocabulary through feature engineering approaches (removal of stop words, N-gram scheme, reduce the feature size and balancing the data) as well as the results of the classification performance of ML and DL classifiers. For their study they took into consideration single words (unigram), sequences of two successive words (uni- + bigrams), as well as expressions of three successive words (uni- + bi- + trigrams). Their results showed that for the binary classification (*abusive*/*hate* vs *normal*), most of the F1-Scores obtained by the classical ML classifiers were above 0.90, so the classical models outperformed the DL ones in this case. The best results for the binary classification were uni- + bigrams and TF of > 2 with 0.92 F1-Score for NB; and for the 3-way classification 0.84 F1-Score for the SVM. They also wanted to note that LR predicted more efficiently with uni- + bigrams, and both SVM and RF kept the same performance whatever the N-gram scheme was. In the case of the monolingual Transformers, they remarked that MARBERT (0.78 F1-Score for binary, and 0.66 for 3-way classification) outperformed ARBERT (0.74 F1-Score for binary, and 0.64 for 3-way), which confirmed that models trained on Arabic dialects performed better. Finally, XLM-R showed outstanding results, with 0.85 F1-Score in binary and 0.75 in 3-way classification, outperforming the monolingual Transformers. As it could be appreciated, toxicity had a big meaning, which sometimes made the task of detecting it challenging. In addition, we could observe different subtasks as well, such as binary classification or multicategory classification. In that case, the authors presented a new Tunisian dataset for the task of toxicity detection and performed extra binary, 3-way, and 7-way classification tasks. In the case of binary, the best results were achieved by the XLM-R model with 0.85 F1-Score, which outperformed the BERT-based models. Other studies that evaluated XLM-R were Mina et al. (2022) with similar (slightly lower) performance in binary classification in misogyny detection in English and Spanish. With similar performances we could also appreciate Ranasinghe & Zampieri (2020) in offensive language in Spanish, Bengali and Hindi. As we could observe, it seemed that XLM-R worked better in minoritarian languages than in languages with more resources like Spanish or English, as the performances in Tunisian, Bengali and Hindi improved. In multiclass classification, we could observe that the present model for Tunisian improved the performance as well in contrast to previous models (Mina et al. 2022). Here, the authors remarked the added difficulty in discriminating between different types of toxicity rather than between toxic or not toxic, however they achieved the higher results in a 3-way classification with classical ML models (0.90 Macro-F1).

Another study where they explored the performance of ML models was made by Ožegović & Celin (2020). Here their goal was to limit toxic comments written by the users and flag them as inappropriate. This project was focused on ML models to identify toxicity in online conversations and flag them as *rude, disrespectful*, or *likely to make someone leave discussion*. If these comments could be identified that would lead to safe and more collaborative threads. The main challenge was to build multilingual models for toxicity classification with English-only training data ("Toxic Comment Classification Challenge", Kaggle). Their approach was to downgrade the dataset, use a classifier of their choice, and proceed to feature extraction (comment length, punctuation count, uppercase words, count of bad words, sentiment, polarity, and subjectivity). As a conclusion, the authors showed that using RF they achieved 0.9459 ROC AUC, while the total best value was 0.9536.

As we could appreciate earlier, ROC AUC values are often quite high. For example, Founta (2018) achieved 0.96 ROC AUC values (cyberbullying and offensive language), 0.92 (hate speech), and 0.98 (sarcasm detection); as well as the 0.90 from Warholm (2021). These values did not give us a detailed explanation of the true performances of the models as good as Precision, Recall or F1-Score did. The reason might fall into that Accuracy values in unbalanced datasets greatly increased the values of ROC AUCs. However, saying that a model performs 0.98 ROC AUC seemed insufficient information to understand completely the model. Values of Precision or Recall (mostly Recall in toxicity detection tasks), are more valuable metric results to observe due to the objectives of such tasks. With this claim, we could understand that if ROC AUC correctly illustrated the models' performances, the problem would be almost solved. As say, if the *real world* was correctly represented in the collections, which was not the case because datasets presented different understandings of what was toxic, or they were highly imbalanced, which illustrated the existing bias in the datasets. Even with this consideration, we could appreciate that this task's performance was a more "medium score", given the fact that in this challenge the main goal was to train a classifier with English-only data and test it in a multilingual set. In addition, the results from the present task were achieved by a classic ML model, in contrast to the ones from Warholm (2021), who used BERT; but, in this case, they were accompanied by extra features from the dataset and the feature extraction performed by the authors.

Finally, Van Aken et al. (2018) performed a comparison of different DL (RNN, CNN) and ML (LR) approaches, and proposed an ensemble (with gradient boosting decision trees) that outperformed all individual models. They used different datasets in fields such as hate speech, racism/sexism, or harassment, and from different domains such as Wikipedia ("Toxic Comment Classification Challenge" dataset from Kaggle) and Twitter (Hate Speech dataset, Davidson et al. 2017). With their study they showed that the ensemble learned to choose an optimal combination of classifiers based on a set of comment features. Because the classifiers had different strengths and weaknesses, they expected the ensemble to outperform each individual classifier. The ensemble improved in Macro-F1 (especially on sparse classes and data with high variance). They also used 2 different pre-trained WE to compensate idiosyncratic and misspelled words, however the results did not lead to strongly different predictions. We could appreciate that the best results were achieved by the ensemble model with 0.88 Recall and 0.79 F1-Score for the Kaggle dataset; and 0.83 Recall and 0.79 F1-Score for the Twitter dataset. In addition, we could also appreciate that the best individual model was the Bi-GRU with attention and with pre-trained word embeddings (FastText): 0.87 Recall and 0.78 F1-Score in the Kaggle dataset; and 0.83 Recall and 0.79 F1-Score in the Hate Speech dataset. These performances could be compared to the ones obtained by López-Úbeda et al. (2021) in the anorexia task with a Bi-LSTM model, which outperformed the present results; or the GRU with attention from Paraschiv (2020) which was outperformer by Van Aken et al. (2018). The results by Koratana & Hu (2018) also showed interesting findings with their Bi-LSTM with attention and FastText embeddings achieving 0.66 F1-Score in the same Kaggle dataset, which is outperformed by Van Aken et al. (2018) as well. In the case of the Hate Speech dataset, we could appreciate that Founta et al. (2018) outperformed the former results in the case of Recall (0.89) but they were outperformed in terms of ROC AUC: 0.95 by Van Aken et al. (2018), in contrast with the 0.92 from Founta et al. (2018). The authors finally concluded that combining their shallow learner approach with NN was highly effective, but that the different WE used did not lead to remarkable different predictions: that was beneficial where there was high variance within the data and on classes with fewer examples. Also, as a final note, they highlighted that word and character N-grams learned by the LR classifier produced strong predictions that could be combined for increasing Accuracy. In addition, most of the error analysis showed that the misclassifications were made due to missing training data or to texts with highly idiosyncratic or rare vocabulary.

2.2. Conclusions

As it could be appreciated, several studies performed many different approaches to the problem of detection concerning/toxic behaviour online. Some of them utilized DL or Transformer-based techniques, which gained popularity recently. However, as some authors remarked, the results those models return could be more complicated to interpret than the ones from classical models (MacAvney et al. 2019, Kovács et al. 2021 for hate speech; García-Díaz et al. 2021 for misogyny). Also, the simpler, faster performance and application of classical models made them still be used by some authors, which, even though they tested

DL models as well, they decided to continue with the classical ones if their results were confident enough (Koratana & Hu 2018 for toxicity detection), they were not too far from the results of the DL models (Shen & Rudzicz 2017 for anxiety studies; Plaza-del-Arco et al. 2021a for offensive language) or they could even contribute helping the DL models (Van Aken et al. 2018 in toxicity detection). The studies that reached good results with classical algorithms were working with domain and task adaptation techniques, meaning that the authors used several or a combination of datasets from different domains (Twitter, Wikipedia, YouTube, etc.) and from different tasks (binary classification, 3-way classification, cross-categorical, etc.). In addition, the use of cross-lingual data also seemed to result in improvements, for example, adding Spanish (Frenda et al. 2018, Plaza-del-Arco et al. 2021b) or Italian (Shushkevich & Cardiff 2019, Pamungkas et al. 2020) data to the training set or even training in one language (mainly English) and testing on other languages, such as Spanish, Bengali, or Hindi (Samghabadi et al. 2020, Ranasinghe & Zampieri 2020).

From the studies performed on the different kind of concerning behaviours online, it could be appreciated that, from the classical models, the one that seemed to work best was SVM. This model was used in combination to TF-IDF and achieved the highest results in anorexia (López-Úbeda et al. 2019), in hate speech (Davidson et al. 2017), in morbidity (Dessì et al. 2020), misogyny (Pamungkas et al. 2020, Plaza-del-Arco et al. 2019) and suicide (Ramírez-Cifuentes et al. 2020). There were also studies where the N-gram scheme was modified to study the differences among the results, such as the case of Gharbi et al. (2021) and Van Aken et al. (2018) for toxicity, Frenda et al. (2018) for misogyny, and Ranasinghe & Zampieri (2020) for offensive language, among others. The authors mostly chose between SVM, LR and RF models, being SVM the favourite. However, in some cases LR showed high Precision (misogyny), as well as RF (toxicity). At the same time, authors also explored with ensembles of different ML classifiers, which improved the results in some of the cases due to the expansion of performances combined in one single classifier, or to ensemble layers of the same classifier, such as the case of a multi-view SVM proposal for hate speech detection.

Another interesting approach was the use of ensembles. In some cases, the ensembles were built of different classical models (Shushkevich & Cardiff 2019 for misogyny), while other times they were combinations of ML and DL classifiers (Guest et al. 2021 for misogyny); in addition, another beneficial aspect was the combination of embeddings (word or sentence) as text representations with classical classifiers (mostly SVM), such as in misogyny tasks (García-Díaz et al. 2021) or in sentiment analysis (Cumalat Puig 2020).

Text representations were used widely in combination with ML models. It was possible to observe that authors explored more classical text representations such as BOW, uni-, bi- and trigrams (both in word and character level), and more modern representations such as WE (Word2Vec, GloVe and FastText) or the use of Transformers (BERT) for text representation. Authors mostly used FastText (Koratana & Hu 2018, Van Aken et al. 2018, Ahluwalia et al. 2018) and Word2Vec (Shen & Rudzicz 2017, Tadesse et al. 2020, Dessì et al. 2020), and in fewer terms, also GloVe (Dessì et al. 2020); being FastText the favourite among the studies. In addition, some of the authors decided to perform minimal pre-processing, but others suggested that POS tagging (to detect the core words: nouns), Tokenization, Lexical resources (lists of words), removal of stop-words, Lemmatisation, and even NER and Sentiment analysis (emotion detection, sentiment classification, sarcasm detection) could be relevant to achieve better results. Also, it was important to note that the use of FastText could help overcome OOV problems, as well as its combination with Stemming seemed beneficial (while it worsened the performance with GloVe). It could also be appreciated that self-trained models (vectors trained on the domain dataset or on the same characteristics as the test set) improved the performance as well. Finally, the use of community approaches seemed to achieve better results in some of the studies, and syntactic or semantic dependency based on WE (finding words that worked similar or could be used in similar contexts), and word similarity and relatedness methods in addition to discourse information showed improvements and promising results.

Next, the use of linguistic, contextual, or semantic information as well as lexical features was also beneficial in ML models. Keyword-based approaches were performed by MacAvney et al. (2019) for hate speech, and Ramírez-

Cifuentes et al. (2020) for suicide detection. Linguistic features analysis was made by García-Díaz et al. (2021), and Frenda et al. (2018) for their misogyny study on stylistic features (N-grams), semantic (sentiment information), and topic information and context, in addition to using lexicons (meaningful words); Aldana-Bobadilla et al. (2021) focused on semantic information (sentiment) for misogyny tasks; and, for suicide, Ji et al. (2020) paid special attention to textual context analysis (lexicon-based filtering and word cloud visualization) and feature engineering (textual and affective features). In addition, the use of topic detection, combined with lexical features (N-grams) proved beneficial in anxiety studies (Shen & Rudzics 2017). Feature selection was a crucial step in the successful studies: lexicon-based features, such as having word lists or polarity lexicons improved the results of the models, and text-based features also played an important role, for example in the use of socioeconomic or health-related information in the case of suicide detection, or topic detection in the case of anorexia. Finally, semantic features also seemed to improve the performance of the classifiers, such as with sentiment or polarity analysis

In addition, some authors also added domain data in the training set to feed an SVM classifier in the case of abusive language detection (Karan & Snajder 2018). In some cases, the best performances oscillated between classical models: SVM and RF (Ryu et al. 2019 for suicide, Ožegović & Celin 2020 for toxicity), LR (Guest et al. 2021 for misogyny, O'Dea et al. 2015 for suicide, Koratana & Hu 2018 for toxicity) or NB (Gharbi et al. 2021 for toxicity).

Regarding multilingualism, classical methods proved useful in less-resourced languages or other languages than English. For example, in the use of SVM for Spanish in offensive language detection (Ranasinghe & Zampieri 2020) or in the case of Tunisian for toxicity (Gharbi et al. 2021); or in addition, in sentiment analysis tasks for Catalan and Spanish (Cumalat Puig 2020).

As a final remark, it could be appreciated that lately tasks on toxicity detection were focused mostly on DL models and, specially, BERT-based models. However, as seen in other controversial behaviour studies such as anorexia, misogyny, or suicide, it may lack variation and different approximations. It was proved that classical ML models (whether combined with embeddings or not) still performed in a competitive and confidently enough way that deserved to be further studied. For that reason, some of the authors concluded that the benefits of using classical classifiers were *interpretability*: showing that it was easier to interpret and analyse the results and the decision choices of classical algorithms; and *timing*, which was faster in ML models. Also, some authors that tested both ML and DL models seemed to achieve similar results between both. For that reason, they concluded that given the few differences of performance, classical models were a better option to experiment.

Systems based on fine-tuned pre-trained models had already outstanding results and they were hard to improve, but at the same time they were black boxes and did not allow to advance much in the knowledge of the problem. Once we got the results of such models, we could still ask ourselves: why did we get the prediction that a sentence is toxic? Based on what? If we had better knowledge of the problem -in this case toxicity detection- and which elements could benefit or improve the performance of such models, it would be possible to apply previous filters to detect toxicity in early moments or in the exact moment of the publication of the toxic text. This was the starting point of the research proposed in this thesis.

3. Methodology

In this chapter, we introduced and presented some preliminary concepts of *semantic vector space* and *semantic composition*. We also presented the design of the algorithm and experiments, in addition to the collections and evaluation metrics we used to evaluate the performances of the models.

3.1. Preliminaries

For semantic vector space we meant the way of representing the meaning of words as dense vectors in a high-dimensional space. We related this spatial arrangement in terms of isometry, i.e., the correspondence between the orientation of the embeddings within the vector space and the meaning space.

On the other hand, semantic composition (formally known as “Principle of Compositionality”, Partee et al. 2012) is a process by which the meaning of a whole was represented by the meaning of its parts. Thus, in sentence composition, the meaning of whole sentences is determined by the meaning of their parts: in our case, the words composing a sentence. We also introduced two possible composition functions: *global avergae* and *f_inf*.

3.1.1. Semantic vector space

In semantic vector space models, the meaning of a word is represented as a vector in a high-dimensional space, and these spaces are assumed to maintain semantic isometry. In this work we wanted to test whether there was also some kind of ***anisotropy*** (Liang et al. 2021) within the vector space of word meaning and toxicity content; that is, whether word vectors were distributed throughout the multidimensional space uniformly, or whether words representing toxicity tended to be oriented in certain directions. This is one of the fundamental hypotheses of this research work: if vectors related to toxicity were oriented in narrow cones (which would represent an anisotropy) and whether we could use this information for the detection of toxic messages in social media.

To establish this anisotropy in toxicity we measured average similarity (by means of the cosine function) between toxicity-related words and non-toxicity-related words and checked if the cosine differed in both cases. If we found that in toxicity the average cosine was higher than the cosine in non-toxicity, we could say that there was some associated anisotropy and use this fact to detect whether a message was toxic or not. In the vector space we work, we assumed semantic isometry: a correspondence between the proximity of the statements in the representation space versus their (true) affinity of meaning.

For representing and illustrate such isometry in the semantic space we used Static Word Embeddings (SWE), which maintain some linguistic relations between words, while Contextual models did not always maintain this isometry. It was shown that Contextual Embeddings (CE) concentrated the representation of words in areas of space, known as the *representation degradation problem*. This resulted in that they were not very effective in terms of text representation within semantic space (Gao et al. 2019, Reimers & Gurevych 2019, Amigó et al. 2022). This representation degradation showed some biases, for example that most frequent tokens concentrated in a cone in the embeddings space, while the less frequent ones had a sparser space (Gao et al. 2019). Contrastive learning methods illustrated that anisotropy was part of the problem in these Contextual models. However, after several experiments, Jiang et al. (2022) encountered certain biases in the BERT model which showed that the anisotropy was not always related to poor semantic isometry. These biases were present in the embedding space, which made encode non-semantic information (like frequency of tokens), which distorted the cosine similarity and thus, leaded to poor performances (Fuster & Fresno 2022, Fuster 2022). Nevertheless, the observations by Jiang et al. (2022) lighted controversy as they contradicted previous works (Gao et al. 2019, Ethayarajh 2019, and Li et al. 2020). In addition to this problem, there was an added fact which was that there was no standard method to evaluate anisotropy, as Ethayarajh (2019) evaluated it at word level, but Jiang et al. (2022) evaluated it at sentence level (Fuster & Fresno 2022, Fuster 2022). Therefore, in this thesis we focused on Static Word Embeddings' vector space. In addition, sentence representations within this semantic vector space were obtained by means of semantic composition function (Amigó et al. 2022).

3.1.2. Semantic composition

Composition is defined as *the meaning of a complex expression is determined by its structure and the meaning of its constituents*[22]. An example of this process could be observed in the ambiguous sentence: *Visiting relatives can be boring*. Here, in addition to the independent meaning of the word or word-groups for the full meaning of the sentence (*meaning of its constituents*), we also needed to comprehend how the *structure* influences its meaning. This sentence is ambiguous, thus, it could be understood as “when someone is visiting their relatives can be boring” or as “relatives who visit can be boring”. As seen in the example, this sentence could have two different meanings depending on how the composition process is made regarding its syntactic structure: if “relatives” is an argument of the verb “visiting” or if it is a constituent group “visiting relatives”.

Applying this process to vector composition, we obtained the embeddings of each of the words of the sentence using Word2Vec. The reason was that it approximated more the semantic isometries, and, for that reason, we expected that non-supervised composition of sentences would work better in this space than in the space of the contextual vectors of, for example, BERT, which suffers from the problem of representation degradation, as indicated above.

In terms of semantic composition of sentences, in this thesis we followed the paper *Information Theory-based Compositional Distributional Semantics* by Amigó et al. (2022), in which the authors first established formal properties for embedding, composition and similarity functions based on Shannon's Information Theory; and then proposed a parameterizable composition function that generalized traditional approaches while fulfilling the formal properties. Finally, the authors performed an empirical study, showing that managing formal properties affected positively the Accuracy of text representation models in terms of semantic isometry.

The composition functions used here were *global average* and *f_inf* (Amigó et al. 2022). The vector average function (global average) is a strong baseline for compositional tasks (Amigó et al. 2022, Boleda 2019, Lenci 2018, Blacoe & Lapata 2012, among many others). However, a negative point of this

[22] https://plato.stanford.edu/entries/compositionality/

function is that its sensitivity to the addition of null information, thus, it did not satisfy desirable conditions of composition with neutral elements (null information components did not affect the composition). It also failed to satisfy composition of normal monotonicity (the norm of the composite vector is monotonic with respect to the angle between the compound vectors). To satisfy those properties, Amigó et al. (2022) proposed a general composition function based on Information Theory: *f_inf*, which also generalized approaches such as the vector sum or pairwise average. This function satisfied the property that two vectors with the same direction resulted in a longer one, and consequently, this meant that adding redundant information did not affect the original embeddings (it did not increase the amount of information). Another addition to *f_inf* was that it satisfied another desirable property: sensitivity to structure, where linguistic structure and order in which linguistic units were composed affected the composed embedding. See a detailed explanation of these properties in Amigó et al. (2022).

- *Global average*

To illustrate the composition process using *global average* (*f_ga*), we firstly calculated each vector of each word that composed the sentence and then we calculated the average of all those vectors per each sentence. The result was a vector that represented the average of each of the elements of the sentence, thus, representing the sentence vector. The process is illustrated below:

1) Sentence = word_1 word_2 word_3 word_4
   Sentence vectors = {vec_1, vec_2, vec_3, vec_4}
   Composition by means of global average: $(\text{vec}_1 + \text{vec}_2 + \text{vec}_3 + \text{vec}_4) / n = \text{vec}_z$, being *n* the length of the sentence and vec_z the final composed sentence vector

- *F_inf*:

In the case of *f_inf* we used a recursive method per each pair of words of the sentence. We firstly obtained the vectors of the words that composed the sentence with Word2Vec. Afterwards, we calculated the composed vector result of the first two vectors (v_1, v_2) of the first two words of the sentence. The resulted vector (v_x) is again composed with the third vector (v_3) of the third word of the sentence, and so on until we got the last vector (v_z) which was the vector of the whole sentence. This process of recursive composition is illustrated below:

2) Sentence = $word_1$ $word_2$ $word_3$ $word_4$

   Sentence vectors = vec_1, vec_2, vec_3, vec_4

   Composition:

   Recursion step 1: *f_inf* (vec_1, vec_2) = vec_x

   Recursion step 2: *f_inf* (vec_x, vec_3) = vec_y

   Recursion step 3: *f_inf* (vec_y, vec_4) = vec_z

   Being vec_z the final composed sentence vector

3.2. Evaluation

In this section we presented the evaluation collections we used for training and testing our proposals: Kaggle, OLID, UCC and AMI

In addition, we presented the evaluation metrics, which are used to evaluate the performance of a classification model and allowed us to interpret the results and compare them with other models' results.

3.2.1. Evaluation collections

Our main dataset, which we used for both training and testing, is Kaggle.

Kaggle. The Jigsaw – Toxic Comment Classification Challenge[23] was published in a Kaggle competition in 2018. The justification on such challenge was the increasing amount of toxic content in online platforms. Such behaviour could impact in different ways how people expressed themselves on the internet getting to the point that they would even stop communicating in online platforms at all. Jigsaw and Google worked towards the improvement of online conversations, which focused on the study of negative online behaviours, like toxic comments (rude, disrespectful, etc.). The training dataset conforms approximately 160000 English comments from Wikipedia's talk page edits that had been labelled by humans for toxic behaviour. The toxic distribution consisted

[23] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

of around 16500 (10%) comments labelled in one of these different types of toxicity: *toxic*, *severe toxic*, *obscene*, *threat*, *insult*, and *identity hate*. The rest were *not toxic* (90%). If a comment presented any kind of toxicity, it was labelled as “1” in the corresponding column, otherwise, if the comment was “fine”, it was labelled as “0” in all the toxic columns. Its test set had around 64000 sentences, which were divided into 6243 (10%) toxic ones and 57735 (90%) not toxic.

We also wanted to introduce that we performed some observatory experiments on other 3 datasets: OLID, UCC and AMI; in addition to the Kaggle train set. The reason was that we wanted to verify if the observations extracted from the Kaggle dataset were systematic and consistent with other datasets.

OLID. The Offensive Language Identification Dataset (Zampieri et al. 2019) was used in task 6 from OffensEval: Identifying and Categorizing Offensive Language in Social Media[24] in 2019. The motivation was to identify *offense*, *aggression*, and *hate speech* in user-generated content. Offensive language was pervasive in social media, and individuals took advantage of the perceived anonymity of computer-mediated communication, using this to engage behaviour that many of them would not consider in real life. Online communities, social media platforms, and technology companies had been investing heavily in ways to cope with offensive language to prevent abusive behaviour in social media. OLID contained 14100 English tweets annotated in 3 different labels, but we only focused on the labels for offensive language identification which was *NOT* for not offensive tweets and *OFF* for tweets containing offensive language. The train dataset contained 13240 tweets of which 8.840 (67%) were labelled as *not offensive*, and 4400 (33%) were labelled as *offensive*. The test set had 860 sentences, and it was divided as 620 (72%) not toxic and 240 toxic (28%).

UCC. The Unhealthy Comments Corpus was a widely used corpus in different tasks. For example, it was listed as one of the corpora to be used for the Workshop on Online Abuse and Harms[25] (WOAH). The main goal of the task was to focus on social bias and unfairness in online abuse detection. The corpus consisted of around 35500 comments identifying subtle attributes which contribute to unhealthy conversations online. The labels were *healthy* or

[24] https://competitions.codalab.org/competitions/20011

[25] https://www.workshopononlineabuse.com/cfp

unhealthy. In addition, binary labels for the presence of six potentially *unhealthy* sub-attributes: (1) *hostile*; (2) *antagonistic, insulting, provocative or trolling*; (3) *dismissive*; (4) *condescending or patronising*; (5) *sarcastic*; and/or (6) an *unfair generalisation*. The division was around 31425 (89%) *healthy* comments and 4077 (11%) *unhealthy*. The test set consisted of 4384 sentences: 523 (12%) toxic (in any of the toxic categories) and 3861 (88%) considered healthy.

AMI. The AMI dataset was used in the Automatic Misogyny Identification[26] (AMI) task (from the IberEval 2018 Workshop). This task proposed the automatic identification of misogynous content both in English and in Spanish in Twitter. Unfortunately, nowadays more and more episodes of harassments against women arose and misogynistic comments could be found in social media, where misogynists hide behind the security of anonymity. Therefore, it is very important to identify misogyny in social media. The dataset is composed of 3307 Spanish tweets and 3251 English tweets. For our study we were mostly interested in the first annotation process of the corpus: if a tweet is *misogynous* (1) or *not misogynous* (0).

3.2.2. Evaluation metrics

To describe the output of the prediction of a class, we talk in terms of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Positive or Negative indicate the model prediction and the True and False indicate whether the prediction is correct or wrong (Paraschiv 2020).

Accuracy is measured by how close or far a measurement is from its true value. However, as most datasets are imbalanced, a high Accuracy can be misleading. For example, in a dataset with 90% positive classes and only 10% negative ones, if the system classified all the samples as the majoritarian class, it would most probably value an Accuracy of 0.90 due to this data misbalancing.

Precision is defined as the TP divided by the number of predicted positive values, thus, models with high precision would have most of the positive predicted classes correctly classified (Paraschiv 2020).

[26] https://amiibereval2018.wordpress.com/

Recall measures how many of the real positive class items were detected, thus a low recall showed that many positive class items went undetected (Paraschiv 2020). In toxicity (or other forms of toxic behaviour) detection, such as our case, this is the most important metric to consider. With Recall we make sure that we classified as *toxic* actual toxic content, even if that means that we classified as *toxic* content that was not toxic (which would lower the Precision of the classifier). However, even paying this price, it is better to have a false alarm in not toxic messages classified as toxic, than to miss a truly toxic message.

F1-Score is the harmonic mean of Precision and Recall. Thus, higher F1-Score would indicate better overall performance of a model (Paraschiv 2020). In addition to F1-Score, there exist more F1 weighting schemes: Micro-F1 and Macro-F1. **Micro-F1** does not consider class membership for any test sample, that means that the calculation is made over the dataset, while **Macro-F1** gives an equal weight for each class with positive sample count regardless of the specific count (Harbecke et al. 2022).

ROC AUC stands for Receiver Operating Characteristic (ROC), which is a probability curve, and Area Under the Curve (AUC), that represents the degree or measure of separability. In more detail, ROC is a graph-like metric that shows the performance of a classifier at all classification metrics and is based on two parameters: True positive rate (Recall), and False positive rate, (dividing TP by the sum of FP and TN). On the other hand, the AUC measures the area under the ROC curve, where it provides the measure of performance of all possible classification threshold, which is a probability of a random positive to be positioned to the right of a random negative sample[27]. This metric reflects the overall ranking performance of a classifier (Narkhede 2018).

[27] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=es_419

3.3. Baselines

To corroborate our hypotheses, we proposed a baseline based on TF-IDF weighing function within a bag-of-words (BOW) approach. Here, TF represented the Term Frequency (*t*) in a sentence or document (*d*), and IDF (Inverse Document Frequency) gave more weighting importance to terms that happened less frequently in the document or corpus (*D*) and less weighting importance to terms that happened more often. Finally, the TF-IDF was calculated as the product between TF and IDF: TF-IDF(t, d, D) = TF(t, d) · IDF(t, D).

In addition, we showed results from other different systems (based on traditional ML algorithms, BERT, etc.) in the test collection. As our main test collection was from the Kaggle – Toxic Comment Classification Challenge (see Section 3.2.1), we selected other systems from the challenge page: from the top 3 results from the competition's leaderboard, which were evaluated within the competition's evaluation metrics: ROC AUC.

- *Toxic Crusaders* (To Train Them Is MY Cause & Chun Ming Lee): with diverse pre-trained embeddings, translations as Train/Test-Time Augmentation (TTA), rough-bore pseudo labelling (PL), and a robust CV and stacking framework. They tested standard models and showed that both TTA and PL techniques worked effectively. They finally decided to work with a Bi-GRU model with which they scored in first place with 0.98856 ROC AUC;
- *neongen & Computer says no* (Computer says no & neongen): they built an ensemble of RNN, DPCNN and GBM models, trained on pre-trained embeddings, with TTA using translations to German, French and Spanish and back to English, and trained on translations using DE, FR, ES BPEmb pre-trained embeddings. They scored second with 0.98822 ROC AUC; and
- *Adversarial Autoencoder* (Alexander Burmistrov, Andre Naef, Bohan Tunguz & ryches): implemented a RNN model with word embeddings, concatenating FastText and GloVe, with a lot of text normalization work such as correcting misspellings with TextBlob dictionary, created lists of words that appear often in each category, and finding word vector neighbourhoods with FastText for the OOV words. They scored third with 0.98805 ROC AUC.

We also selected three systems among the participants that used different approaches and evaluation metrics. For that we filtered the *best value* section and selected the top ones (bronze category) that evaluated their systems with Accuracy and F1-Score:

- Abhishek Kumar Mishra (AKM)[28] fine-tuned a BERT model achieving 0.93 Accuracy, 0.81 Micro-F1 and 0.71 Macro-F1;
- Ananta Raj (AR)[29] used a custom sequential model with GloVe, reaching 0.93 Accuracy and 0.65 F1-Score; and
- Mustafa Fatakdwala (MF)[30] used TF-IDF and LR model, scoring 0.95 Accuracy and 0.77 F1-Score.

We could refer to two authors from the literature review (see Section 2.1.11) that use this same dataset and illustrate again their results as well:

- Koratana & Hu (2008) (K&U) used a Bi-LSTM (Attention + FastText) model, with which they scored 0.66 F1-Score and 0.99 Accuracy,
- Van Aken et al. (2018) (VA) used an Ensemble model with Gradient Boosting Decision Trees, scoring 0.88 Recall, 0.79 F1-Score and 0.98 ROC AUC.

3.4. Experiment description

In this section we presented the design and preparation of the **two experimental lines**: the first one focused on ***semantic orientation*** and the second one focused on ***linguistic structure***.

3.4.1. Experiment design

The experiments were aimed to answer the two research questions presented in this thesis: RQ1: *Does semantic representation in toxicity have any kind of*

[28] https://www.kaggle.com/code/eggwhites2705/transformers-multi-label-classification
[29] https://www.kaggle.com/code/vortexkol/glove-toxic-comments-classification
[30] https://www.kaggle.com/code/mastmustu/toxic-comments-classifications-using-ml

orientation bias in the semantic representation space? RQ2: *Does toxicity have any kind of inherent syntactic structure?*

If the answer to those question was “yes”, we would wonder whether we could use such information to detect new toxic messages.

3.4.2. Algorithm proposals

The used algorithm was based on the median proximity. For that we understood the proximity to the median vector of the embeddings, either to the toxic or the not toxic words, or to the toxic or not toxic sentences.

We were working in the Word2Vec vector representations; thus, we obtained the vector representations of our embeddings from that space. To calculate the proximity to the median we used the cosine function, as it was proved that Word2Vec and the semantic similarity metric from the cosine in the Word2Vec space respected semantic isometry (Levy and Goldberg 2014, and Arora et al. 2016). Thus, our algorithm proposals focused on the mean vector and the mean cosine.

- *Mean vector*

In this first proposal, we calculated the mean vector (or centroid vector) of two lists of words: a list of *toxic words* and a list of *not toxic words*. Once we obtained the lists of words, we extracted their vectors using Word2Vec and we finally calculated the mean vector of all the words per each list.

Then, we calculated each sentence vector (for each sentence of the training dataset) using the semantic composition approximation introduced in Section 3.1.2, and we calculated the total mean vector of all the sentence vectors. This allowed us to calculate and compare the angular distance between the toxic and non-toxic words and sentences, and then we were able to apply this same process to new words or sentences to calculate whether they were toxic or not. As say, for each new sentence, we compounded and calculated the sentence representation vector and we calculated the cosine similarity between the new sentence vector and the previously calculated mean vector of the lists of toxic and not toxic words, and the mean vector of the toxic and not toxic sentences. Finally, we assigned the correspondent label focusing on which one had the fewest distance with the median vector of the words and of the sentences.

- *Mean cosine*

With our second proposal, the mean cosine, we illustrated if there was an orientation bias in toxic sentences, or if toxic sentences were in the semantic space without any kind of orientation bias. With this we wanted to prove whether toxic content representations (words and sentences) had a semantic orientation towards any direction in the space. If that assumption was correct, toxic content representations would form a cone around the median vector. Finally, if there was a cone, it would mean that toxic sentences had an orientation bias in the semantic space and, thus, we would be able to apply this same process to new words or sentences to calculate if their cosine fell inside or outside the toxic cone.

3.4.2.1. Semantic Orientation

For calculating the semantic orientation of toxic words and sentences and prove if there really was a semantic-directional bias (as Word2Vec should maintain the isometry) we developed two algorithms: a TF-IDF baseline and a proposal using Word2Vec. The reason to choose both representations in our experimentation was because TF-IDF and Word2Vec are representations within *different* vector spaces. We wanted to prove if there was a contrast between TF-IDF (a classical baseline based on Vector Space Model) and Word2Vec, a semantic representation space. Thus, the latter would show directional bias that we could use when classifying messages as toxic or not toxic.

First, we calculated the mean vector of a list of toxic words and non-toxic words. To get the list of toxic and not toxic words, we used the library ProfanityFilter[31]. This library allowed us to perform a filtering, either by using *is_profane* or *is_clean* on the words from the training set to extract a basic list

[31] https://github.com/rominf/profanity-filter

of *toxic* and *not toxic* words. We extracted 200 *toxic* words and batches of 200 *not toxic* words, ordered by most to less frequent.

Once we got the list of words, we obtained the Word2Vec vectors for our proposal (note that words could not be represented with the TF-IDF term weighting function, which did only allow sentences to be represented).

Then, we calculated the cosine of all the vectors of all the words to finally extract the mean cosine. The results showed that the mean cosine of the 200 toxic words was 0.30 for Word2Vec, while the mean cosines of the batches of 200 not toxic words went from 0.14 (200 most frequent words) down to 0.08 (200 less frequent words). These results supported our claims that there was indeed a toxic bias in the semantic space. The higher value in mean cosines represented a more closed cone, which was related to a greater bias in toxic content. These results started giving light to our first research question: there seemed to be a toxic orientation in the semantic representation space, and terms considered toxic presented a bias in their representation inside the semantic representation space.

In the case of the sentences, we performed this study mainly in the Kaggle training set but, in addition, we corroborated the result with the other three datasets. For Kaggle, we selected batches of 15000, 5000 and 2000 sentences, and concluded that using 5000 sentences was the optimum number to proceed, as there were minor changes between this value and bigger ones.

To extract the cosine values, we calculated the vector with TF-IDF for each sentence of the dataset, extracted the cosine of all the vectors and finally calculated the mean cosine. The results in Table 1 showed that the mean cosine of the *toxic* sentences was 0.14, while for the *not toxic* sentences was 0.18.

For Word2Vec, we performed a similar process going further with the composition functions: both *f_inf* and *f_*ga. We composed the sentence vector with the functions and calculated the cosine similarities between all the resulting sentence vectors. We finally calculated the mean cosine of all the similarities.

We concluded to proceed using *f_ga* as the main composition function as the results between the mean cosines were more relevant and illustrative than the ones using *f_inf* (see Table 2), which mostly just showed more closing of the cones of the cosines (higher values), but without further relevant information.

TF-IDF			
Dataset	*Number of sentences*	*Toxic*	*Not toxic*
Kaggle	5000	0.14	**0.18**
OLID	3400	0.16	**0.23**
AMI	1500	0.06	**0.11**
UCC	3000	0.19	**0.23**

Table 1. Mean cosine using TF-IDF representations

Word2Vec		***F_ga***		***F_inf***	
Dataset	*Number of sentences*	*Toxic*	*Not toxic*	*Toxic*	*Not toxic*
Kaggle	5000	**0.42**	0.26	0.55	**0.59**
OLID	3400	**0.17**	0.12	**0.51**	0.47
AMI	1500	0.19	**0.26**	**0.58**	0.52
UCC	3000	**0.16**	0.12	**0.54**	0.51

Table 2. Mean cosine using Word2Vec with f_ga and f_inf

As it could be appreciated from Table 1, there was no clear correlation between toxic and not toxic cosines while using TF-IDF. This was an expected behaviour and made sense in terms that TF-IDF was *not* a semantic space, thus, there were no bias in the representation and, consequently, there was no isometry in the semantic space. In the case of Word2Vec, it could be appreciated from the results that toxicity had indeed a semantic orientation in the vector space (isometry), which could be observed as the cones were more closed (the cosine similarity was higher) in the toxic side of the datasets.

We would like to note that the AMI dataset was slightly different, that is because this dataset was created to study misogyny detection and they did not consider other forms of toxicity. After inspecting the dataset in detail, we could appreciate that sentences considered not misogynistic ("not toxic" for us) had proper toxic elements, such as insults, profanity, etc. Thus, toxic terms and sentences were mixed in the categories of this dataset. Consequently, we will not be considering this dataset for further experimentation.

The results of our experimentation support our question of whether toxic messages tended to have a semantic orientation towards a specific direction in the semantic representation space, as we could observe that the cosines were more closed (higher values) in the case of toxic sentences, which translated into bias towards a “toxic” orientation (isometry).

With these first experiments we could answer “yes” to our RQ1 and therefore prove that **there exists a semantic orientation in toxic content while using Word2Vec embeddings**. Next steps would be to experiment if we could use semantic orientation to detect new toxic messages, which will be examined in the following chapter where we experiment with the test sets.

3.4.3. Representation proposals

The representations we used to corroborate our hypothesis were divided into two main groups: linguistic structure and composition functions.

3.4.3.1. Linguistic Structure

Regarding linguistic structure, for the TF-IDF baseline we did not consider any special case because this representation model is based on the Independence Principle, which assumed that there was no relationship between the words within a sentence and, therefore, no linguistic structure could be assumed. This is obviously false, but it is a widely used approach and, in many cases, allowed simplifying the problem and obtaining good results. Thus, we calculated the vectors of the sentences considering stopwords and all morphosyntactic categories. However, in the case of Word2Vec, we experimented with different combinations of categories, and both with and without stopwords.

- *Stopwords*

In the case of with or without stopwords, we considered that stopwords should not contribute in a semantically relevant way to the toxic orientation of the sentences. We considered words such as copulative verbs, articles or pronouns to have a rather neutral meaning. Still, there were other elements that could play an important role in meaning: prepositions, negation particles or determiners.

Prepositions

We considered that there were some relevant prepositions that could play an important role in explaining and contrasting the semantic meaning in a sentence, which could completely change its toxicity content. For example, let us observe the following minimal pair of sentences:

(1) I feel something *towards* her.
(2) I feel something *against* her.

In this case, we could appreciate that both sentences had the same syntactic structure, used the same verb, and had the same subject and object. However, the only difference was in the preposition, which completely changed the meaning of (1), with a positive meaning, and (2), implying a negative meaning. Although this could be a relevant factor for toxicity detection, it would be proper to think that prepositions are more an "added" factor in toxicity, as many of the most common phrases that express toxicity are formed by the structure N + prep + N, such as "*son* (N) *of* (prep) *a bitch* (N)" in English or "*hijo* (N) *de* (prep) *puta* (N)" in Spanish. These examples showed that the preposition does not really add any relevant semantic meaning to the structure but rather is a way of union or relation.

Determiners

In second place, determiners could also affect the polarity of a sentence. We had in mind comparatives, as a comparison could change a lot its meaning if we used "more" or "less" as determiners:

(3) You are *more* intelligent than me.
(4) You are *less* intelligent than me.

As we could observe, sentence (3) had a positive meaning towards the subject, while (4) had a negative one. Again, this were aspects that could be relevant in tasks of toxicity detection and are lost when eliminating stopwords. Although this example shows a relevant side of determiners, there is a big number of other determiners that could be considered fillers and not provide further semantic meaning, such as articles or demonstratives.

Negation

Finally, we considered negation as a crucial and complicated factor. It has been widely studied in both theoretical linguistics (Klima 1964; Jackendoff 1969; Espinal 1992; Ripley 2009; among many others) as well as in NLP (Abderrouaf et al. 2019; Britto & Khandelwal 2020; Khandelwal & Sawant 2019; Scaboro et al. 2021) without clear conclusions. Negation is a complex phenomenon that could be expressed in many ways and with many different linguistic structures, in addition to being differently manifested among languages.

Martí et al. (2016) compiled different ways of expression negation in Spanish, such as simple negation (5), using one negative particle; complex negation (6), using more than one negative particle (with different subtypes); and negative structures that do not express negation (7), among many others.

(5) *Sin* conexión (*Without* connection)

(6) *No* vino *nunca* (*NO* He *never* came)

(7) *Sin* pena *ni* gloria (*Neither* very negative *nor* very positive; average)

However, let us observe (8) and (9) as well. As it was logic, it was not the same to say either one, thus the fact that we are, again, changing the polarity of the sentence with one single element should be something to consider in NLP and in toxicity detection tasks because the first example was clearly a positive sentence, while the second one had a negative meaning.

(8) I do like you.

(9) I do *not* like you.

As we could appreciate negation is expressed in many ways, which made it even more difficult to present an approach on how to treat it, and further, how to analyse it in NLP. Even so, leaving negation out of the equation in the context of stopword elimination could as well facilitate the analysis of toxicity in the sense that toxic sentences tended to be more complex and linguistically more structured than a simple negation case. An example of how these sentences (and other of this kind, called adversary examples), disturb the performance of classification models in toxicity detection tasks could be observed in Hosseini et al. (2017), where they experiment with Google's Perspective[32] system to prove the difficulty added and how harmful those examples are. We tested the sentences from (10) and (11) using Perspective, and we could surprisingly observe that they were both labelled as *toxic* even though they expressed a completely contrary meaning.

(10) You are a bitch and an asshole.

(11) You are *neither* a bitch *nor* an asshole.

Although Perspective labelled (10) as toxic with a 97% confidence, which is correctly identified; it also labelled (11) as toxic with an 81% confidence. Even though the confidence value was lower, it was still a high value given the fact that the sentence presented two negative particles. This proved the complex nature of negation and the added difficulty to identify it. Thus, it could be a good idea to leave it out in studies where there was plenty of data where negation was not as present nor relevant, but it was still an issue that deserved further study as examples like the one in (11) could be found in online content and should not be labelled as toxic and banned.

Applied to our study, we were either removing stopwords in the moment of the data pre-processing or not. So, in the case of structure without stopwords, we added a step that removed them from the sentence and proceeded with a sentence without stopwords. For that we used NLTK[33]'s stopwords package.

[32] https://perspectiveapi.com/

[33] https://www.nltk.org/

- *Morphological categories*

In a parallel to stopwords consideration, we could also contemplate the use or not use of selected morphological categories. Some stopwords such as articles, did not add valuable meaning to a sentence and we wanted to examine if that could be also the case in some morphological categories. For simplicity, we divided them in the four main ones: Adverbs, Adjectives, Nouns, and Verbs.

<u>Adverbs</u>

It was possible to imagine that some categories would be more relevant than others in tasks of toxicity detection, for example we could expect that Adverbs could be more present in healthy sentences, which are longer and more correctly formed, rather than in toxic sentences where the aggressiveness and arouse of negative emotions could make the authors of such messages to not use that much such category. For example, it would probably more common to find sentences such as (12) and (13) in healthy contexts rather than in toxic ones:

(12) She spoke *softly*.
(13) She spoke *hurtfully*.

<u>Adjectives and Nouns</u>

In the case of Adjectives and Nouns, we could consider them to be used in both contexts: toxic and healthy. However, we could think about the use of adjectives in noun forms (14) or in noun compounds (15), illustrating its use in toxic contexts. Nouns are widely used in toxic content as they were plenty used in toxic phraseology like in (16) and in many ways to insult (17).

(14) Nasty [person] (Adj).
(15) Hateful (Adj) bastard (N).
(16) Piece of shit (N).
(17) Asshole (N).

Verbs

Finally, in the case of Verbs, there were no toxic verbs per se, however many could have negative connotations (18) or be used in contexts where they acquire toxic meaning (19):

(18) Kill yourself vs. Kill all the spiders.

(19) Eat my pants vs. Eat the pasta.

All morphological categories could be used in toxicity context, however it seemed that some tended to be more present than others. That could be the case of Nouns and Verbs, and in minor account for Adjectives; while Adverbs would be the category that we expected the less in toxic messages.

In our study, we considered morphological categories in the moment before performing the composition. We tokenized and performed a POS tagging using NLTK's word_tokenize and pos_tag to the elements of the sentence and selected only the ones that had the desired category, and thus, only compose with the selected elements. As say, if we were interested in composing only with Nouns, we would only select the 'N' tags; but if we are interested in both Nouns and Verbs, we would select 'N' and 'V' tags.

3.4.3.2. Composition functions

In this study we considered two ways of performing composition: ***f_ga*** and ***f_inf*** (see Section 3.1.2). In the case of *f_ga*, we compounded all the vectors that formed the sentence at once: we sum the vectors at once, no matter the order. However, in the case of *f_inf*, as the composition was recursive, it depended on directionality. In this case we explored two possible directions: *left to right* and *right to left*. Thus, we considered if we started the process from the leftmost vectors of the sentence or, on the other hand, the rightmost vectors.

- *Left to right (L2R)*

In this case we considered the order of a sentence in left to right (L2R) direction. In this case we performed the composition of the vectors of the sentence starting from the first element in the left with the second leftmost element. Once we obtained the vectors results of the composition, we proceed to compound the resulting vector with the third leftmost element, and so on until we reached the end of the sentence. In conclusion, in a sentence that contained 4 words (it has 4 vectors) this process would look like this: $[[[v_1\ v_2]\ v_3]\ v_4]$

$$f\_inf\,(f\_inf\,(f\_inf\,(v_1, v_2), v_3), v_4)$$

- *Right to left (R2L)*

In the second case, we used the order right to left (L2R). Now we performed a reversed process of composition, where we started with the vectors from the ending of the sentence, as say the last and the second-to-last vectors. We performed the composition of those vectors, and the resulting vector would be compounded with the third-to-last vector, and so on until we reached the beginning of the sentence. In this case, this process in a sentence that contained 4 words, and consequently had 4 vectors, would look like this: $[v_1\ [v_2\ [v_3\ v_4]]]$

$$f\_inf\,(v_1, f\_inf\,(v_2, f\_inf\,(v_3, v_4)))$$

- *Variables*

The composition process was performed after the stopwords and morphological categories were selected. That meant that we performed the composition processes as many times as needed with all the combination of variables: on the one hand, we performed the lineal composition, and both L2R and R2L processes with and without stopwords.

On the other hand, we fulfilled both L2R and R2L directions, with and without stopwords, and adding the morphological category selection, that meant that we performed the experiment with all four categories: N-V-ADJ-ADV; with all possible combinations of three categories (N-V-ADJ, N-V-ADV, N-ADJ-ADV, V-ADJ-ADV), with all the possible combinations of two categories (N-

V-, N-ADJ, N-ADV, V-ADJ, V-ADV, ADJ-ADV), and with each of the categories on its own (N, V, ADJ, ADV).

Table 3 shows a selection of the experiments' results. Here we could appreciate the use of all category tags and the use of each individual tag (N, V, ADJ and ADV), as well as the results with and without stopwords. The experiment was performed using the *f_ga* function.

F_ga		**Without Stopwords**		**With Stopwords**	
Kaggle 5000 sentences	*Type of POS*	*Toxic*	*Not toxic*	*Toxic*	*Not toxic*
	Without POS	**0.35**	0.32	**0.61**	0.51
	N	**0.30**	0.19	**0.37**	0.28
	V	0.02	**0.03**	0.02	**0.05**
	Adj	0.24	**0.34**	0.26	**0.40**
	Adv	**0.10**	0.08	**0.32**	0.28

Table 3. Mean cosines using POS categories and with or without stopwords

Once we obtained the results, it could be appreciated that the most relevant category to perform the experiments was the Nouns. We also observed that in the case of V and ADV, there were no bias at all, as the values were small, and in the case of the ADJ the bias tended to orientate towards not toxic content. It was in N where we could observe clearer the bias of toxic content. Also, all the experiments that contained stopwords closed more the cones, but not in a relevant way. That was the reason we decided to choose not to include them for our algorithm design. This corroborated previous experiments where this was also noted. For example, Jiang et al. (2022) or Ethayarajh (2019) noted that stopwords were contextual and, thus there was an increased similarity when they were considered for their experiments. To prove it, we also calculated the cosine similarity of a stopwords list, and we could observe that the mean cosine similarity of the NLTK's stopwords list (around 170 stopwords) was of 0.18.

We also performed this experiment with the other datasets and with the *f_inf* function, and the results continued in the same line. In the case of *f_inf*, we observed that the results for both L2R and R2L were similar, thus we proceed the experimentation only with L2R. In addition, *f_inf* showed more closed cones in all the cases, which translated into higher cosine values. We extracted similar results in the other datasets. In terms of morphological categories, we concluded that using N as the only tag for the sentence filtering was the one that gave better results in terms of the existence of a toxic bias, as well as we observed earlier.

These results seemed illuminate the idea of whether toxicity has any kind of inherent syntactic structure. We could observe that, although a recursive structure (either in L2R or R2L order) did not seem to highly influence the results and a lineal structure was proven to work well so far, there were other elements that showed relevance in structure terms. The presence of some categorical elements, such as stopwords (as they have their own bias), and some morphological categories, such as V and ADV, which did not present any kind of bias, seemed not to be important elements to consider in toxicity tasks.

The case of ADJ was surprising as we could observe that they tended to be biased towards healthy comments, probably more elaborated sentences tended to have more elements and, thus, more adjectives. Finally, we could prove that the presence of N was the relevant aspect in toxicity bias, as nouns showed a semantic orientation towards toxic content while reducing the orientation of healthy content. That can be due to a lot of phraseology or insults, that are usually nouns. This allowed us to give an open door for the experimentation to check if we could use such structure to detect new toxic messages.

As a conclusion, we decided to select an algorithm composed by the *f_ga* function: a function that performed composition of each of the vectors from each of the words of the sentence in a linear order and all at once. We also concluded that using stopwords was not relevant for our experiment because stopwords already had a semantic orientation and it influenced in the closeness of the cones (however irrelevant for our experiments). Finally, we only considered words tagged as Nouns as they were the elements of the sentence that showed more differences when calculating the mean cosine of the sets in terms of semantic orientation of toxic content.

Thus, our proposal algorithm, which we called *Ansun* (Average, No StopWords, Noun), will have the following properties: “f_ga + no_sw + pos_N”.

4. Experiments

In this chapter we presented the experiments we developed to test our hypothesis.

The first experiment involved testing *vector proximity* in a space with semantic isometry between new test sentences, those of which we want to know whether they are toxic, and the mean vector of a list of toxic and not toxic words representing the hypothetical orientation of "toxicity" in the space. The proximity was calculated with the cosine similarity between both mean vectors.

The second experiment tested *orientation proximity* of the mean vector of toxic words and toxic sentences. As seen, toxic messages seemed to present some degree of bias in the representation space, that is why we wanted to verify if new toxic messages would present the same bias or some sort of semantic orientation towards semantic toxicity. Thus, we would verify if toxicity has indeed semantic orientation.

The third experiment consisted in evaluating a first naïve algorithm proposal for detecting toxicity, for that we used the *n nearest neighbours* algorithm applied to the test sentences and to corroborate if they were closer to words (or sentences) considered toxic or to not toxic ones, by seeing which odd number of nearest neighbours is higher.

4.1. Experiment 1: Mean vector proximity

This experiment consisted of evaluating if, once we obtained the mean vector of all categories (toxic words, not toxic words, toxic sentences, and not toxic sentences), whether a new sentence vector had more angular proximity to the mean vector of the toxic category or to the not toxic category of words or to the toxic or not toxic sentences. In other words, we wanted to verify if there existed some bias in toxicity using the centroid vector of our list of toxic words, not toxic words, toxic sentences, and not toxic sentences, and checked if the centroid vector of a new sentence was closer to which one. If the new sentence was toxic and the mean vector proximity was closer to toxic words/sentences, we would prove that there exists semantic orientation bias in toxic content.

For that, we performed the same process as with the training set. In the test set, we first extracted the vectors of the words that compound the sentence either with the TF-IDF baseline or with Word2Vec. With Word2Vec we applied the composition function *f_ga* to obtain the sentence vector. With this process we aimed to represent sentences in the same vector space as words, that is why we used the composition function on word vectors, and thus, we did not exit this same vector space. Once we obtained the sentence vector of the new sentences, we calculated its cosine similarity with the mean vector of the toxic words and the not toxic words: if the cosine similarity is greater (closer to 1) in the case of the closeness to toxic words, we would assign that sentence the label "toxic", as the cone would be more closed, and the similarity greater (and we will be observing a bias). Otherwise, we would assign it the label "not toxic". This same process was performed with the mean vector of toxic and not toxic sentences.

As we searched directional bias in toxicity, with this experiment we wanted to verify if such direction existed. As we already knew the label of the sentences in the test set, if the group of sentences already labelled as toxic were closer to toxic words or sentences, we would then corroborate our hypothesis as a toxic label would be assigned to them. Otherwise, we would have to claim that there did not exist directional bias in toxicity.

Results

We performed this experiment with the three datasets of our study (see Section 3.2.1). First and mainly, we performed the experiment on the Kaggle dataset while comparing our TF-IDF system and the proposed algorithm, in addition to the rest of the systems extracted from the competition and literature. Next, we performed the experiment with the additional datasets: OLID and UCC. As we commented previously, we did not continue with the AMI dataset as their aim goal was to study *misogyny* and *not misogyny* but without filtering other kinds of toxicity, which were observed in both labels.

After performing the experiment in the Kaggle dataset, we evaluated the model with the following evaluation metrics: ROC AUC, Accuracy and F1-Scores. We evaluated the system with ROC AUC to compare our performance to the top three performances of the best models in the Toxic Comment Classification Challenge from Kaggle. We also evaluated the Accuracy and the F1-Score to compare ourselves with the other three systems from the bronze category (AKM, AR and MF), and two systems from the literature (K&H and VA). One of the systems from the bronze category (MF) was also evaluated with ROC AUC so we included it as well.

The results of vector proximity to toxic and not toxic words mean vector are illustrated in Table 4, in addition to the results of other systems who were evaluated with ROC AUC. In Table 5, we illustrated the results evaluated with Accuracy and F1-Scores. As our systems (on a green background) were evaluated in both words and sentences, we would distinguish between:

- **TF-IDF** for the sentences in the case of the baselines; and
- **Wansun** for the words Ansun, and
- **Sansun** for the sentences Ansun algorithms.

Systems	ROC AUC
Toxic Crusaders	**0.98856**
neongen & Computer says no	0.98822
Adversarial Autoencoder	0.98805
MF	0.959021
TF-IDF	0.52395
Wansun	0.75
Sansun	0.77418

Table 4. Mean vector proximity results using ROC AUC

As it could be appreciated in Table 4 the results from the other systems are outstanding, going over 0.95 and 0.98. In our case, the results are 0.75 in the case of the mean vector of the list of words, and 0.77 in the case of the sentence vector. The baseline TF-IDF performs poorly with 0.52 ROC AUC. In this case, comparing the test set with the vector of the sentences gave better results than with the words. That could be explained in the sense that, even though we filtered stopwords out and we just studied words tagged as nouns, there were more elements in the training set than in a list of two hundred of words. Thus, the orientation of the mean vector was better set to further comparison. The results we achieved were not outstanding as seen in the other systems, but we considered them acceptable as they got closer to 0.80 in the case of the sentences. Note that this was a very naïve algorithm simply based on trying to exploit the hypothetical semantical/directional bias in the word embeddings spaces.

From the results illustrated in Table 5 below (considering Accuracy and F1-Scores), we could appreciate that the TF-IDF baseline performs poorly in all cases. However, it was interesting to appreciate the good performance of MF, who used a TF-IDF and LR system and achieved the best Accuracy and F1-Scores. In the case of the proposed algorithm, we could observe that the use of the mean vector of the words list worked better than the use of the sentences mean vector. In both cases, the Accuracy was high, but that could be due to the unbalancing of the dataset, and the number of not toxic instances is greater than the toxic ones. In the case of F1-Scores (F1-Score for the toxic category, and Macro- and Micro-F1), we could appreciate that for the *toxic* category, the system did not perform well, going under 0.50 in the case of the word vector and under 0.40 in the case of the sentence vector. In the case of Macro- and Micro-F1, the results were not far from the other systems, even though in the case of the words we could still appreciate better results and, in the case of the Micro-F1 our system outperformed the other systems with the mean vector of words with a value of 0.86.

These results illustrated that the use of the mean vector of the sentences of the training set, even if it was with a simple linear word embedding composition function as global average, could be of great use and a good step to consider for systems that study task of toxicity detection. On its own, it showed acceptable ROC AUC results but did not perform that well on the F1-Score of the toxic category. However, we could see competitive results in the case of Macro- and Micro-F1 values.

Systems	Accuracy	F1-Score	Micro-F1	Macro-F1
MF	0.95155	0.77028		
AR	0.93178	0.64925		
AKM	0.92924		0.81181	**0.71353**
K&H	**0.99**	0.66		
VA		**0.79**		
TF-IDF	0.60	0.17	0.59731	0.50887
Wansun	0.86	0.46	**0.85651**	0.65903
Sansun	0.74	0.38	0.74392	0.61153

Table 5. Mean vector proximity results using Accuracy and F1-Scores

Table 6 down compilated the results from our systems in all the evaluation metrics. Here we could appreciate that the best system was Wansun, the one that uses the words list. This could be because we had a closed list of toxic words, which made a greater difference in contrast to the toxic sentences, which may include neutral words. However, we could appreciate that the F1-Score of the *toxic* category was lower than 0.50, which may indicate that the systems were not classifying correctly toxic content in this case.

Experiment 1	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.52	0.60	0.17	0.60	0.51
Wansun	0.75	**0.86**	**0.46**	**0.86**	**0.66**
Sansun	**0.77**	0.74	0.38	0.74	0.61

Table 6. Results with Kaggle dataset (Experiment 1)

In addition, we would like to observe the Recall values of our systems and the only reported Recall metric observed in the literature review, the one from Van Aken et al. (2018) (VA), who also used the Kaggle dataset, as we considered that the most important metric in toxicity evaluation should be Recall. In terms of Recall we observed that our proposed systems did not perform so badly. In particular, we performed over 0.80 Recall values in the case of Sansun, while VA scored a Recall of 0.88. The results can be observed in Table 7 below.

Systems	*Recall*
TF-IDF	0.43
Wansun	0.62
Sansun	0.81
VA	**0.88**

Table 7. Results using Recall (Experiment 1)

As none of our systems relevantly outperformed the other ones, that was why we decided to only perform this comparison in the Kaggle dataset. For the other datasets (OLID and UCC), we will just consider our TF-IDF baseline and the proposed algorithm, Asnsun, in its both versions: Wansun in the case of the words, and Sansun in the case of the sentences.

- *OLID dataset*

After performing the experiment in the OLID dataset, we appreciated that Sansun achieved the best results among the models. The biggest differences were found in Accuracy and Micro-F1, where Ansun models both achieved values of 0.70 and 0.71 respectively, while the TF-IDF model only scored around 0.50. The rest of the values were closer to each other, around 0.45-0.50 in the case of TF-IDF and 0.50-0.65 in the case of Ansun. These results illustrated in Table 8 showed the positive performance of the proposed algorithm in the case of the sentences in Experiment 1 with the OLID dataset.

Experiment 1	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.50	0.47	0.37	0.47	0.50
Wansun	0.58	0.70	0.36	0.70	0.60
Sansun	**0.64**	**0.71**	**0.49**	**0.71**	**0.65**

Table 8. Results with OLID dataset (Experiment 1)

- *UCC dataset*

In the case of the UCC dataset, we could appreciate similar results as the ones in OLID in metrics such as ROC AUC or Macro-F1, where Sansun outperformed the rest of the systems. In the case of F1-Score of the toxic category, the results were lower than in the previous dataset but similar to the results from the Kaggle dataset. Finally, it was interesting to note that the Accuracy and Micro-F1 results were outperformed in the case of Wansun, which we could also observe in the Kaggle dataset. The results from the UCC dataset are illustrated in Table 9.

Experiment 1	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.51	0.71	0.16	0.71	0.50
Wansun	0.52	**0.77**	0.17	**0.77**	0.52
Sansun	**0.58**	0.58	**0.25**	0.58	**0.54**

Table 9. Results with UCC dataset (Experiment 1)

As a conclusion, we could observe that in OLID Sansun outperformed the rest of the systems, meanwhile in the case of UCC, we could appreciate a competition between Wansun and Sansun. This dichotomy between systems demonstrated that both approaches still have potential for further development and how the characteristics of the dataset affected in each case. A reasoning behind the differences between Kaggle and UCC in contrast to OLID, could be the type of the data. Both Kaggle and UCC were post comments, while OLID consisted of tweets, which were more consistent in form and length and, thus why Sansun outperforms in all metrics in OLID. In the case of Kaggle and UCC, we could explain the inconsistency of lengths and topics due to their nature, that was why we could observe that both Sansun and Wansun competed in best results.

4.2. Experiment 2: Orientation proximity

This second experiment consisted of evaluating whether a new sentence vector fitted inside the cone made by the mean vector of toxic words and toxic sentences. This means that we developed a classifier based on the direction hypothesis, thus, all toxic content would fit in a closed cone around the median direction within the representation space.

For that, we started performing the same process as with the previous experiment. In the test set, we first obtained the vectors of the words that compound the sentence with our TF-IDF baseline and with Word2Vec to further apply the *f_ga* composition function to obtain the sentence vector. Once we obtained the sentence vector of the new sentences, we calculated its cosine similarity with the mean vector of the toxic words and the toxic sentences. In this experiment we took in consideration the mean cosine of all toxic words, which was 0.30 for the proposed algorithm; and the mean cosine of all the toxic sentences, which was 0.14 in the case of the TF-IDF baseline and a bit over 0.30 in the case of the proposed algorithm. As we wanted to verify if the cosine from a new sentence fitted inside the cone, we needed to consider the value of half of the cone we have. As say, if the mean cosine was 0.30 for the toxic words, we would consider 0.60 as "half the cone", which would go for both sides of the mean cosine of the toxic words. For the sentences, the cosine was 0.14 and slightly over 0.30, thus, we considered 0.30 and 0.65 so, once more, we gave angle margin on both sides of the cone.

Once we established the values, we calculated the cosine similarity between the vector of each new sentences with the mean vector of the words and the mean vector of the sentences: in the case of comparing with the mean word vector, when the cosine similarity was greater than 0.60 we would assign that sentence with the label *toxic* (as the similarity was inside one of the sides of the centre of the cone), otherwise we considered it *not toxic*. The same process was followed for the sentences, but in this case the sentence was labelled as *toxic* when the cosine similarity between the mean vector of the toxic sentences and the vector of the new sentence went over 0.30 or 0.65, otherwise, it was considered *not toxic*.

Results

After performing the experiment on the Kaggle dataset, we evaluated the model with the same evaluation metrics as the previous one: ROC AUC, Accuracy and F1-Score. ROC AUC was used to compare the results with the top best performances of the challenge, and the rest of the metrics to evaluate with the rest of the systems. As noted earlier, one of the systems also evaluated with ROC AUC was included.

The results of the orientation proximity to the mean cosine of toxic words are showed in Table 10, in addition to the results of the rest of the systems evaluated with ROC AUC. In Table 11, we presented the results compared with the systems evaluated with Accuracy and F1-Scores.

Systems	ROC AUC
Toxic Crusaders	**0.98856**
neongen & Computer says no	0.98822
Adversarial Autoencoder	0.98805
MF	0.959021
TF-IDF	0.46887
Wansun	0.65933
Sansun	0.61831

Table 10. Orientation proximity results using ROC AUC

Systems	Accuracy	F1-Score	Micro-F1	Macro-F1
MF	0.95155	0.77028		
AR	0.93178	0.64925		
AKM	0.92924		**0.81181**	0.71353
K&H	**0.99**	0.66		
VA		**0.79**		
TF-IDF	0.47	0.05	0.80266	0.47207
Wansun	0.88	0.39	0.66160	**0.88124**
Sansun	0.77	0.27	0.56134	0.76954

Table 11. Orientation proximity results using Accuracy and F1-Scores

As it could be appreciated in Table 10 our results were lower than in the previous proposal. We scored 0.66 ROC AUC with Wansun, when the algorithm used the word vector; and 0.47 (TF-IDF) and 0.62 (Sansun) when we used the sentence vector. Contrary than before, in the case of the proposed algorithm, we achieved better results when we applied the mean vector of the words list to calculate the cosine similarities with the test set. This could be explained as the list of words was more limited than the list of sentences, thus cosine similarities were calculated with less variance. So, in the moment of comparing new sentences, the cosine similarity was more stable with the word vector.

From the results in Table 11, we could again observe that the mean vector of the words was higher than the sentence vector in the case of our proposals. If we focused on the proposed algorithm, we could appreciate that the Accuracy was still high, over 0.85 in the case of the word vector and over 0.75 in the case of the sentence vector. However, again we must take in consideration the unbalancing of the dataset. In the case of F1-Scores, we could appreciate that for the toxic category the system performed worse: under 0.40 in the case of words and under 0.30 for the sentences. For the Macro- and Micro-F1, we could appreciate that the Micro-F1 worsened (outperformed by the TF-IDF baseline), but there was an improvement in Macro-F1 where our system outperformed in both cases the other systems' results. The best result was observed using the word vector scoring 0.88 Macro-F1.

These results illustrated that the use of the mean vector of the words could be useful for systems that studied the task of toxicity detection with orientation parameters. However, the results were a bit worse than in the previous experiment regarding the ROC AUC metric. On the other hand, it showed acceptable results in Macro-F1, even though it did not perform that well on the F1-Score of the toxic category and on Micro-F1. The results of our systems are compilated in Table 12:

Experiment 2	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.47	0.80	0.05	**0.80**	0.47
Wansun	**0.66**	**0.88**	**0.39**	0.66	**0.88**
Sansun	0.62	0.77	0.27	0.56	0.77

Table 12. Results with Kaggle dataset (Experiment 2)

In terms of Recall we could observe that our proposed systems performed quite poorly. We appreciated that our highest Recall was achieved by Sansun (not even reaching 0.50). In contrast, we could remember the good result from VA, who scored 0.88. The results are presented in Table 13 below:

Systems	***Recall***
TF-IDF	0.05
Wansun	0.38
Sansun	0.43
VA	**0.88**

Table 13. Results using Recall (Experiment 2)

Next, we performed the experiment in the other two datasets: OLID and UCC; and present the results for our systems: TF-IDF, Wansun and Sansun.

- *OLID*

In the case of OLID, we performed the TF-IDF division at 0.32, the double of the mean cosine of the toxic sentences (0.16). For Wasun, we established the division at 0.60, the double of the mean cosine of the toxic list of words (0.30), and at 0.40 for Sansun, as it was the double of the mean cosine of the toxic sentences (around 0.19). In this case, we observed that Wansun outperformed in most of the metrics, and it was just outperformed by Sansun in terms of F1-Score for the toxic category. That showed us, that in the case of cosine orientation it was more efficient to check the mean cosine of a list of word, probably as it was a more defined and closed list than a set of sentences, which could have bigger semantic space. The results are illustrated in Table 14:

Experiment 2	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.49	0.69	0.06	0.69	0.46
Wansun	**0.56**	**0.73**	0.27	**0.73**	**0.64**
Sansun	0.53	0.37	**0.44**	0.37	0.54

Table 14. Results with OLID dataset (Experiment 2)

- *UCC*

In the case of UCC dataset, for the words the metrics are established at 0.60 in the case of Wansun, the same as before. In the case of the sentences, the TF-IDF limit was established at 0.30, and at 0.32 for Sansun, as those were the valuesof the double of the mean cosine of the sentences in the training set. In this case, there was a clear competition between Sansun and TF-IDF. It was explicit the superiority of using sentences over words, even though, TF-IDF outperformed in Accuracy and Micro-F1, and Sansun outperformed in ROC AUC, F1-Score (toxic category) and Macro-F1. This difference could be divided as TF-IDF mostly classifies the sentences as *not toxic*, that was why there was such a big Accuracy value, but the F1-Score of the toxic category was almost inexistent. Thus, the classification performed by Sansun, even if it was not showing amazing results, could be a better starting point from which for further research in classification. The results can be observed in Table 15:

Experiment 2	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.49	**0.85**	0.02	**0.85**	0.46
Wansun	0.50	0.84	0.08	0.84	0.50
Sansun	**0.52**	0.21	**0.22**	0.21	**0.52**

Table 15. Results with UCC dataset (Experiment 2)

4.3. Experiment 3: Nearest Neighbours

This third experiment consisted of evaluating whether a new sentence vector would be closer to the vectors of toxic words and sentences, or to the vectors of not toxic words and sentences.

First, we needed to balance the Kaggle dataset. For that, we extracted 16226 toxic and not toxic sentences, the value was the number of total toxic sentences in the training set. Next, we calculated all the sentence vectors from the training set of both toxic and not toxic words, and of the toxic and not toxic sentences. We then fitted the resulted vectors in a KNeighboursClassifier[34] from sklearn. For the test set, we extracted the vectors of the words that compound the sentence with our Word2Vec model to further apply our composition function (*f_ga*) and obtain the sentence vector. Once we obtained the sentence vector of the new sentences, we used the vectors as the test set to predict with the algorithm. We tested with different number of nearest neighbours (1, 3, 5, 7, 11, 21, 51, 101…), and if the greatest number of neighbours were from the toxic words or sentences, the new sentence was labelled as *toxic*; on the other hand, if the most number of neighbours were from the not toxic words or sentences, the new sentence was labelled as *not toxic*.

Results

Once more, we evaluated the algorithms with ROC AUC to compare the top best systems; and Accuracy and F1-Scores, to compare the rest of the systems.

The results of the nearest neighbours algorithm on the Kaggle dataset using ROC AUC are illustrated in Table 16, and the results of the other evaluation metrics (Accuracy and F1-Scores) are shown in Table 17. For illustrating the results, we selected the value n = 21, which seemed best one to choose from the different values tested.

[34] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

Systems	ROC AUC
Toxic Crusaders	**0.98856**
neongen & Computer says no	0.98822
Adversarial Autoencoder	0.98805
MF	0.959021
TF-IDF	0.57849
Wansun	0.68432
Sansun	0.78777

Table 16. Nearest Neighbours results using ROC AUC

As it could be appreciated in Table 16, we obtained 0.68 ROC AUC with Wansun (word vector), and 0.58 with TF-IDF and 0.79 with Sansun (sentence vector). In this case, while considering our proposed algorithm, we achieved better results with the sentence vectors. Probably the composition function with which we obtained the sentences' vectors was the most detailed one and thus it was reflected in best results.

From the results in Table 17, we can observe that we achieved better results with the sentence vectors, with 0.75 Accuracy, 0.40 F1-Score, and 0.75 and 0.62 Micro- and Macro-F1 respectively. In the case of the proposed algorithm with word vectors, the results were poorer than with the sentences. These results illustrated that the use of vectors of toxic sentences as nearest neighbours could be of good use for the task of toxicity detection, even though it still had some space of improvement. Our four approaches' results are presented in Table 18:

Systems	Accuracy	F1-Score	Micro-F1	Macro-F1
MF	0.95155	0.77028		
AR	0.93178	0.64925		
AKM	0.92924		**0.81181**	**0.71353**
K&H	**0.99**	0.66		
VA		**0.79**		
TF-IDF	0.69	0.22	0.68737	0.53282
Wansun	0.53	0.27	0.53494	0.56530
Sansun	0.75	0.40	0.75452	0.61853

Table 17. Nearest Neighbours results using Accuracy and F1-Scores

Experiment 3	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.58	0.69	0.22	0.69	0.53
Wansun	0.68	0.53	0.27	0.53	0.56
Sansun	**0.79**	**0.75**	**0.40**	**0.75**	**0.62**

Table 18. Results with Kaggle dataset (Experiment 3)

In terms of Recall we could observe that our proposed systems performed very well. We observed that our highest Recall value (0.87) was achieved by Wansun. In addition, Sansun performed over 0.80 as well, with 0.83 Recall. The results from Wansun were not far from the ones from VA, who scored 0.88 Recall. The results can be observed below in Table 19:

Systems	***Recall***
TF-IDF	0.44
Wansun	0.87
Sansun	0.83
VA	**0.88**

Table 19. Results using Recall (Experiment 3)

Finally, we experimented with the rest of the datasets, OLID and UCC, and comparing our systems.

- *OLID*

For this experiment we also had to balance the dataset down to 8800 sentences, divided as 4400 toxic and 4400 not toxic ones. As it could be appreciated from the Kaggle experimentation, we continued testing with the value n = 21 of neighbours.

The results followed the same scores as appreciated before. The TF-IDF system outperformed in Accuracy and Micro-F1 score. However, we it was not far from the proposed models, and Sansun even outperformed in the other metrics (as previously observed in Kaggle as well). The results were similar to the Kaggle dataset in Macro-F1 and F1-Score in the toxic category, but the ROC AUC was slightly lower. The results are illustrated in Table 20.

Experiment 3	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.51	**0.60**	0.31	**0.60**	0.51
Wansun	0.58	0.51	0.46	0.51	0.58
Sansun	**0.62**	0.56	**0.49**	0.56	**0.60**

Table 20. Results with OLID dataset (Experiment 3)

- *UCC*

In the case of the UCC dataset, the data reduction was made to 4077 *toxic* and *not toxic* sentences to balance the dataset. Again, the results followed the same pattern as seen. TF-IDF outperformed in Accuracy and Micro-F1, meanwhile Sansun outperformed in ROC AUC, F1-Score (toxic category) and Macro-F1. The best classification approach was achieved by the Sansun algorithm. In this dataset, the results were slightly lower than the ones earlier, even though the ROC AUC was still around 0.60. These results are presented in Table 21.

Experiment 3	**Evaluation metrics**				
Systems	***ROC AUC***	***Accuracy***	***F1-Score***	***Micro-F1***	***Macro-F1***
TF-IDF	0.51	**0.74**	0.16	**0.74**	0.51
Wansun	0.57	0.51	0.24	0.51	0.53
Sansun	**0.58**	0.47	**0.25**	0.47	**0.54**

Table 21. Results with UCC dataset (Experiment 3)

As a conclusion, it could be appreciated that in all cases TF-IDF outperformed in two of the evaluation metrics. However, in the case of Sansun in the Kaggle dataset, we could observe the best ROC AUC value from all experiments and systems: 0.79, which was already a great result. As seen, although there was still space for improvement in the case of the toxic category, the results of our proposed Ansun systems were promising in terms of what we claimed in our study: not to use them as stand-alone algorithms for classification but, instead, as a space to learn and improve in combination with other algorithms, such as DL or BERT models.

4.4. Conclusions

As it could be appreciated from the experiments, considering semantic orientation and linguistic structure seemed relevant. In Experiment 1, *Mean vector proximity*, we appreciated that although the best ROC AUC results were achieved with Sansun, the rest of the evaluation metrics alternated between Sansun and Wansun. In Experiment 2, *Orientation proximity*, we could observe more variation in the results. In the Kaggle and OLID dataset almost all the results were outperformed by Wansun, while in UCC dataset there was a competition between TF-IDF and Sansun. In Experiment 3, *Nearest Neighbours*, Sansun outperformed in Kaggle, while there was a rigid competition between TF-IDF and Sansun along the other datasets. However, the best values and results after a more detailed observation were achieved by our Sansun algorithm. Also, in this case in the Kaggle dataset, we achieved the best ROC AUC score, which went up to 0.79. The best Recall was achieved by Wansun in Experiment 3, where it scored 0.87 Recall, even though Sansun was not far with 0.83 Recall.

That was an interesting result in comparison with the system from Van Aken et al. (2018), who scored 0.88 Recall with an ensemble of GBDT.

These results illustrated that there was indeed a toxic orientation bias within the semantic representation space of Word2Vec. In more detail, using a finite list of words considered *toxic* and a list of words considered *not toxic* differentiated well the orientation of the toxicity in the messages. Meanwhile, in the case of the sentences, there were elements that could be considered *not toxic*, as a sentence was formed by words that did not all fall into that category. The reason behind this result could be that a list of *toxic* words had elements only considered toxic, thus, the orientation of the vectors was purely guided by the toxic elements. Even so, our results supported our hypothesis that there existed bias towards toxic content inside the Word2Vec semantic space.

Finally, it was interesting to note that Sansun outperforms Wansun in many of the metrics. This could help us understand that, even though some of the results showed similar performance between Wansun and Sansun, most of the best values were achieved with Sansun. As a sum up, we could appreciate that, overall, TF-IDF achieved the best value in 20% of the cases, Wansun in the 30% of the cases, and Sansun outperformed in 50% of the cases. This illustrated the general outperformance among experiments and datasets of Sansun.

5. Conclusions and Future work

In this chapter, we summarized the findings and proposed lines of future work.

5.1. Conclusions

The field of toxicity detection, in addition to detection of other toxic or concerning behaviours online has significantly advanced in the recent years, thanks to deep learning and Transformer-based models. Yet, although those models obtain outstanding results, they are black boxes that do not allow us directly to know why they decide or predict a message as *toxic* or *not toxic* the way they do. In this sense, classical machine learning models and Word Embeddings, still give us crucial information to advance in the study of toxicity detection online. Furthermore, Static Word Embeddings maintain the semantic properties of the meaning of the words, which is not as clear in deep learning models, such as BERT. In this context, we defend that there is still a need for further research in classical models or Word Embeddings to give light and show explainability and interpretability to the results of the models and further improve the results of deep learning models as well.

In terms of semantic understanding, this thesis has made two contributions, which answer the two RQ asked: 1) the study the semantic orientation of the toxic messages to verify if toxic messages have a semantic orientation bias in the semantic vector space, and 2) the study if whether there exists something such as a linguistic toxic structure, meaning that toxic messages present a more particular structure from which we can try to detect further toxicity.

In relation to the first part, we explored the semantic orientation of toxic messages in the semantic vector space of Word2Vec, considering that toxic messages would present an orientation or direction bias within such space. We performed experiments on vector proximity (mean vector and nearest neighbours) and orientation proximity, where we verified 1) if whether a new sentence would be closer to the mean vector of a list of toxic words and a set of toxic sentences, 2) if the cosine similarity between the vector of a new sentence and the mean vector from the list of toxic words or sentences, would fit inside the cone of similarities, 3) if the vector of a new sentence would be closer to the vectors of the list of toxic or not toxic words or to the toxic or not toxic sentences. This allowed us to check if we could predict the toxicity of new sentences using semantic proximity. *The results showed that there was indeed a semantic orientation bias in toxic messages* and, even though our algorithm did not outperform current Deep Learning systems (which we already assumed since the beginning), it could be helpful as a feature to consider and add to those systems to improve results in this kind of tasks. Also, this was a light and easy to interpret model, which could be used and implemented in any device, as well as being accessible without the need of large computational systems or data.

Regarding the second part, we explored if the linguistic structure was a relevant component for detecting toxic messages. In particular, if toxic messages had an inherent linguistic structure that could give us clues to further detect new toxic messages. We performed experiments with different linguistic structures, such as sentence order: we performed sentence composition to obtain sentence vectors in a sequential, recursive left-to-right (L2R) and recursive right-to-left (R2L) orders. For the sequential order we used the well-known *f_ga* (*global average*) composition function, and for the L2R and R2L orders we used the *f_inf* function, however all orders seemed to perform equally. These results corroborate previous insights in compositional distributional semantics (Amigó et al., 2022). We also experimented with and without the use of stopwords: we verified that stopwords had some inherent semantic orientation bias, thus they would influence the direction of toxic messages. The main point was to check if that influence was relevant for toxicity detection, which resulted in irrelevant information to consider in our case, and that was why we performed the analysis of our algorithm without stopwords. Finally, we considered grammatical categories to prove if there were more relevant categories that helped detect toxic messages: we performed POS tagging and calculated the cosine similarities of the training sentences to verify the influence in the results. After checking the four main grammatical categories (Nouns, Verbs, Adjectives and Adverbs), we could observe that *Nouns were the category that most importance showed for the detection of toxicity in messages*.

As a final conclusion, our algorithm, which was built with a sequential composition function without considering stopwords and using just words tagged as nouns, was able to show acceptable results in terms of ROC AUC. In the case of F1-Score, our model did not perform as good, but in the case of Micro- and Macro-F1, we obtained interesting results worth considering in further investigations. Finally, in terms of Recall we could appreciate that in some cases the proposed algorithm achieved high results that were competitive with the results obtained by the literature.

5.2. Future work

We believe that our study offers room for improvement, especially in relation to the methodology and experimentation.

First, we considered that our results illustrated an interesting possibility for improvement applied to DL. We could specifically use the information obtained so that DL models paid attention to semantic orientation in the tasks of toxicity detection. This would allow interpretability in the systems' results.

Another aspect would be to investigate more forms of linguistic structure. In our study we focused in morphological categories and in sequential and lineal orders. Thus, it would be interesting to apply more detailed syntactic structures such as constituents or dependencies. Using the composition functions in this sense would allow us to take in consideration the most important elements of the sentence and their relatedness and compose the sentence vector following such more complex orders. We considered that using the *f_inf* composition function with these other structures would result in different and probably more improved results as it contemplated the natural parsing of the language. Another interesting aspect that fell out of the scope of this thesis was the use of quirky elements, such as negation. We could observe the complex nature of negation in other NLP and linguistic studies and, since negation is still a problematic element, it would be interesting to perform further studies focusing on it to explore how it affects the detection of toxic content.

References

Abderrouaf, C., & Oussalah, M. (2019, December). On online hate speech detection. effects of negated data construction. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 5595-5602). IEEE.

Aggarwal, C. C., & Zhai, C. (2012). A survey of text classification algorithms. In *Mining text data* (pp. 163-222). Springer, Boston, MA.

Ahluwalia, R., E. Shcherbinina, E. Callow, A. Nascimento, M. De Cock. (2018, September). Detecting Misogynous Tweets. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018). CEUR Workshop Proceedings. CEUR-WS.org, Seville, Spain.

Aldana-Bobadilla, E., Molina-Villegas, A., Montelongo-Padilla, Y., Lopez-Arevalo, I. and S. Sordia, O. (2021). A Language Model for Misogyny Detection in Latin American Spanish Driven by Multisource Feature Extraction and Transformers. Appl. Sci. 2021, 11, 10467.

Amigó, E., Ariza, A., Fresno, V., & Martí, M. A. (2022). Information-Theoretic Compositional Distributional Semantics. (*To be published*).

Aragón, M. E., Jarquín-Vásquez, H. J., Montes-y-Gómez, M., Escalante, H. J., Pineda, L. V., Gómez-Adorno, H., Posadas-Durán, J. P., & Bel-Enguix, G. (2020, September). Overview of MEX-A3T at IberLEF 2020: Fake News and Aggressiveness Analysis in Mexican Spanish. In *IberLEF@ SEPLN* (pp. 222-235).

Arora, S., Li, Y., Liang, Y., Ma, T., & Risteski, A. (2016). A latent variable model approach to pmi-based word embeddings. *Transactions of the Association for Computational Linguistics, 4*, 385-399.

Arora, S., Liang, Y., & Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.

Astoveza, G., Obias, R. J. P., Palcon, R. J. L., Rodriguez, R. L., Fabito, B. S., & Octaviano, M. V. (2018, October). Suicidal behavior detection on twitter using neural network. In *TENCON 2018-2018 IEEE Region 10 Conference* (pp. 0657-0662). IEEE.

Badjatiya, P., Gupta, S., Gupta, M., & Varma, V. (2017, April). Deep learning for hate speech detection in tweets. In *Proceedings of the 26th international conference on World Wide Web companion* (pp. 759-760).

Basile, V., C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso & M. Sanguinetti. (2019). SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *Proceesings of the 13th International Workshop on Semantic Evaluation*. Minneapolis, Minnesota, USA. Association for Computational Linguistics. Pp. 54-63.

Bhattacharya, S., Singh, S., Kumar, R., Bansal, A., Bhagat, A., Dawer, Y., Lahiri, B., & Ojha, A. K. (2020). Developing a multilingual annotated corpus of misogyny and aggression. *arXiv preprint arXiv:2003.07428*.

Blacoe, W., & Lapata, M. (2012, July). A comparison of vector-based representations for semantic composition. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 546-556).

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching word vectors with subword information. Transactions of the association for computational linguistics, 5, 135-146.

Boleda, G. (2019). Distributional semantics and linguistic theory. *arXiv preprint arXiv:1905.01896*.

Britto, B. K., & Khandelwal, A. (2020). Resolving the scope of speculation and negation using transformer-based architectures. *arXiv preprint arXiv:2001.02885*.

Cañete, J., Chaperon, G., Fuentes, R., Ho, J. H., Kang, H., & Pérez, J. (2020). Spanish pre-trained bert model and evaluation data. *Pml4dc at iclr, 2020*, 1-10.

Cellier, P. (2019). Machine Learning and Knowledge Discovery in *Databases: International Workshops of ECML PKDD 2019*, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I. Springer Nature.

Chatzakou, D., Kourtellis, N., Blackburn, J., De Cristofaro, E., Stringhini, G., & Vakali, A. (2017, June). Mean birds: Detecting aggression and bullying on twitter. In *Proceedings of the 2017 ACM on web science conference* (pp. 13-22).

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2019). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Coppersmith, G., Leary, R., Whyne, E., & Wood, T. (2015, August). Quantifying suicidal ideation via language usage on social media. In *Joint statistics meetings proceedings, statistical computing section, JSM* (Vol. 110).

Cumalat Puig, Eudald. (2020). Sentiment analysis on short Spanish and Catalan texts using contextual word embeddings. MS thesis. Universitat Politècnica de Catalunya.

Dadvar, M., Trieschnigg, D., Ordelman, R., & Jong, F. D. (2013, March). Improving cyberbullying detection with user context. In *European Conference on Information Retrieval* (pp. 693-696). Springer, Berlin, Heidelberg.

Davidson, T., Warmsley, D., Macy, M., & Weber, I. (2017). Automated hate speech detection and the problem of offensive language. Proceedings of the eleventh international conference on web and social media, AAAI (pp. 512-515).

De Gibert, O., Perez, N., García-Pablos, A., & Cuadros, M. (2018). Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*.

Dessì, D., Dragoni, M., Fenu, G., Marras, M., & Recupero, D. R. (2019, April). Evaluating neural word embeddings created from online course reviews for sentiment analysis. In *Proceedings of the 34th ACM/SIGAPP symposium on applied computing* (pp. 2124-2127).

Dessì, D., Helaoui, R., Kumar, V., Recupero, D. R., & Riboni, D. (2020). TF-IDF vs word embeddings for morbidity identification in clinical notes: An initial study. *arXiv preprint arXiv:2105.09632*.

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

Espinal, M.T. (1992). Expletive negation and logical absorption. *The Linguistic Review, 9*, 333–358.

Ethayarajh, K. (2019). How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics. - *arXiv preprint arXiv:1909.00512.*

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., & Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. *the Journal of machine Learning research, 9*, 1871-1874.

Fersini, E., Nozza, D., & Rosso, P. (2018a). Overview of the Evalita 2018 task on automatic misogyny identification (AMI). *EVALITA Evaluation of NLP and Speech Tools for Italian, 12*, 59.

Fersini, E., Rosso, P., & Anzovino, M. (2018b). Overview of the Task on Automatic Misogyny Identification at *IberEval 2018. Ibereval@ sepln, 2150*, 214-228.

Founta, A. M., D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali & I. Leontiadis. (2018). A Unified Deep Learning Architecture for Abuse Detection. n: Proceedings of the 10th ACM Conference on Web Science, WebSci '19, pp. 105–114. ACM, New York, NY, USA.

Frenda, S., B. Ghanem, M. Montes-y-Gómez. (2018, September). Exploration of Misogyny in Spanish and English tweets. In: Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018), co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018). CEUR Workshop Proceedings. CEUR-WS.org, Seville, Spain.

Friedman, J. (2001). Greedy boosting approximation: a gradient boosting machine. Ann. Stat. 29, 1189–1232. doi: 10.1214/aos/1013203451.

Fuster, A. (2022, September). MS Thesis. Máster Universitario en Investigación en Inteligencia Artificial. Universidad Nacional de Educación a Distancia (UNED).

Fuster, A., & Fresno, V. (2022). Is anisotropy really the cause of BERT embeddings not being semantic? Sent to *The 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (*To be published*).

Gao, L., & Huang, R. (2017). Detecting online hate speech using context aware models. *arXiv preprint arXiv:1710.07395*.

Gao, J., He, D., Tan, X., Qin, T., Wang, L., & Liu, T. Y. (2019). Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*.

García-Díaz, J. A., Cánovas-García, M., Colomo-Palacios, R., & Valencia-García, R. (2021). Detecting misogyny in Spanish tweets. An approach based on linguistics features and word embeddings. *Future Generation Computer Systems*, *114*, 506-518.

Gharbi, S., Arfaoui, H., Haddad, H., & Kchaou, M. (2021). TEET! Tunisian Dataset for Toxic Speech Detection. *arXiv preprint arXiv:2110.05287*.

Guest, E., Vidgen, B., Mittos, A., Sastry, N., Tyson, G., & Margetts, H. (2021, April). An expert annotated dataset for the detection of online misogyny. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*: Main Volume (pp. 1336-1350).

Henry, S., Buchan, K., Filannino, M., Stubbs, A., & Uzuner, O. (2020). 2018 n2c2 shared task on adverse drug events and medication extraction in electronic health records. *Journal of the American Medical Informatics Association, 27*(1), 3-12.

Ho, T. K. (1995, August). Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition* (Vol. 1, pp. 278-282). IEEE.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks,* 2(5), 359-366.

Hosseini, H., Kannan, S., Zhang, B., & Poovendran, R. (2017). Deceiving google's perspective api built for detecting toxic comments. *arXiv preprint arXiv:1702.08138.*

Jackendoff, R. S. (1969). An interpretive theory of negation. *Foundations of language,* 218-241.

Jensen, M. H. (2020). Detecting hateful utterances using an anomaly detection approach. Master’s thesis, Norwegian University of Science and Technology (NTNU).

Ji, S., Yu, C. P., Fung, S. F., Pan, S., & Long, G. (2018). Supervised learning for suicidal ideation detection in online user content. *Complexity, 2018.*

Ji, S., Pan, S., Li, X., Cambria, E., Long, G., & Huang, Z. (2020). Suicidal ideation detection: A review of machine learning methods and applications. *IEEE Transactions on Computational Social Systems, 8*(1), 214-226.

Jiang, T., Huang, S., Zhang, Z., Wang, D., Zhuang, F., Wei, F., Huang, H., Zhang, L., & Zhang, Q. (2022). PromptBERT: Improving BERT Sentence Embeddings with Prompts. *arXiv preprint arXiv:2201.04337*.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Karan, M., & Šnajder, J. (2018, October). Cross-domain detection of abusive language online. In *Proceedings of the 2nd workshop on abusive language online (ALW2)* (pp. 132-137).

Khandelwal, A., & Sawant, S. (2019). NegBERT: a transfer learning approach for negation detection and scope resolution. *arXiv preprint arXiv:1911.04211*.

Klima, E. (1964). Negation in English. In J. A. Fodor & J. J. Katz, eds. *The Structure of Language*, 246–323. Englewood Cliffs. Prentice-Hall.

Kolhatkar, V., Wu, H., Cavasso, L., Francis, E., Shukla, K., & Taboada, M. (2019). The SFU opinion and comments corpus: A corpus for the analysis of online news comments. Corpus Pragmatics, (pp. 1-36).

Koratana, A., & Hu, K. (2018). Toxic Speech Detection. *URL: https://web. stanford. edu/class/archive/cs/cs224n/cs224n, 1194*.

Kovács, G., Alonso, P., & Saini, R. (2021). Challenges of hate speech detection in social media. *SN Computer Science, 2*(2), 1-15.

Kumar, R., Ojha, A. K., Malmasi, S., & Zampieri, M. (2018a). Benchmarking aggression identification in social media. In Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018) (pp. 1-11)

Kumar, R., Reganti, A. N., Bhatia, A., & Maheshwari, T. (2018b). Aggression-annotated corpus of hindi-english code-mixed data. *arXiv preprint arXiv:1803.09402*.

Kweon, S., Kim, Y., Jang, M. J., Kim, Y., Kim, K., Choi, S., Chun, C., Khang, Y. H., & Oh, K. (2014). Data resource profile: the Korea national health and nutrition examination survey (KNHANES). *International journal of epidemiology, 43*(1), 69-77.

Lample, G., & Conneau, A. (2019). Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291.*

Lenci, A. (2018). Distributional models of word meaning. *Annual review of Linguistics, 4,* 151-171.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Advances in neural information processing systems, 27,* 2177-2185.

Li, B., Zhou, H., He, J., Wang, M., Yang, Y., & Li, L. (2020). On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* pages 9119–9130, Online. Association for Computational Linguistics. - *arXiv preprint arXiv:2011.05864.*

Li, K. (2021). HAHA at EmoEvalEs 2021: Sentiment Analysis in Spanish Tweets with Cross-lingual Model. In *IberLEF@ SEPLN* (pp. 49-58).

Liang, Y., Cao, R., Zheng, J., Ren, J., & Gao, L. (2021, September). Learning to remove: Towards isotropic pre-trained BERT embedding. In *International Conference on Artificial Neural Networks* (pp. 448-459). Springer, Cham.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692.*

López-Úbeda, P., Plaza-del-Arco, F. M., Díaz-Galiano, M. C., Lopez, L. A. U., & Martín-Valdivia, M. T. (2019, September). Detecting anorexia in Spanish tweets. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing* (RANLP 2019) (pp. 655-663).

López-Úbeda, P., Plaza-del-Arco, F. M., Díaz-Galiano, M. C., & Martín-Valdivia, M. T. (2021). How Successful Is Transfer Learning for Detecting Anorexia on Social Media?. *Applied Sciences*, 11(4), 1838.

Losada, D. E., & Crestani, F. (2016, September). A test collection for research on depression and language use. In *International Conference of the Cross-Language Evaluation Forum for European Languages* (pp. 28-39). Springer, Cham.

MacAvaney, S., Yao, H. R., Yang, E., Russell, K., Goharian, N., & Frieder, O. (2019). Hate speech detection: Challenges and solutions. *PloS one, 14*(8), e0221152.

Mandl, T., Modha, S., Majumder, P., Patel, D., Dave, M., Mandlia, C., & Patel, A. (2019, December). Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th forum for information retrieval evaluation* (pp. 14-17).

Martí, M. A., Taulé, M., Nofre, M., Marsó, L., Martín-Valdivia, M. T., & Jiménez-Zafra, S. M. (2016). La negación en español: análisis y tipología de patrones de negación. *Procesamiento del Lenguaje Natural*, (57), 41-48.

Martínez-Cámara, E., Díaz-Galiano, M.C., García-Cumbreras, M.A., García-Vega, M., Villena-Román, J. (2017). Overview of TASS 2017. *Proceedings of TASS* (pp. 13-21).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems, 26*.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., & Joulin, A. (2017). Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.

Milne, D. N., Pink, G., Hachey, B., & Calvo, R. A. (2016, June). Clpsych 2016 shared task: Triaging content in online peer-support forums. In *Proceedings of the third workshop on computational linguistics and clinical psychology* (pp. 118-127).

Mina, S., Jaqueline, B., Daria, L., Djordje, S., Armin, K., Manuel, H., Bogensperger, J., Schlarb, S., Schindler, A. & Matthias, Z. (2021). Automatic sexism detection with multilingual transformer models. *arXiv preprint arXiv:2106.04908.*

Mubarak, H., Darwish, K., & Magdy, W. (2017, August). Abusive language detection on Arabic social media. In *Proceedings of the first workshop on abusive language online* (pp. 52-56).

Murgado, J. A. M., Plaza-del-Arco, F. M., López-Ubeda, P., & Martın-Valdivia, M. T. (2021). A Social Monitor for Detecting Inappropriate Behavior. In *Annual Conference of the Spanish Association for Natural Language Processing 2021: Projects and Demonstrations*, pages 41-44. Málaga, Spain, September.

Nemirovski, A., Juditsky, A., Lan, G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization, 19*(4), 1574-1609.

Noble, W. S. (2006). What is a support vector machine?. *Nature biotechnology, 24*(12), 1565-1567.

O'dea, B., Wan, S., Batterham, P. J., Calear, A. L., Paris, C., & Christensen, H. (2015). Detecting suicidality on Twitter. *Internet Interventions, 2*(2), 183-188.

Ophir, Y., Tikochinski, R., Asterhan, C. S., Sisso, I., & Reichart, R. (2020). Deep neural networks detect suicide risk from textual facebook posts. *Scientific reports*, *10*(1), 1-10.

Oževović, G., & Celin, S. (2020). Jigsaw Multilingual Toxic Comment Classification Kaggle Competition. *ACM*. DOI: 10.1145/1235.

Pamungkas, E. W., Cignarella, A. T., Basile, V., & Patti, V. (2018). 14-ExLab@ UniTo for AMI at IberEval2018: Exploiting lexical knowledge for detecting misogyny in English and Spanish tweets. In *3rd Workshop on Evaluation of Human Language Technologies for Iberian Languages, IberEval 2018* (Vol. 2150, pp. 234-241). CEUR-WS.

Pamungkas, E. W., Basile, V., & Patti, V. (2020). Misogyny detection in twitter: a multilingual and cross-domain study. *Information Processing & Management, 57*(6), 102360.

Pappie, Y. (2019). Predicting Online Toxic Content. MS thesis. Vrije Universiteit Amsterdam.

Paraschiv, A. (2020). Detecting Toxic Behavior in Social Media and Online News. MS thesis. University Politehnica of Bucharest.

Partee, B. B., ter Meulen, A. G., & Wall, R. (2012). *Mathematical methods in linguistics* (Vol. 30). Springer Science & Business Media.

Pavlopoulos, J., Malakasiotis, P., & Androutsopoulos, I. (2017). Deep learning for user comment moderation. *arXiv preprint arXiv:1705.09993*.

Pavlopoulos, J., Sorensen, J., Dixon, L., Thain, N., & Androutsopoulos, I. (2020). Toxicity detection: Does context really matter?. *arXiv preprint arXiv:2006.00998*.

Pennebaker, J. W., Boyd, R. L., Jordan, K., & Blackburn, K. (2015). *The development and psychometric properties of LIWC2015*.

Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

Pham, Q. H., Nguyen, V. A., Doan, L. B., Tran, N. N., & Thanh, T. M. (2020, November). From universal language model to downstream task: improving RoBERTa-based Vietnamese hate speech detection. In *2020 12th International Conference on Knowledge and Systems Engineering (KSE)* (pp. 37-42). IEEE.

Pires, T., Schlinger, E., & Garrette, D. (2019). How multilingual is multilingual BERT?. *arXiv preprint arXiv:1906.01502*.

Pitsilis, G. K., Ramampiaro, H., & Langseth, H. (2018). Detecting offensive language in tweets using deep learning. *arXiv preprint arXiv:1801.04433*.

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

Plaza-del-Arco, F. M., Molina-González, M. D., Martin-Valdivia, M. T., & López, L. A. U. (2019, June). SINAI at SemEval-2019 Task 6: Incorporating lexicon knowledge into SVM learning to identify and categorize offensive language in social media. In *Proceedings of the 13th International Workshop on Semantic Evaluation* (pp. 735-738).

Plaza-del-Arco, F. M., Strapparava, C., Lopez, L. A. U., & Martín-Valdivia, M. T. (2020, May). EmoEvent: A multilingual emotion corpus based on different events. In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 1492-1498).

Plaza-del-Arco, F. M., Casavantes, M., Escalante, H. J., Martín-Valdivia, M. T., Montejo-Ráez, A., Montes-y-Gómez, M., Jarquín-Vásquez, H. & Villaseñor-Pineda, L. (2021a). Overview of MeOffendEs at IberLEF 2021: Offensive Language Detection in Spanish Variants.

Plaza-del-Arco, F. M., Molina-González, M. D., & Alfonso, L. (2021b). Sexism Identification in Social Networks using a Multi-Task Learning System. In *IberLEF 2021*, September 2021, Málaga, Spain.

Plaza-del-Arco, F. M., Montejo-Ráez, A., López, L. A. U., & Martín-Valdivia, M. T. (2021c, September). OffendES: A New Corpus in Spanish for Offensive Language Research. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pp. 1096-1108.

Plaza-del-Arco, F. M., Jiménez Zafra, S. M., Montejo Ráez, A., Molina González, M. D., Ureña López, L. A., & Martín Valdivia, M. T. (2021d). Overview of the EmoEvalEs task on emotion detection for Spanish at IberLEF 2021. In *Procesamiento del Lenguaje Natural*, 67, september de 2021, pp. 155-161.

Price, I., Gifford-Moore, J., Flemming, J., Musker, S., Roichman, M., Sylvain, G., Thain, N., Dixon, L., and Sorensen, J. (2020). Six attributes of unhealthy conversation. *arXiv preprint arXiv:2010.07410.*

Rajadesingan, A., Zafarani, R., & Liu, H. (2015, February). Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the eighth ACM international conference on web search and data mining* (pp. 97-106).

Ramírez-Cifuentes, D., Freire, A., Baeza-Yates, R., Puntí, J., Medina-Bravo, P., Velazquez, D. A., Gonfaus, J. M., & Gonzàlez, J. (2020). Detection of suicidal ideation on social media: multimodal, relational, and behavioral analysis. *Journal of medical internet research, 22*(7), e17758.

Ranasinghe, T., & Zampieri, M. (2020). Multilingual offensive language identification with cross-lingual embeddings. *arXiv preprint arXiv:2010.05324*.

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP),* pages 3982–3992, Hong Kong, China. Association for Computational Linguistics. - *arXiv preprint arXiv:1908.10084*.

Resnik, P., Armstrong, W., Claudino, L., Nguyen, T., Nguyen, V. A., & Boyd-Graber, J. (2015). Beyond LDA: exploring supervised topic modeling for depression-related language in Twitter. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality* (pp. 99-107).

Ripley, D. W. (2009). *Negation in natural language.* https://doi.org/10.17615/xjnm-ct07

Rodríguez-Sánchez, F., Carrillo-de-Albornoz, J., Plaza, L., Gonzalo, J., Rosso, P., Comet, M., & Donoso, T. (2021). Overview of exist 2021: sexism identification in social networks. *Procesamiento del Lenguaje Natural, 67*, 195-207.

Ross, B., Rist, M., Carbonell, G., Cabrera, B., Kurowsky, N., & Wojatzki, M. (2017). Measuring the reliability of hate speech annotations: The case of the european refugee crisis. *arXiv preprint arXiv:1701.08118.*

Ryu, S., Lee, H., Lee, D. K., Kim, S. W., & Kim, C. E. (2019). Detection of suicide attempters among suicide ideators using machine learning. *Psychiatry investigation, 16*(8), 588.

Samghabadi, N. S., Patwa, P., Pykl, S., Mukherjee, P., Das, A., & Solorio, T. (2020, May). Aggression and misogyny detection using BERT: A multi-task approach. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying* (pp. 126-131).

Scaboro, S., Portelli, B., Chersoni, E., Santus, E., & Serra, G. (2021). NADE: a benchmark for robust adverse drug events extraction in face of negations. *arXiv preprint arXiv:2109.10080*.

Schofield, A., & Davidson, T. (2017). Identifying hate speech in social media. *XRDS: Crossroads, The ACM Magazine for Students*, *24*(2), 56-59.

Shen, J. H., & Rudzicz, F. (2017, August). Detecting anxiety on reddit. In *Proceedings of the Fourth Workshop on Computational Linguistics and Clinical Psychology—From Linguistic Signal to Clinical Reality* (pp. 58-65).

Shushkevich, E., & Cardiff, J. (2019). Automatic misogyny detection in social media: A survey. *Computación y Sistemas, 23*(4), 1159-1164.

Tadesse, M. M., Lin, H., Xu, B., & Yang, L. (2020). Detection of suicide ideation in social media forums using deep learning. *Algorithms*, *13*(1), 7.

Taulé Delor, M., Ariza, A., Nofre, M., Amigó Cabrera, E., & Rosso, P. (2021). Overview of DETOXIS at IberLEF 2021: DEtection of TOXicity in comments In Spanish.

Van Aken, B., Risch, J., Krestel, R., & Löser, A. (2018). Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*.

Van Huynh, T., Nguyen, V. D., Van Nguyen, K., Nguyen, N. L. T., & Nguyen, A. G. T. (2019). Hate speech detection on vietnamese social media text using the bi-gru-lstm-cnn model. *arXiv preprint arXiv:1911.03644*.

Vioulès, M. J., Moulahi, B., Azé, J., & Bringay, S. (2018). Detection of suicide-related posts in Twitter data streams. *IBM Journal of Research and Development*, *62*(1), 7-1.

Vitiugin, F., & Barnabò, G. (2021). Emotion Detection for Spanish by Combining LASER Embeddings, Topic Information, and Offense Features. In *IberLEF@ SEPLN* (pp. 78-85).

Vu, X. S., Vu, T., Tran, M. V., Le-Cong, T., & Nguyen, H. (2020). HSD shared task in VLSP campaign 2019: Hate speech detection for social good. *arXiv preprint arXiv:2007.06493*.

Warholm, J. (2021). Detecting Ungealthy Comments in Norwegian using BERT. MS thesis. UiT The Arctic University of Norway.

Waseem, Z. (2016, November). Are you a racist or am I seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science* (pp. 138-142).

Waseem, Z., & Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop* (pp. 88-93).

Wiegand, M., Siegel, M., & Ruppenhofer, J. (2018). Overview of the germeval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval*.

Wulczyn, E., Thain, N., & Dixon, L. (2017, April). Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web* (pp. 1391-1399).

Xu, J. M., Jun, K. S., Zhu, X., & Bellmore, A. (2012, June). Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 656-666).

Yates, A., Cohan, A., & Goharian, N. (2017). Depression and self-harm risk assessment in online forums. *arXiv preprint arXiv:1709.01848.*

Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Predicting the type and target of offensive posts in social media. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technology (NAACL-HLT). - arXiv preprint arXiv:1902.09666.*

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z. & Çöltekin, Ç. (2020). SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the 14th international workshop on semantic evaluation. - arXiv preprint arXiv:2006.07235.*

Zhang, T. (2004, July). Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning* (p. 116).

Zhang, Z., Robinson, D., & Tepper, J. (2018, June). Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference* (pp. 745-760). Springer, Cham.

Zimmerman, S., Kruschwitz, U., & Fox, C. (2018, May). Improving hate speech detection with deep learning ensembles. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018).*