Procesamiento del Lenguaje Natural, Revista nº 67, septiembre de 2021, pp. 209-221 recibido 03-07-2021 revisado 10-07-2021 aceptado 14-07-2021

Overview of DETOXIS at IberLEF 2021: DEtection of TOXicity in comments In Spanish

Visión general de DETOXIS en IberLEF 2021: DEtección de TOXicidad en comentarios En Español

Mariona Taulé[1], Alejandro Ariza[1], Montserrat Nofre[1], Enrique Amigó[2], Paolo Rosso[3]
[1]CLiC, Universitat de Barcelona, Spain
[2]Research Group in NLP and IR, Universidad Nacional de Educación a Distancia, Spain
[3]PRHLT Research Center, Universitat Politècnica de València, Spain
{mtaule, alejandro.ariza14, montsenofre}@ub.edu, enrique@lsi.uned.es, prosso@dsic.upv.es

Abstract: In this paper we present the DETOXIS task, DEtection of TOxicity in comments In Spanish, which took place as part of the IberLEF 2021 Workshop on Iberian Languages Evaluation Forum at the SEPLN 2021 Conference. We describe the NewsCom-TOX dataset used for training and testing the systems, the metrics applied for their evaluation and the results obtained by the submitted approaches. We also provide an error analysis of the results of these systems.
Keywords: Toxicity detection, Rank Biased Precision, Closeness Evaluation Measure, NewsCom-TOX corpus.

Resumen: En este artículo se presenta la tarea DETOXIS, DEtección de TOxicidad en comentarios en español, que tuvo lugar en el Iberian Languages Evaluation Forum workshop (IberLEF 2021) en el congreso de la SEPLN 2021. Se describe el corpus NewsCom-TOX utilizado para entrenar y evaluar los sistemas, las métricas para evaluarlos y los resultados obtenidos por las distintas aproximaciones utilizadas. Se proporciona también un análisis de los resultados obtenidos por estos sistemas.
Palabras clave: Detección de toxicidad, Rank Biased Precision, Closeness Evaluation Measure, NewsCom-TOX corpus.

1 Introduction

The aim of the DETOXIS task is the detection of toxicity in comments posted in Spanish in response to different online news articles related to immigration. The DETOXIS task is divided into two related classification subtasks: Toxicity detection task and Toxicity level detection task, which are described in Section 2. The presence of toxic messages on social media and the need to identify and mitigate them leads to the development of systems for their automatic detection. The automatic detection of toxic language, especially in tweets and comments, is a task that has attracted growing interest from the Natural Language Processing (NLP) community in recent years. This interest is reflected in the diversity of the shared tasks that have been organized recently, among which we highlight those held over the last two years: HateEval-2019[1] (Basile et al., 2019) on hate speech against immigrants and women in English and Spanish tweets; TRAC-2 task on Aggression Identification[2] (Kumar et al., 2020) for English, Bengali and Hindi in comments extracted from YouTube; the OffensEval-2020[3] on offensive language identification (Zampieri et al., 2020) in Arabic, Danish, English, Greek and Turkish tweets; GermEval-2019 shared task[4] on the Identification of Offensive Language for German on Twitter (Struß et al., 2019); and the Jigsaw Multilingual Toxic Comment Classification Challenge [5], in which the task is focused on building multilingual models (English, French, German, Italian, Portuguese, Russian and Spanish) with English-only training data from Wikipedia comments.

DETOXIS is the first task that focuses

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

[2]https://sites.google.com/view/trac2/shared-task
[3]https://sites.google.com/site/offensevalsharedtask/
[4]https://projects.fzai.h-da.de/iggsa/the
[5]https://www.kaggle.com/c/jigsaw-multilingual-toxic-comment-classification

ISSN 1135-5948. DOI 10.26342/2021-67-18

on the detection of different levels of toxicity in comments posted in response to news articles written in Spanish. The main novelty of the present task is, on the one hand, the methodology applied to the annotation of the dataset that will be used for training and testing the participant models and, on the other hand, the evaluation metrics that will be applied to evaluating the participant models in terms of their system use profile applying four different metrics (F-measure, Rank Biased Precision (Moffat and Zobel, 2008), Closeness Evaluation Measure (Amigó et al., 2020) and Pearson's correlation coefficient). The methodology proposed aims to reduce the subjectivity of the annotation of toxicity by taking into account the contextual information, i.e. the conversational thread, and by annotating different linguistic features, such as argumentation, constructiveness, stance, target, stereotype, sarcasm, mockery, insult, improper language, aggressiveness and intolerance, which allowed us to discriminate the different levels of toxicity.

The rest of the overview is structured as follows. In Section 2 we present the two subtasks of DETOXIS. In Section 3 the corpus NewsCom-TOX used as dataset is described together with the way it was gathered and annotated. In Section 4 the different metrics used for the evaluation of the systems and the results they obtained are presented, as well as a description of the techniques and models used by systems. Finally, in Section 5 the conclusions are drawn.

2 Task Description

The aim of the DETOXIS task is the detection of toxicity in comments posted in Spanish in response to different online news articles related to immigration. The DETOXIS task is divided into two related classification subtasks:

- Subtask 1: Toxicity detection task is a binary classification task that consists of classifying the content of a comment as toxic (toxic=yes) or not toxic (toxic=no).
- Subtask 2: Toxicity level detection task is a more fine-grained classification task in which the aim is to identify the level of toxicity of a comment (0=not toxic; 1=mildly toxic; 2=toxic and 3=very toxic).

Although we recommended to participate in both subtasks, participants were also allowed to participate just in one of them. Teams were also encouraged to submit multiple runs (maximum 5).

3 Dataset: The NewsCom-TOX corpus

We used as dataset the NewsCom-TOX corpus, which consists of 4,359 comments posted in response to different 21 articles extracted from Spanish online newspapers (ABC, el-Diario.es, El Mundo, NIUS, etc.) and discussion forums (such as Menéame[6] and Forocoches[7]) from August 2017 to July 2020. These articles were manually selected taking into account their controversial subject matter, their potential toxicity, and the number of comments posted (minimum 50 comments). We used a keyword-based approach to search for articles related mainly to immigration. The comments were selected in the same order in which they appear in the time thread in the web. The author (anonymized), the date and the time when the comment was posted were also retrieved. The number of comments ranges from 65 to 359 comments per article. On average, 31.16% of the comments are toxic.

3.1 Annotation Scheme

We considered that a comment is toxic when it attacks, threatens, insults, offends, denigrates or disqualifies a person or group of people on the basis of characteristics such as race, ethnicity, nationality, political ideology, religion, gender and sexual orientation, among others. This attack can be expressed in different ways –explicitly (through insult, mockery and inappropriate humor) or implicitly (for instance through sarcasm)– and at different levels of intensity, that is at different levels of toxicity (from impolite and offensive comments to the most aggressive, the latter being those comments that incite hate or even physical violence). We use toxicity as an umbrella term under which we include different definitions used in the literature to describe hate speech ((Nockleby, 2000), (Waseem and Hovy, 2016), (Schmidt and Wiegand, 2017), (Davidson et al., 2017)) and abusive (Nobata et al., 2016), aggressive (Kumar et al., 2018), toxic (Kolhatkar et

[6]https://www.meneame.net/
[7]https://www.forocoches.com/

al., 2020) or offensive (**?**) language. In fact, these different terms address different aspects of toxic language (Poletto et al., 2020).

We annotated each comment in two categories 'toxic' and 'not toxic', and then we assigned different levels of toxicity: 'toxicity_level_0=not toxic' (1), 'toxicity_level_1=mildly toxic' (2), 'toxicity_level_2=toxic'(example 3) or 'toxicity_level_3=very toxic' (4) to those that were annotated first as toxic [8].

(1a) Contra la desinformación datos. En el INE tenéis los datos de los inmigrantes. Delitos, cotizaciones, prestaciones, etc. <toxicity=not> <toxicity_level=0>

(1b) Against misleading data. In the INE you have data for immigrants. Crimes, contributions, subsidies, etc. <toxicity=not> <toxicity_level=0>

(2a) Esta gentuza se carga al país en dos telediarios
<toxicity=yes> <toxicity_level=1>

(2b) This rabble could destroy the country with two TV news programs
<toxicity=yes> <toxicity_level=1>

(3a) Lo que importa en realidad es sentirse mejor con uno mismo. Sumar karma. A los putos negros les pueden joder bien. <toxicity=yes> <toxicity_level=2>

(3b) What really matters is feeling better about yourself. Accumulating karma. Fucking blacks can get fucked up pretty good.
<toxicity=yes> <toxicity_level=2>

(4a) A estos putos animales sarnosos que los encierren y tiren la llave. <toxicity=yes> <toxicity_level=3>

(4b) With these mangy fucking animals, lock them up and throw away the key. <toxicity=yes> <toxicity_level=3>

In addition to annotating whether or not a comment was toxic and its level of toxicity, we also annotated the following features: argumentation, constructiveness, stance, target, stereotype, sarcasm, mockery, insult, improper language, aggressiveness and intolerance. All these features have binary values except the toxicity level (the value '0' is assigned for indicating the absence of the corresponding feature in the comment, and the value '1' is assigned when the feature is present). We first annotated these features and we then used them to establish the toxicity of comments and to determine their level of toxicity. It is worth noting that the level of toxicity is especially determined by the type of features combined. All this information was included only in the training dataset that was used for the task. In order to assign each of these features and be able to interpret the global meaning of comments, it was crucial to take the context into account, that is, the conversational thread. For instance, the conversational threat was very useful for interpreting and annotating sarcasm, mockery and stance. The identifier of each conversation thread ('thread_id' feature) was also provided for the participants, as well as the identifier of the previous comment in the thread ('reply_to' feature) and the 'comment_level' feature. The latter is a categorical attribute with two possible values: '1' for indicating that the comment is a direct comment to an article and '2' for indicating that the comment is a reply to another comment. The 'topic' feature was also provided in the training dataset. This feature has three possible values (CR=crime, MI=migration, SO=social) to distinguish the topic of the news article.

3.2 Annotation Process

Each comment was annotated in parallel by three annotators and an inter-annotator agreement test was carried out once all the comments on each article had been annotated. Then, disagreements were discussed by the annotators and a senior annotator until agreement was reached. The team of annotators involved in the task consisted of two expert linguists and two trained annotators, who were students of linguistics. Table 1 shows the results obtained in the inter-annotator agreement test, the average observed percentage of agreement between the three annotators, 81.99% for toxicity and 73.95% for toxicity level, and Krippenddorfs' alpha (0.59 and 0.62 respectively), which ensures annotation reliability. The results obtained are quite acceptable considering the difficulty of both tasks.

[8]The examples contain text that may be considered offensive.

Feature	Average observed agreement	Krippendorff's α
Toxicity	81.99%	0.59
Toxicity_level	73.95%	0.62

Table 1: Inter-Annotator Agreement Test.

3.3 Training and Test Dataset

We provided participants with 80% of the NewsCom-TOX corpus for training their models (3,463 comments), and the remaining 20% of the corpus (896 comments) was used for testing their models.[9] Both training and test files were provided in *csv* format. The training dataset contains all the features included in the annotation of the NewsCom-TOX corpus (see Subsection 3.1), whereas the test dataset only contains the following features: 'thread_id', 'comment_id', 'reply_to', 'comment_level' and 'comment'.

Table 2 shows the distribution of toxic comments by toxicity level. The dataset consists of 4,359 comments, of which 1,385 are toxic (31.16%): 996 mildly toxic (21.90%), 310 toxic (7.36%) and 79 very toxic (1.81%).

4 Systems and Results

This section contains a brief description of the baselines provided as a benchmark, followed by an overview of both subtasks (toxicity detection and toxicity level detection) mentioning the reasons behind the selection of the evaluation metrics and their implication when models are compared. Finally, some interesting systems insights and a brief error analysis of the submitted approaches are presented.

4.1 Baselines

A benchmark was set with the introduction of three different baselines: RandomClassifier, BOWClassifier and ChainBOW. First, RandomClassifier assigns a random of toxicity from four possible values {0, 1, 2, 3} to each comment in the test set without any kind of weighting strategy. Second, the BOWClassifier consists of a simple Support Vector Classifier (SVC) that receives the features extracted by a TF-IDF Vectorizer (an advanced version of the classical Bag-Of-Words technique). In particular, the SVC model uses a linear kernel with a regularization parameter equal to 1.0 and a "one-versus-rest" decision function strategy for the toxicity level detection subtask. Furthermore, the TF-IDF Vectorizer, which is responsible for feature extraction, constructs a vocabulary of 5,000 entries including unigrams and bigrams after performing a simple preprocessing stage (lowercasing and removal of accents). Finally, due to the fact of having an unbalanced dataset, we decided to include a baseline that processes both subtasks sequentially. Therefore, ChainBOW contains two BOWClassifiers, one for each subtask (toxicity detection and toxicity level detection), connected in a hierarchical fashion with the same configuration as mentioned before. It is worth mentioning that, given their baseline nature, no hyperparameter optimization was performed on any of the models. In addition, all baselines were implemented using the Python scikit-learn library.

4.2 Subtask 1: Toxicity Detection Task

Subtask 1 consists of a binary classification (toxic vs. non-toxic). In this subtask, the metrics precision, recall and their combination by means of the F-measure were used. Table 5 ranks the best run of each of the participating teams according to F-measure. In general, the SINAI system outperforms the rest of systems. SINAI is outperformed by some of the other approaches in terms of recall, although at the cost of a significant precision loss. In relation to the baselines, RandomClassifier achieves a mid-ranking position in the ranking, with low precision but medium high recall. In a similar position is ChainBOW, with low recall and medium high precision. In general, there is significant room for improvement between participants runs and the Gold Standard.

Figure 1 illustrates the precision and recall scores achieved by the systems. The precision of all systems is between 0.25 and 0.75.

[9] In order to avoid any conflict with the sources of comments regarding their Intellectual Property Rights (IPR), the data was privately sent to each participant that was interested in the task. The corpus will be only available for research purposes.

Feature	Comments	Percentage
mildly toxic (level 1)	996	21.90%
toxic (level 2)	310	7.36%
very toxic (level 3)	79	1.81%
Total	1,385	31.16%

Table 2: Distribution of toxic comments.

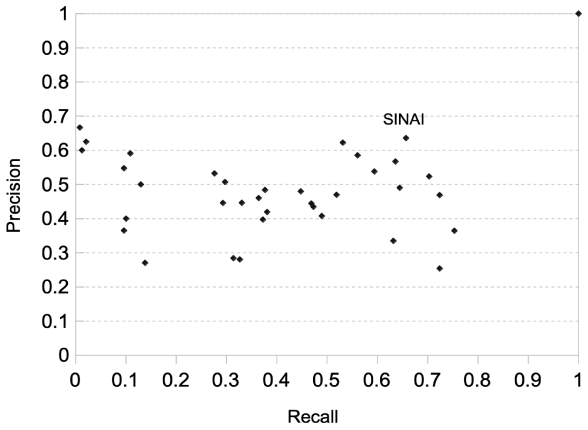

Figure 1: Precision and Recall for the best run of each team in Subtask 1.

4.3 Subtask 2: Toxicity Level Detection Task

Regarding Subtask 2, the toxicity level detection task is a more fine grained classification task in which the aim is to identify the level of toxicity of a comment (0= not toxic; 1= mildly toxic; 2= toxic and 3: very toxic). For this task we considered four evaluation metrics.

Closseness Evaluation Metric (CEM): The metric CEM (Amigó et al., 2020) considers the proximity between predicted and real categories. Unlike measures based on absolute error, CEM has two particularities. First, it assumes no equidistance between categories. That is, an error between category 0 and 2 is not necessarily twice as serious as an error between categories 0 and 1. In addition, it assigns more weight to infrequent categories. This avoids over-weighting systems that tend to classify items in majority classes. CEM is appropriate for unbalanced data sets, and when the relative distance between categories beyond the way they are ordered is unknown.

Rank Biased Precision (RBP): RBP is a ranking metric (Moffat and Zobel, 2008). We rank the output of the system and the Gold Standard on the basis of toxicity levels (from highest to lowest) and compare the two rankings. Basically, RBP is computed as the sum of the actual values (Gold Standard) along the ranking generated by the system with a weight function that decreases as we decay in the ranking of the system output. d corresponds to the categorized items in the system output s, with $pos(d)$ being the ranking position of d in the system output s and $g(d)$ being the real category value in the Gold Standard for this item. RBP is computed as:

$$RBP = (1-p)\sum_{d} f(d)g(d)$$

where f is a decay function that decreases with position i, concretely $f(d) = p^{pos(i)-1}$. The value p is a parameter which is typically fixed at X. Originally, RBP applies to rankings with one item in each ranking position. However, in our scenario, the items are ordered in four levels in the ranking generated by the system (level 4= not toxic; level 3= mildly toxic; level 2= toxic and level 1: very toxic). Therefore, we modify RBP by considering the average i position that each item would occupy in the system output if we were to randomly order the ties:

$$f(d) = \frac{1}{n_{s(d)}} \sum_{i=MinPos(d)}^{MaxPos(d)} p^{i-1}$$

where $n_{s(d)}$ represents the number of items at level $s(d)$ in the system output. $MinPos(d)$ and $MaxPos(d)$ represent the minimum and maximum position that item d could occupy in the system output s according to its level. This metric is appropriate when sorting items according to their toxicity. For instance, the scenario in which the user needs to

find the most toxic comments fits in this metric.

Accuracy: Accuracy is the most popular classification metric. This metric does not consider the order between categories. That is, an error is penalised regardless of the distance to the actual category. Furthermore, it does not compensate for the effect of imbalance in the data set. This metric is not appropriate for most possible scenarios, although it has the advantage of being very easy to interpret in terms of the percentage of hits.

Mean Average Error (MAE): MAE averages the absolute difference between predicted and actual categories. This metric assumes equidistance between categories and does not compensate for the effect of imbalance between categories in the data set. MAE is appropriate, for example, when predicting the average toxicity value in a comment set. Note this requires assuming numerical intervals between categories.

Pearson Coefficient: Like MAE, using Pearson coefficient requires assuming equidistance between categories. The difference with respect to MAE is that it does not require the system to predict the category of each item, but generates a scale that is linearly correlated with the actual scale. In addition, it compensates for the effect of imbalance between categories by giving more weight to those categories that are more infrequent. Obtaining a high Pearson value is interesting, for example, when predicting the evolution of toxicity over time in a comment stream or compare the average toxicity of two streams.

Table 6 shows the results obtained by the best run of each team in terms of CEM. The baselines are in the middle positions of the systems ranking. In this case, the BOW-Classifier approach performs better than the other baselines developed by the task organisers.

Figure 2 illustrates the CEM results vs. other evaluation metrics. In this subtask, the systems SINAI and Team Sabari outperform most of the other approaches for all metrics. There is only one exception. The system output DCG outperforms SINAI and Sabari by a wide margin in terms of ranking (RBP metric). This means that, in a scenario where the objective is to prioritise particularly toxic comments, DCG is more effective than SINAI.

In Figure 2, we find a high correspondence between CEM and the Pearson correlation coefficient. CEM does not consider numeric intervals. The Pearson coefficient does not take into account the proximity of the estimated categories to the actual ones but the correlation between the values (numeric category labels) in the system output and the gold standard. This means that the effect of 'scale shifts' and the effect of assuming numeric intervals between categories is not particularly relevant in this evaluation benchmark.

However, we did not find as much correspondence between the CEM metric and the MAE and Accuracy metrics. The main difference is that CEM, like Pearson, compensates for the imbalance between categories in the data set, giving more weight to errors or hits in infrequent categories. As Figure 2 shows, there is a set of runs that obtain similar Accuracy and MAE values, but nevertheless present important differences in terms of CEM. From a usage scenario perspective, this result indicates that these runs are comparable if we are interested in predicting or approximating the actual category in as many cases as possible. For example, this would be the case if we wanted to calculate the average toxicity of a set of comments. However, if we are interested in detecting particular cases of toxicity (low, medium or high) then we can assert that some of these runs will be more effective than others.

4.4 Systems Insights

A total of 31 teams (Subtask 1) and 24 teams (Subtask 2) sent a maximum of five submissions per subtask to be evaluated. Not surprisingly, and based on the recent success of transfer learning with pre-trained Transformer-like language models, the top five teams in both subtasks achieved their best scores using BETO[10] (the Spanish version of the BERT model). The difference in performance between their respective submissions thus lies in the fine-tuning techniques, data augmentation strat-

[10] https://github.com/dccuchile/beto

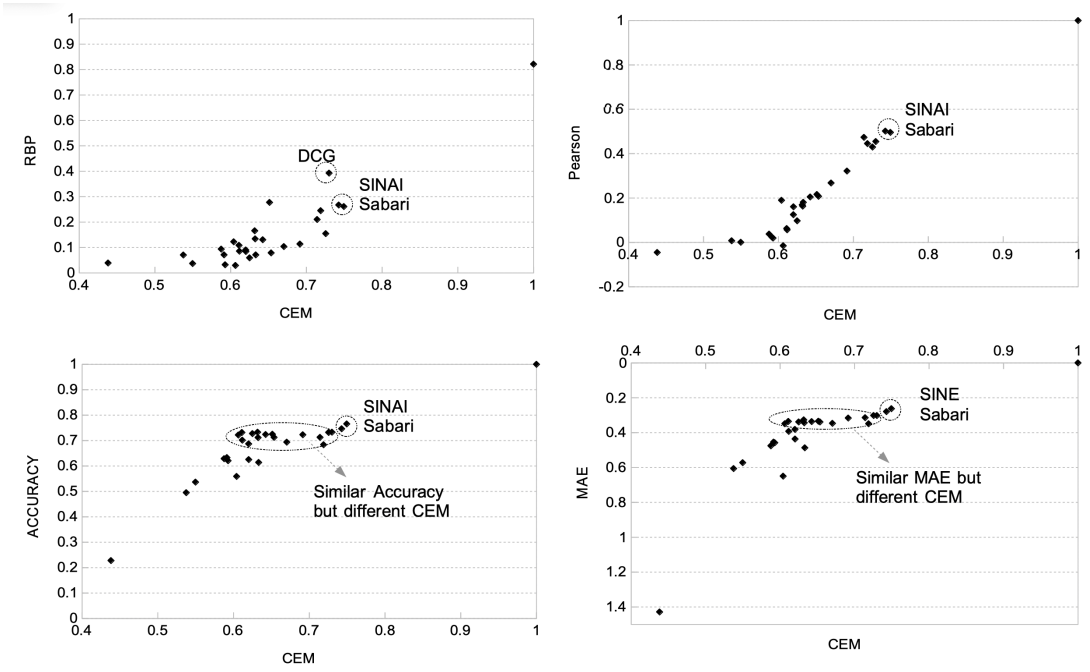

Figure 2: Correspondence between CEM and other metrics in Subtask 2.

egy and/or preprocessing steps. Although the performance of classical machine learning models such as TF-IDF with Random Forests/Support Vector Machines/Logistic Regression Classifiers were not as high as those achieved by BETO, they were also used as part of ensemble architectures and analyzed by multiple participants. In fact, several teams provided valuable insights and comparisons of models for a task that is incredibly challenging, even for trained expert annotators. The most interesting conclusions are summed up below.

This competition introduced several challenges in addition to the toxicity detection problem. First, the language of the dataset (comments in Spanish) differs from the usual English language present in most of the general benchmark NLP datasets. Therefore, the models needed to account for the different directionality of the text, as well as for the selection of other pre-processing steps. For instance, the GTH-UPM team performed an analysis of the multiple pre-processing steps applied prior to a pre-trained Transformer-like model (BETO). Apparently, a basic text normalization (the replacement of special Named Entities such as MAIL, DATE, URL... with their respective shared token) improves BETO's performance as opposed to other steps e.g. the removal of stopwords and punctuation or lemmatization. GTH-UPM also validated the extension of pre-trained models' vocabulary with domain-specific terms as a useful technique to improve task-oriented results.

Another analysis that is worth paying attention to regarding the language of the texts is a performance comparison among pretrained multilingual models, English models with a proper Spanish-to-English translator and Spanish models. Alejandro Mosquera built a stacked model composed of at least one model for each variant in conjunction with additional features and extracted the feature importance of the individual models via cross-validation on the training dataset. The results show that pretrained multilingual models such as XLM do not provide as much predictive power in this toxicity context as a neural network with a capsule network architecture using SBWC i25 GloVe Spanish embeddings. However, the difference is not so pronounced when we compare it with the translator followed by the model trained on English embeddings. A similar comparison was performed by the AI-UPV team task in which classical machine learning models (both generative and discriminative), multilingual BERT and BETO performed cross-validation with a hyperparameter setup. Not only did they conclude that transformer-like architectures outperform classical statistical models in this complex task, but they also

found that BETO achieved substantially better results than multilingual BERT for all of their top five best configurations and in both subtasks.

Furthermore, Alejandro Mosquera's was the only team to introduce side information related to the topic of the articles each comment refers to and the thread to which they belong. As a preliminary analysis using cross-validation, the use of information related to the comments present in the thread/conversation seems to enrich overall model performance. Further research on how to better exploit thread and topic information may be an interesting and promising direction.

Last but not least, the winners of both DETOXIS subtasks (SINAI team proved the importance of enriching the model by fine-tuning it on similar tasks related to sentiment and emotion analysis, thus increasing the available training data and making more precise predictions thanks to this Multi-Task Learning strategy. In fact, SINAI was the only team in the competition that took the provided extra features (see Section 3.1) into account, thereby giving their systems an important boost in comparison to the other participants. However, a study of the influence of each extra feature on the model's predictive power is yet to be performed.

4.5 Error Analysis

The tasks of toxicity and toxicity level detection are particularly difficult for machine learning models. In fact, even the most recent transformer models struggle with linguistic devices users sometimes use such as sarcasm or irony. However, thanks to the fact that the dataset includes 13 additional features that classify the text according to other helpful semantic dimensions from the raw text, we can easily identify the most challenging comments and the possible reasons behind the difficulty of their detection. Table 3 contains the average performance of the top five best submissions per task (based on the F1-score and CEM metrics respectively) considering the subset of samples where each feature is present and a single (best) run per team.

Regarding the first task (toxicity detection), the official metric focuses on the ability of the models to detect all toxic comments and pays zero attention to the negative class. Therefore, comments with *argumentative* cues (generally linked to non-toxic comments) that are marked as toxic are usually the most difficult ones to detect (see examples 5, 8, 10, 13 and 14 in Table 4). A similar situation appears in comments with a *positive stance* that are actually toxic. Other dimensions that clearly increase the difficulty of this task are *sarcasm* and *stereotype*. The existence of a greater number of implicit stereotypes rather than explicit ones in hate speech posts which ultimately require models to learn related world knowledge beforehand is well known. Furthermore, this dataset contains comments that are replies to other users' comment. Consequently, a broader context than just the current text is sometimes required to make an informed decision on whether the comment is toxic or not.

On the other hand, the difficulty of linguistic cues related to aggressiveness, mockery, sarcasm and stereotype is even clearer in the task of toxicity level detection when CEM scores are considered. Even though a comment may not apparently be very toxic according to the explicit words in the text, implicit messages and emitter intention can be really harmful for the receiver. Consequently, an effort to build automatic systems able to capture toxicity levels in comments in which such subtle hidden messages are included is of utmost importance. For instance, example 12 is usually marked as non-toxic by the top performing submitted systems although the actual toxicity is at its maximum level due to the implicit aggressiveness within the message. This difference between the predicted and ground truth labels is highly penalized by the official CEM metric.

5 Conclusions and Future Work

This paper has described the DETOXIS challenge at IberLEF 2021 and summarized the participation of several teams in both subtasks which evidenced interesting approaches and conclusions. Although all of the top-performing systems made use of the Spanish version of the BERT model (a Transformer encoder), a variety of insights into the importance of different strategies become clearer for researchers to build upon or explore further in their respective studies on hate speech and toxicity detection. As opposed to previous challenges on hate speech, we have

Feature	Size (Toxic)	Subtask 1	Subtask 2
1. argumentation	459 (113)	0.6006	0.7591
2. constructiveness	282 (0)	-	0.8929
3. positive_stance	22 (7)	0.6545	0.8465
4. negative_stance	252 (125)	0.6919	0.6764
5. target_person	108 (90)	0.7243	0.5672
6. target_group	67 (62)	0.7945	0.5444
7. stereotype	41 (38)	0.6938	0.5110
8. sarcasm	52 (42)	0.6653	0.5497
9. mockery	80 (77)	0.7443	0.4824
10. insult	85 (83)	0.8541	0.5202
11. improper_language	74 (51)	0.7985	0.6542
12. aggressiveness	15 (15)	0.7091	0.3892
13. intolerance	15 (13)	0.8516	0.5655

Table 3: Average score (F1-score and CEM respectively) of the five best performing teams in each task for the subset of comments where the corresponding feature is positive together with the size of each subset and the number of toxic instances in them.

provided 13 additional features and a fine-grained toxicity degree target with four possible labels that go beyond the classical binary toxicity classification. Furthermore, the collection of comments includes the thread to which they belong and the topic of the article they are posted in, thus, allowing a broader context to be used by innovative solutions.

Unsurprisingly, multilingual models are outperformed by the Spanish counterparts in this challenge most probably due to the specific embedding space fully optimized in the given language and the use of a more language-oriented token vocabulary. Techniques such as data augmentation with Masked Language Modeling and the addition of Multi-Task Learning with datasets belonging to similar tasks in order to increase the available data used at the finetuning step has turned out to be beneficial in a scenario in which the number of comments is small and classes were unbalanced. Moreover, despite the fact that further analysis and more elaborate ways to extract information from the conversation and the topic are needed, the only team that took advantage of such information achieved positive results in the final outcome by combining them with other common strategies.

Finally, it is important to mention that, depending on the final application and its specific requirements, the selection of the model may require a different evaluation metric. Thanks to our selection of metrics, we were able to provide a simple visualization of different trade-offs when opting for a certain system and how models perfectly suitable for a particular goal may not perform at the same level when those requisites change. A clear example we have presented is the comparison between the SINAI and DCG systems, in which the former was better at detecting all toxicity levels according to an ordinal classification metric such as CEM but was outperformed by the latter in scenarios in which prioritizing the most toxic comments is the priority (according to the RBP ranking metric).

Regarding future work, systems are as yet far from the Gold Standard and this is mainly due to the difficulty these models encounter when detecting implicit features such as aggressiveness, mockery and sarcasm. Moreover, the lack of examples corresponding to each individual feature makes this an even more challenging task. Additional research on the influence of these implicit features on the final toxicity of the comment and the creation of systems able to detect and mitigate such subtle cues is highly necessary in a society in which messages and ideas are spread faster than ever. Some of the paths that were not explored by participants and could be worth looking into are: larger language models or possible distillations of them, improvements to the semantic contextualization of language models by the use of techniques that reduce their anisotropicity problem and the incorporation of sentence encoders or compositional representations to the ensemble architectures.

ID	Features	FN Ratio	Toxicity	Comment
1	8	1.0000	1	Como se echa de menos la opinión de los guardianes de la moral en la noticia de la ejecución en Portland. Bueno, en muchos casos están sus negativos pero no sus siempre dignos comentarios.
2	10, 11	1.0000	1	Directamente :peineta:
3	4	1.0000	1	Los andaluces, no los españoles
4	-	0.9677	1	Y 600000 votos más, para el pandemias.
5	1, 4, 12	0.9677	1	No intentaba poner ni a favor ni en contra. Ni intentaba realizar una crítica en el origen de los EEUU. Mi comentario solo trataba de dar perspectiva del problema en USA. En mi opinión personal todos los usanos con armas podrían salir a la calle hasta matarse los unos a los otros que el mundo en un plazo más corto que largo iría a mejor. Pero es mi opinión.
6	3, 4, 5, 10	0.9677	1	Que fácil es hablar bocachancla
7	5	0.9677	1	Expulsion de echenique de españa ya1
8	1, 4, 6, 9, 10	0.9355	1	En Madrid no hay campo, no hablo de la púrria de ciudad. Hablo de la gente que todo los años entra en España para hacer estos trabajos estacionales + los que están siempre aquí y que se han ido
9	4, 11	0.9355	1	Como conseguir 600.000 votos. Vomitivo
10	1, 5	0.9355	1	El segundo es el que le tira al suelo, pero mira el primero también, figura. youtu.be/neUnhYO2Ehc
11	6, 7	0.9355	1	Pues más razón para hacer como ellos contra ellos. La diferencia es que nosotros sabemos que esta mal, ellos creen que es lo correcto. Nosotros podemos parar cuando lo estimemos, ellos no.
12	12	0.9355	3	Tengo esperanzas en que legalicen la noche de 'La Purga'. La peli cuenta como las clases acomodadas impulsan esta celebración para que el lumpen se autorregule. Es sencilla, pero a mi me entretuvo mucho :)
13	1, 4, 5, 9	0.9355	1	Echenique miente mas que corre....aun regularizando estos irregulares, no serian de hecho "ciudadanos españoles" y por lo tanto SIN derecho a voto. Pura propaganda bolchevique.
14	1, 4, 5	0.9355	1	ss. 29.304 simplemente exige que tengas licencia de caza (que no se sabe publicamente si el implicado la tiene o no) [...] Ergo, podemos deducir que el acusado no estaba violando la 948.60(2)(a) al usar un arma larga con 17 años. Hay que hacer los deberes antes de soltar afirmaciones tan categóricas. O sea que necesita una licencia de caza y como no sabemos si la tiene o no vamos a suponer que SI la tiene pero el que tiene que hacer los deberas antes de hacer afirmaciones categóricas soy yo. Pos fueno, pos fale, pos malegro, campeón. al parecer al final no va a haber acusación por la tenencia de armas, ¿Fuente?
15	5, 8	0.9355	2	Si le hacen tantas pregunta a "NUMERO 3", es posible que se cortocircuite?
16	4, 13	0.8710	1	Deportaciones masivas ya!

Table 4: List of the top 16 comments in the test set that are misclassified as non-toxic by the majority of the systems. The Features column shows the feature identifiers from Table 3 marked as positive for the given comment, FN Rate is the ratio of systems that marked the comment as non-toxic and Toxicity refers to the four-levels annotation for that specific comment.

Acknowledgments

The work has been carried out in the framework of the following projects: MISMIS project (PGC2018-096212-B), funded by Ministerio de Ciencia, Innovación y Universidades (Spain), CLiC SGR (2027SGR341), funded by AGAUR (Generalitat de Catalunya) and STERHEOTYPES project (Challenges for Europe), funded by Fondazione Compangia di San Paolo.

References

Amigó, E., J. Gonzalo, S. Mizzaro, and J. Carrillo-de Albornoz. 2020. Effectiveness metric for ordinal classification: Formal properties and experimental results. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Basile, V., C. Bosco, E. Fersini, N. Debora, V. Patti, F. M. R. Pardo, P. Rosso,

M. Sanguinetti, et al. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *13th International Workshop on Semantic Evaluation*, pages 54–63. Association for Computational Linguistics.

Davidson, T., D. Warmsley, M. Macy, and I. Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Kolhatkar, V., H. Wu, L. Cavasso, E. Francis, K. Shukla, and M. Taboada. 2020. The sfu opinion and comments corpus: A corpus for the analysis of online news comments. *Corpus Pragmatics*, 4(2):155–190.

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

Kumar, R., A. K. Ojha, S. Malmasi, and M. Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 1–5.

Moffat, A. and J. Zobel. 2008. Rank-biased precision for measurement of retrieval effectiveness. *ACM Transactions on Information Systems (TOIS)*, 27(1):1–27.

Nobata, C., J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153.

Nockleby, J. T. 2000. Hate speech. *Encyclopedia of the American constitution*, 3(2):1277–1279.

Poletto, F., V. Basile, M. Sanguinetti, C. Bosco, and V. Patti. 2020. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, pages 1–47.

Schmidt, A. and M. Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the fifth international workshop on natural language processing for social media*, pages 1–10.

Struß, J. M., M. Siegel, J. Ruppenhofer, M. Wiegand, M. Klenner, et al. 2019. Overview of germeval task 2, 2019 shared task on the identification of offensive language. In *Preliminary proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*.

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

Zampieri, M., P. Nakov, S. Rosenthal, P. Atanasova, G. Karadzhov, H. Mubarak, L. Derczynski, Z. Pitenis, and Ç. Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447. International Committee for Computational Linguistics.

A Appendix: Results for both Subtasks

This appendix provides the results obtained by each team, selecting their best scoring output, for both DETOXIS subtasks. Table 5 ranks the best run of each of the participating teams according to F-measure. Table 6 shows the results obtained by the best run of each team in terms of CEM (column 3). The rest of columns show the results in terms of MAE, RBP, Pearson and Accuracy metrics. The baselines are in the middle positions of the systems ranking.

Ranking	Team	F-measure	Precision	Recall
1	SINAI	0.6461	0.6356	0.6569
2	GuillemGSubies	0.6000	0.5234	0.7029
3	AI-UPV	0.5996	0.5672	0.6360
4	DCG	0.5734	0.6225	0.5314
5	GTH-UPM	0.5726	0.5852	0.5607
6	alejandro mosquera	0.5691	0.4688	0.7238
7	Team Sabari	0.5646	0.5379	0.5941
8	maia	0.5570	0.4904	0.6444
9	address	0.4930	0.4697	0.5188
10	Javier_García_Gilabert	0.4911	0.3644	0.7531
11	Dembo	0.4632	0.4798	0.4477
12	ToxicityAnalizers	0.4562	0.4444	0.4686
13	DetoxisLuciaYNora	0.4529	0.4346	0.4728
14	YaizaMiñana	0.4449	0.4077	0.4895
15	VG UPV	0.4377	0.3348	0.6318
16	Datacism	0.4235	0.4839	0.3766
17	Calabacines cósmicos	0.4065	0.4603	0.364
18	Ivan_i_Jaume	0.3991	0.4194	0.3808
19	CarlesyJorge	0.3844	0.3973	0.3724
20	NEON2	0.3798	0.4463	0.3305
	RandomClassifier	0.3761	0.2540	0.7238
	ChainBOW	0.3747	0.5071	0.2971
21	JOSAND10	0.3636	0.5323	0.2762
22	GCP	0.3535	0.4459	0.2929
23	LlunaPerez-JúliaGregori	0.3017	0.2806	0.3264
24	GalAgo	0.2982	0.2841	0.3138
25	Just Do It	0.2060	0.5000	0.1297
	BOW Classifier	0.1837	0.5909	0.1088
26	Benlloch	0.1828	0.2705	0.1381
27	LNR_SaraSoto_ClaraSalelles	0.1637	0.5476	0.0962
28	LNR_IñigoPicasarri_JoanCastillo	0.1637	0.5476	0.0962
29	ElenaLopez_MartaGarcia_LNR	0.1605	0.4000	0.1004
	Word2VecSpacy	0.1523	0.3651	0.0962
30	Iker&Miguel	0.0405	0.6250	0.0209
31	JOREST	0.0246	0.6000	0.0126

Table 5: Results of Subtask 1.

Ranking	Team	CEM	MAE	RBP	Pearson	Accuracy
	Gold_Standard	1	0	0.8213	1	1
1	SINAI	0.7495	0.2626	0.2612	0.4957	0.7654
2	Team Sabari	0.7428	0.2795	0.2670	0.5014	0.7464
3	DCG	0.7300	0.3019	0.3925	0.4544	0.7329
4	GTH-UPM	0.7256	0.3019	0.1545	0.4298	0.7318
5	GuillemGSubies	0.7189	0.349	0.2449	0.4451	0.6835
6	AI-UPV	0.7142	0.3143	0.2101	0.4734	0.7127
7	address	0.6915	0.3165	0.1136	0.3215	0.7228
8	Dembo	0.6703	0.3468	0.1037	0.2677	0.6936
	ChainBOW	0.6535	0.3389	0.0787	0.2077	0.7127
9	Javier_García_Gilabert	0.6514	0.3345	0.2773	0.2158	0.7250
10	JOSAND10	0.6424	0.3367	0.1306	0.2046	0.7239
11	ToxicityAnalizers	0.6332	0.4871	0.0709	0.1805	0.6139
12	NEON2	0.6324	0.3434	0.1339	0.1632	0.7116
	BOWClassifier	0.6318	0.3266	0.1657	0.1688	0.7329
13	JOREST	0.6250	0.3389	0.0592	0.0972	0.7273
14	Iker&Miguel	0.6250	0.3389	0.0592	0.0972	0.7273
15	Ivan_i_Jaume	0.6201	0.4366	0.0844	0.1604	0.6251
16	DetoxisLuciaYNora	0.6200	0.3816	0.0903	0.1248	0.6869
	Word2VecSpacy	0.6116	0.3928	0.0855	0.0566	0.7015
	GloVeSBWC	0.6111	0.3356	0.1085	0.0623	0.7318
17	CarlesyJorge	0.6094	0.3367	0.0535	NaN	0.7318
18	ElenaLopez_MartaGarcia_LNR	0.6064	0.3490	0.0294	-0.0153	0.7217
19	VG UPV	0.6041	0.6498	0.1224	0.1896	0.5589
20	Just Do It	0.5928	0.4579	0.0320	0.0195	0.6207
21	Benlloch	0.5913	0.4545	0.0711	0.0237	0.633
22	GalAgo	0.5876	0.4759	0.0936	0.0372	0.6285
23	LlunaPerez-JúliaGregori	0.5498	0.5724	0.0369	0.0004	0.5365
24	JosepCarles_CandidogGarcia_LNR	0.5376	0.6061	0.0705	0.0072	0.4949
	RandomClassifier	0.4382	1.4287	0.0390	-0.0455	0.2278

Table 6: Results of Subtask 2.