information

Article

Fake News Spreaders Detection: Sometimes Attention Is Not All You Need

Marco Siino [1,*], Elisa Di Nuovo [2], Ilenia Tinnirello [1] and Marco La Cascia [1]

[1] Department of Engineering, University of Palermo, 90128 Palermo, PA, Italy
[2] Dipartimento di Lingue e Letterature Straniere e Culture Moderne, University of Turin, 10124 Torino, TO, Italy
* Correspondence: marco.siino@unipa.it

Abstract: Guided by a corpus linguistics approach, in this article we present a comparative evaluation of State-of-the-Art (SotA) models, with a special focus on Transformers, to address the task of Fake News Spreaders (i.e., users that share Fake News) detection. First, we explore the reference multilingual dataset for the considered task, exploiting corpus linguistics techniques, such as chi-square test, keywords and Word Sketch. Second, we perform experiments on several models for Natural Language Processing. Third, we perform a comparative evaluation using the most recent Transformer-based models (RoBERTa, DistilBERT, BERT, XLNet, ELECTRA, Longformer) and other deep and non-deep SotA models (CNN, MultiCNN, Bayes, SVM). The CNN tested outperforms all the models tested and, to the best of our knowledge, any existing approach on the same dataset. Fourth, to better understand this result, we conduct a post-hoc analysis as an attempt to investigate the behaviour of the presented best performing black-box model. This study highlights the importance of choosing a suitable classifier given the specific task. To make an educated decision, we propose the use of corpus linguistics techniques. Our results suggest that large pre-trained deep models like Transformers are not necessarily the first choice when addressing a text classification task as the one presented in this article. All the code developed to run our tests is publicly available on GitHub.

Keywords: fake news; misinformation; Natural Language Processing (NLP); transformers; Twitter; convolutional neural networks; text classification; deep learning; machine learning; user classification; author profiling; corpus linguistics; linguistic analysis

Citation: Siino, M.; Di Nuovo, E.; Tinnirello, I.; La Cascia, M. Fake News Spreaders Detection: Sometimes Attention Is Not All You Need. *Information* **2022**, *13*, 426. https://doi.org/10.3390/info13090426

Academic Editor: Kostas Vergidis

Received: 18 August 2022
Accepted: 6 September 2022
Published: 9 September 2022

1. Introduction

Fake News (FN) is not a recent problem. If, centuries ago, someone wrote that "A lie can travel halfway around the world while the truth is putting on its shoes"—quotation wrongly attributed to Mark Twain—now it can be certainly agreed that FN propagate faster, thanks to the advent of internet and, especially, of social media, by which information can reach simultaneously and anonymously every corner of the earth [1]. The growing use of social networks facilitated sharing non-intermediated contents. The evaluation of credibility is left to users' judgment, often compromised by the challenges posed by unfamiliar topics.

Since Russia's recent invasion of Ukraine on 24 February, social media have played a major role in spreading propaganda news, rumours and false or misleading news, photos and videos (https://tinyurl.com/EDMO-factChecking-Ukraine; https://edition.cnn.com/2022/03/05/politics/fact-check-fake-cnn-ukraine/index.html; https://observers.france24.com/en/europe/20220303-debunked-ukraine-russia-resistance; https://www.forbes.com/sites/kateoflahertyuk/2022/02/25/russia-ukraine-crisis-how-to-tell-if-pictures-and-videos-are-fake/, accessed on 15 August 2022). A blurry video claiming to show a Ukrainian girl confronting a Russian soldier generated 12 million views on TikTok and nearly 1 million views on Twitter. But actually it shows the Palestinian girl Ahed Tamimi,

aged 11 at the time, confronting an Israeli soldier after her older brother was arrested in 2012 (https://www.bbc.com/news/60554910, accessed on 15 August 2022). In this sense, FN becomes a useful weapon to mislead the enemy or populations about reality. However, this is only one of the most recent examples of FN. Not far in time, misinformation during COVID-19 pandemic caused what it has been called an *infodemic*. It provoked, among others, deaths due to direct consumption of alcohol and chloroquine-based detergents, ingested to avoid the risk of coronavirus infection (https://edition.cnn.com/2020/03/23/africa/chloroquine-trump-nigeria-intl/index.html, accessed on 15 August 2022) [2]. Looking further back in time, FN threatened our society misleading political events—such as the 2016 U.S. Presidential election or Brexit referendum [3].

The definition of FN is quite complex and a thorough account of the subject goes beyond this article purposes. In this article, we use a broad definition to cover all the different types of FN, i.e., false or misleading information presented as legitimate news which is (un)intentionally spread online by users. For a more consistent background on the subject, from different perspectives see [4–7].

Issues connected to FN are related to the speed at which FN spreads, the increasing amount of published FN, and the fact that FN Spreaders (henceforth, FNS) often add misinformation to verified news. In [8] authors discuss how FN exploit consumers' vulnerabilities triggering negative emotions and irrational reactions. To prevent FN from being spread among online users, a near real-time reaction is crucial. Due to these types of issues, human fact-checking and manual methods for FN detection are not feasible in terms of time and cost. Furthermore, in [9] the author shows that humans' ability to detect FN is only slightly better than chance. This skill is generally complicated by the confirmation bias phenomenon (i.e., believing information that confirms prior beliefs or values) [10]. Artificial intelligence methods are indeed needed.

In order to develop intelligent systems, expert-annotated news data collected on platforms such as Snopes (https://www.snopes.com, accessed on 15 August 2022) or PolitiFact (https://www.politifact.com, accessed on 15 August 2022) have been essential. The rising attention that FN has gained during the last years has led to the development of several datasets (see [11] for a survey), taxonomies stressing the importance of different concepts characterising it [6,12,13], and shared tasks for exploring how machine learning and Natural Language Processing (NLP) techniques can be employed to handle the FN detection issue.

In this article, we compare performances of recent Transformer-based architectures to several other machine learning models (deep and non-deep). In addition, we analyse the best performing model on the dataset for detecting FNS, and try to explain its behaviour guided by the dataset linguistic analysis and the outputs of the model layers. Investigating the responses, the analysis of the best performing model copes with the growing need of explaining and understanding the behaviour of modern AI tools [14,15]. Therefore, the main contributions of our study are:

- Analysing linguistic features of FNS and non-Fake News Spreaders (henceforth, nFNS) using corpus linguistics techniques;
- Comparing several State-of-the-Art (SotA) models to assess the impact of different architectures on the same dataset;
- On the basis of our comparative evaluation and our preliminary linguistic analysis, proving that large pre-trained models are not necessarily the optimal solution for the proposed task;
- Observing and investigating the behaviour of the best performing model (a shallow CNN) through a post-hoc analysis of the model layer outputs.

The remainder of this article is organised as follows: in Section 2 we briefly describe significant studies about FN and FNS. In Section 3, we present the models used for the comparative evaluation, the dataset, the preliminary analysis, and experimental setup to fully replicate our tests. In Section 4 we present and discuss the results of the evaluated models. In Section 5 we inspect how the best performing model operates on some

input samples, investigating and explaining its behaviour via the comparison with our preliminary linguistic analysis findings. In Section 6 we conclude the article.

All the code developed to run our tests is publicly available on GitHub (https://github.com/marco-siino/fake_news_spreaders_detection, accessed on 15 August 2022).

2. Related Work

The FN topic has been tackled in different ways: as a stance detection task (http://www.fakenewschallenge.org, accessed on 15 August 2022), as an information verification task (https://fever.ai/2018/task.html, accessed on 15 August 2022), or as an author profiling task (https://pan.webis.de/clef20/pan20-web/author-profiling.html, accessed on 15 August 2022)—the Profiling Fake News Spreaders on Twitter (henceforth, PFNSoT) shared task organised by PAN committee [16]—or as an issue related to other tasks, such as rumour detection [17,18], clickbait detection [19,20], and bot detection [21,22]. Tackling FN detection issue as an author profiling task puts the attention not just on the FN, which can change over time [8], but on the users sharing them, hence focusing on more stable features going beyond current FN topics. In this section we focus on related work that have approached FN from two different perspectives: as information verification task and as an author profiling task. First, we introduce the task of detecting FN, focusing on the most relevant shared tasks, methods and references. Second, we present recent studies on the detection of FNS. Also in this case, we address the most relevant international shared tasks, the best performing methods, and the main references.

It is worth mentioning a recent increase in the use of Explainable AI (XAI) methods in place of black box-based approaches for text classification tasks. A few of these methods are based on graphs [23] and are used in text classification as well as in traffic prediction [24], computer vision [25] and social networking [26]. Other interesting developments in text classification methods are the ones based on early and late fusion [27], on ensemble [28,29] and on data augmentation [30].

2.1. Fake News Detection

In decreasing time order, the most popular shared tasks on detecting rumours are the Constraint@AAAI2021-COVID19 Fake News Detection challenge [31], RumourEval-2019 [18] and RumourEval-2017 [17]. In [32] the authors present an interesting overview of the methods proposed by participants at the three above-mentioned tasks. According to the authors, the best performing models are often based on ensemble classifiers (e.g., SVM, LSTM, Logistic Regression, NN), CNN or BERT. The main focus of their article is a comparative evaluation of Transformers and other deep models for the task of detecting FN in Arabic. In [33] authors comparatively evaluate five machine learning algorithms: SVM, Naive Bayes, Logistic Regression and RNN models. On the dataset used, experimental results show that SVM and Naive Bayes outperform other methods. They do not report evaluation of transformers nor CNN. In a more extended evaluation, the authors in [34] evaluate seven machine learning models on three different datasets. The models used are based on Random Forest, SVM, Gaussian Naive Bayes, AdaBoost, KNN, Multi-Layer Perceptron and Gradient Boosting Algorithm. In terms of accuracy and F1 score, the Gradient Boosting Algorithm outperforms the other proposed models. However, also in this study, more experiments on deep models are missing. To address FN detection, also non-textual information can be exploited. An interesting study is conducted in [35], where authors propose a model for an early detection of FN on websites by classifying news propagation paths.

To the best of our knowledge, a comparative study on several cutting-edge models (deep and non-deep) based on textual information is missing. However, a relevant survey is presented in [8] where existing algorithms, evaluation metrics and representative datasets are discussed. Finally, it is worth mentioning a study conducted in [36]. There, authors compare performances of humans and automatic classifiers for the task of detecting FN.

2.2. Fake News Spreaders Detection

As reported in the previous section, the detection of FN tackled as verification task has already received considerable research attention. However, there are only few studies that have addressed the problem from a user or author profiling perspective. The rationale behind this perspective is based on the hypothesis that with these profiling tools we could identify FNS in order to raise their awareness on FN. In addition, it is possible to investigate if nFNS have different characteristics compared to FNS. For example, nFNS may use different linguistic patterns when they share posts compared to FNS.

A recent shared task about FNS profiling is the one discussed in [16]. The task, hosted at PAN@CLEF2020, focuses on determining whether or not the author of a Twitter feed is keen to spread FN. A multilingual corpus (i.e., English and Spanish datasets) with Twitter data was provided by the organisers. Participants at the task used traditional approaches, mainly SVM, Logistic Regression or a combination of both depending on the language. Random Forest was the third most used classification algorithm. Ensembles of classifiers were used by various authors. Furthermore, models based on Decision Tree, Random Forest and XGB, Random Forest and Naive Bayes with XGBoost, NN with Dense layer and ensemble of a GRU-based aggregation model with CNN were proposed. The best result has been obtained in *ex aequo* by an SVM-based model and a Logistic Regression ensemble. In particular, the former is based on combinations of character and word n-grams and SVM [37]. The latter, is a Logistic Regression ensemble of five sub-models: n-grams with Logistic Regression, n-grams with SVM, n-grams with Random Forest, n-grams with XGBoost and XGBoost with features based on textual descriptive statistics, such as the average length of the tweets or their lexical diversity [38]. However, only a few participants experimented with more deep learning approaches (e.g., Fully-Connected NN, CNN, LSTM, or Bi-LSTM with self-attention).

In [39] the authors extend the CoAID dataset [40] to address the task of automatic detecting FNS or nFNS of COVID-19 related news. The authors present a stacked and Transformer-based NN that combines the Transformer capabilities of computing sentence embeddings with a deep learning model. In [41], the authors use feed psycholinguistic and linguistic features into a CNN model to profile FNS and nFNS. The experimental results show that their proposed model is effective to classify a user as FNS or nFNS. The authors compare their results on a dataset specifically built for their task. However, the only Transformer used is BERT and deep models are not well-explored. In addition, their proposed model has been tested in [42] on the same dataset we used here (i.e., the one provided for PAN@CLEF2020). They report poor results. Specifically, the model tested reaches a binary accuracy of 0.52 and of 0.51 on the English and Spanish dataset, respectively. In their article [42], they propose a new model that uses personality information and visual features, outperforming the two winning models at PAN@CLEF2020 on both languages.

To the best of our knowledge, our study is the first that compares several SotA Transformers models, and other deep and non-deep models. In addition, we propose the best performing model on the multilingual dataset provided for PAN@CLEF2020. Furthermore, we provide a linguistic analysis of the dataset in order to have a eye-bird view of the data we are feeding to each model. Then we analyse post-hoc the features attentioned by the best performing deep model to classify FNS and nFNS. Based on our results, in contrast to those reported in the literature, Transformers are not the best choice for FNS profiling on this specific task.

3. Materials and Methods

In this section we present the architectures of the models presented, the dataset analysis and the experimental setup.

3.1. Models Architectures

In this subsection we briefly introduce the architectures used in our experiments. Pretrained models are shortly discussed referencing the specific pretrained version. Dealing

with a multilingual dataset, we select the appropriate pretrained versions for English and Spanish.

- BERT. Presented in [43], BERT is one of the first language representation model presented. It is designed to pre-train bidirectional representations, by jointly conditioning on both left and right context, starting from unlabeled text. The model is pretrained using two objectives: (1) Masked language modeling (MLM): taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words, (2) Next sentence prediction (NSP): the models concatenates two masked sentences as inputs during pretraining. Sometimes they correspond to sentences that were next to each other in the original text, sometimes not. The model then has to predict if the two sentences were following each other or not. The model can be fine-tuned with just one additional output layer depending on the task. For the English dataset we implemented the original *bert-base* presented in [43] while for the Spanish dataset we used BETO, the pretrained Spanish version discussed in [44].
- DistilBERT. Given the interesting results obtained in [45] we implemented for our task a DistilBERT [46] model. DistilBERT is a method to pre-train a general-purpose language representation model. The result is a smaller model if compared to BERT. Thanks to a distillation process, in DistilBERT the size of a BERT model is reduced by 40%, while retaining 97% of its language understanding capabilities and being 60% faster. For the English dataset we implemented the original *distilbert-base* presented in [46] while for the Spanish dataset we used the pretrained version extracted from *distilbert-base-multilingual-cased* and discussed in [47].
- RoBERTa. Presenting a replication study of BERT pre-training, authors in [48] improve the performances of BERT operating modifications to the pretrain phase of a BERT model. These modifications include: (1) training the model longer, with bigger batches; (2) removing the next sentence prediction objective; (3) training on longer sequences; and (4) dynamically changing the masking pattern applied to the training data. For the English dataset we implemented the version of RoBERTa presented in [48], while for the Spanish dataset we used the version of RoBERTa pretrained with a total of 570 GB of clean and deduplicated text. The text is compiled from the web crawlings performed by the National Library of Spain (Biblioteca Nacional de España) from 2009 to 2019 and discussed in [49].
- ELECTRA. Presented in [50], instead of masking the input as in BERT, ELECTRA implies replacing some tokens with plausible alternatives sampled from a small generator network. Then, instead of training a model that predicts the original identities of the corrupted tokens, a discriminative model is trained to predicts whether each token in the corrupted input was replaced by a generator sample or not. For the English dataset we implemented the original version presented in [50], for the Spanish dataset we used an ELECTRA model trained on the same Large Spanish Corpus used in BETO.
- Longformer. Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length. To address this limitation in [51] authors introduce the Longformer. Thanks to an attention mechanism that scales linearly with sequence length, processing documents of thousands of tokens or longer should be easier. Longformer uses a combination of a sliding window (local) attention and global attention. Global attention is based on the task to allow the model to learn task-specific representations. For the English dataset we used the original pretrained version presented in [51], for the Spanish dataset we implemented the version pretrained on: (1) Encyclopedic articles from Wikipedia in Spanish, (2) News from Wikinews in Spanish, (3) Texts from the Spanish corpus AnCora (http://clic.ub.edu/corpus/en, accessed on 15 August 2022), which is a mix from different newswire and literature sources.

- XLNet. The approach proposed in [52] is a generalised autoregressive pretraining method. It enables learning bidirectional contexts by maximizing the expected likelihood over all permutations of the factorization order. On several tasks (including question answering, natural language inference, sentiment analysis, and document ranking) XLNet outperforms BERT, often by a large margin. A well-established XLNet pretrained on a Spanish corpus is missing. So in our study we implemented the same pretrained XLNet for both datasets. Evaluating, in this case, a zero-shot cross lingual transfer [53].
- CNN. The shallow CNN tested is based on the one presented in [54] and is shown in Figure 1. This CNN is a novel architecture developed and tuned specifically for this work. In the PAN2021 author profiling task, a very similar architecture was able to win the challenge, ranking first against over 60 participating teams (https://pan.webis.de/clef21/pan21-web/author-profiling.html, accessed on 15 August 2022). The CNN is based on a word embedding layer, a single convolutional layer, a max pooling layer, a dense layer, a global average pooling layer and a final single dense unit layer. As proved by its results, on a similar binary classification task, the model is able to outperform Transformer-based models and several other deep and non-deep model as reported in [55].
- Multi Channel CNN. To further investigate the interesting results obtained by a shallow CNN in [54], we tested a multi channel CNN similar to the one proposed in [45]. Thanks to parallel channels, consisting of word embedding, convolutional and max pooling layers, the model is able to capture different-sized windows of ngrams compared to the single layer and single filter size of the shallow CNN tested here.
- SVM. Based on [56], we tested the *sklearn* SVC implementation (https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html, accessed on 15 August 2022). We used a linear kernel type with a value of 1.0 as regularization parameter.
- Naive Bayes. As firstly discussed in [57] and as empirically proved along the years by the interesting results obtained on several text classification tasks, we implemented a Multinomial Naive Bayes classifier from *sklearn* MultinomialNB implementation (https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html, accessed on 15 August 2022). MultinomialNB implements the naive Bayes algorithm for multinomially distributed data where data are typically represented as word vector counts.

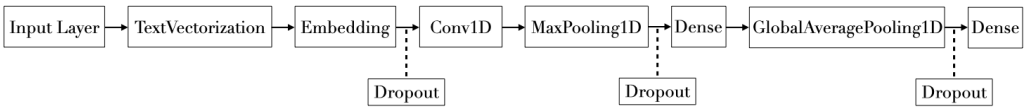


Figure 1. The shallow CNN model used in our experiments. The network is able to outperform any other existing approach on the same dataset and is further analysed in our study.

3.2. Dataset Analysis

Although there are currently several datasets containing fake news spreader feeds, to conduct our study we needed a dataset with a strong baseline, to verify the consistence of the results obtained in our study. Using the PFNSoT dataset we were able to detect that, in fact, responses of the models evaluated here are consistent with some of the participants at the PFNSoT task. By integrating other datasets into our work we would not have been able to deal with such a high number of model architectures as those presented at PFNSoT. In this way we were able to conduct our model and linguistic analysis on consistent results. Eventually, a multilingual dataset as the PFNSoT one, possibly allowed to us the detection of common traits of FNS using different languages.

The PFNSoT is a multilingual dataset made of Spanish and English tweets. For each language, the data collected is made of 100 tweets per author and 150 authors per class (i.e., FNS and nFNS) in the training set, and 100 authors per class in the test set. We decided to use the PFNSoT dataset for two main reasons: PAN has a long tradition in organizing

shared tasks; our extensive tests on several SotA models are in this way comparable with the other participants' results.

Although task organisers encouraged the submission of multilingual models, submissions of models dealing only with one language were also allowed. As reported in the task overview, participant results were lower for the English language in terms of binary accuracy. To know more about the dataset, we quantitatively and qualitatively investigated it using established corpus linguistics methods, implemented in the online well-known corpus linguistics tool, Sketch Engine [58] (https://www.sketchengine.eu, accessed on 15 August 2022).

3.2.1. Compare Corpora

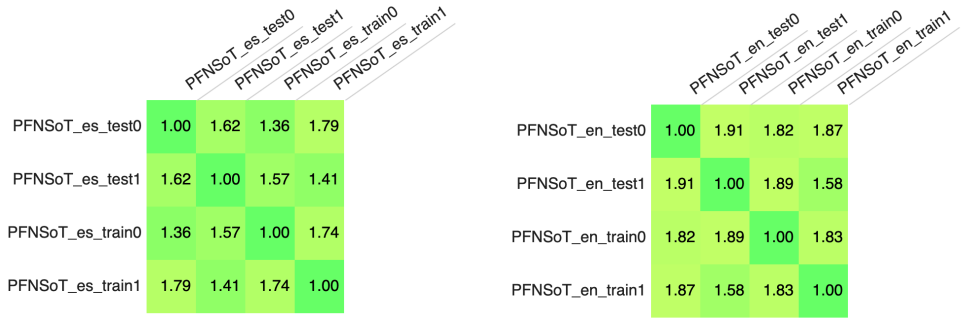
In this subsection we report a quantitative description of the Spanish and English datasets. Since we used corpus linguistics tools to carry out the analysis, in this section we use the term corpus (plural, corpora) to refer to each dataset. Table 1 provides an high-level description of the Spanish and English corpora, in terms of number of tokens identified in the tweets written by the same typology of authors. The corpora are also divided into subcorpora, class-wisely grouping tweets released as training and test data. In the table, each corpus is labelled by specifying the language, the class and the partitioning criterion of the corpus. For example, the corpus es_train_0 collects the Spanish tweets (**es**_train_0) contained in the training set (es_**train**_0) written by nFNS authors (es_train_**0**), while es_1 to the totality (training and test sets) of tweets in Spanish written by FNS. While in the Spanish corpus there is a relevant difference in size between corpus 0 and corpus 1—and this difference in size is kept also in the training and test data—it is not the case in the English dataset, in which the two classes have almost the same number of tokens both in train and test data. However, the difference in size in the Spanish dataset is not as big as to prevent corpus comparison in terms of common tokens (i.e., similar linguistic register used by the authors). For this comparison, we applied a chi-square (X^2) test [59] by using the built-in function of Sketch Engine, Compare Corpora. Thus, we compared train_0, train_1, test_0, and test_1 of both languages. In this way, we obtained two confusion matrices, reported in Figure 2, showing values greater of or equal to 1, with 1 indicating identity. The higher the value, the larger the difference between the two compared subcorpora.

Table 1. Dataset summary.

Subcorpus Name	# Tokens	Percentage	Total
es_0	832,755	53.71%	1,550,505
es_1	717,750	46.29%	
en_0	669,519	50.57%	1,323,982
en_1	654,463	49.43%	
es_train_0	500,003	54.04%	925,152
es_train_1	425,149	45.96%	
en_train_0	402,788	50.92%	791,024
en_train_1	388,236	49.08%	
es_test_0	332,752	53.21%	625,353
es_test_1	292,601	46.79%	
en_test_0	266,731	50.04%	532,958
en_test_1	266,227	49.96%	

Spanish Corpus Matrix. We assumed 1.74 as reference measure for all the other comparisons, since it indicates the difference between train_0 and train_1, i.e., the data that models use for training. As reported in this matrix, the similarity measure between test_0 and train_0 is 1.36, which is 0.38 points smaller than the reference measure. The same applies to test_1 and train_1: their similarity measure is 1.41, which is 0.33 points smaller than the reference measure. The fact that the difference between the reference

measure and the class-wise train and test similarity measure is a bit higher in nFNS might indicate that FNS are slightly more difficult to identify. In addition, it is worth noticing that, since the similarity measure between train_0 and test_1 (i.e., 1.57) is smaller than the reference measure we assumed, this also might support the idea that FNS authors will be more difficult to identify than nFNS authors (in contrast, train_1 and test_0 similarity measure is 1.79, which is bigger than the reference measure, 1.74).



(**a**) (**b**)

Figure 2. Comparing English and Spanish corpora: confusion matrices obtained with the chi-square test. The value 1.00 indicates identity between the compared subcorpora. The greater the value, the more different the subcorpora. (**a**) Spanish dataset. (**b**) English dataset.

English Corpus Matrix. In this matrix the reference measure given by the difference between train_0 and train_1 is 1.83. While the difference between train_1 and test_1 is below this value (i.e., $1.58 < 1.83$)—although with a smaller gap than the same difference in the Spanish dataset (Spanish: 0.33, English: 0.25)—the similarity measure between train_0 and test_0 differs from the reference measure by just 0.01—in the Spanish dataset is 0.38. This might suggest that systems may have more troubles in identifying nFNS. However, if we look at the difference between train_0 and test_1 and train_1 and test_0, we have similarity measures of 1.89 and 1.87, respectively, which are both slightly higher than the reference measure.

Comparing what emerged from these matrices and the error analysis carried out in [16], we noticed that our hypotheses are consistent with the aggregated task participant results. In the Spanish corpus, according to their confusion matrix, nFNS were predicted correctly 80% of the time, while FNS only 65% of the time, confirming *de facto* that FNS were harder to identify than nFNS in this corpus as indicated in our matrix (Figure 2a). In the English corpus, they reported a higher confusion from nFNS towards FNS, with nFNS correctly predicted 64% of the time and FNS 70%, confirming again what emerged from the matrix in Figure 2b. In addition, the fact that systems performed better on the Spanish corpus could be explained by a similarity measure nearer to 1 (i.e., indicating a higher similarity between the training set of that class and the correspondent test set) than that of the English corpus. These matrices obtained comparing corpora on Sketch Engine, then, might be useful to predict system errors in various corpora. However, looking only at these matrices, it is not possible to state *what* differs between the corpora. Then, we used other Sketch Engine facilities to gain insight into what actually differs between them.

3.2.2. Keywords

Despite the term *keyword* is widely used outside corpus linguistics, in this field it is used for quantitatively highlight trends in the corpora. Specifically, through keyword analysis it is possible to retrieve tokens that are statistically characteristic of a (sub)corpus when comparing it with another (sub)corpus (see [60] for a comprehensive exposition of the subject). For both language corpora, we used Keywords to identify what distinguishes the two classes. To do so, we used once FNS as focus corpus and nFNS data as reference corpus, and *vice versa*. In this way, we pointed out focus corpus *keywords* as compared to

the reference corpus. Keywords in Sketch Engine are sorted according to their Keyness score, which is calculated as shown in Equation (1). In the expression, fpm_{focus} stands for normalised per million frequency of the word in the focus corpus, fpm_{ref} stands for normalised per million frequency of the word in the reference corpus, and N indicates the simplemath parameter, which is used to handle words that only occur in the focus corpus and not in the reference corpus (avoiding the problem of dividing by zero), and to decide whether to give importance to more frequent words or to less frequent words. In fact, different values of simplemath can be used to sort the keywords in the list differently. Generally, higher values of simplemath rank higher more common words; lower values of simplemath rank higher more rare words [61]. We decided to focus on core-vocabulary words, neither so rare nor so common, setting the simplemath parameter to 100. In Table 2 we report the first 50 keywords of both corpora.

$$Keynessscore = \frac{fpm_{focus} + N}{fpm_{ref} + N} \quad (1)$$

Spanish Corpus Keywords. Focusing on the authors labelled as nFNS (corpus 0 as focus) and FNS (corpus 1 as focus), we extract keywords which are used differently by the two groups of users (it is possible also that some tokens do not occur in both subcorpora). Based on these keywords and inspecting the linguistic context (i.e., co-text) in which they occur (using the Sketch Engine Concordance facility), we observed that nFNS (corpus 0) share information about technology (4, 9, 10, 15, 17, 19 'mobile' but also referred only to mobile phones, 29 'screen', 35 'users'), FN (14), toponyms (13, 18, 24–43), politics (20, 32), warnings (8, 11). Conversely, FNS (corpus 1) share information about mostly Latin American artists, music and related (5 'premiere', 8–4, 9, 11–13, 15, 17, 29–39, 41, 45, 46, 47, 48–20, 50), videos (2, 3, 10, 19), shocking or last minute news (5, 7–6, 18–22, 35–28, 37), and also galvanize users to get involved (1 'join us', 14 'download', 23, 31 '2ps-forget', 34 'share it').

In addition, it is worth noting the way in which the two groups use capitalization. While focusing on nFNS, keywords are well-written and capitalization is used in a standard manner (with some exceptions specific to the medium of communication, i.e., Twitter), when we look at FNS keywords, we notice misspellings (missing accents in 1, 4, 7, 18, 35), Latin American spelling (2, 3) and much more capitalised words. This led us to decide to keep capitalization during the preprocessing phase.

English Corpus Keywords. Based on the keywords reported in Table 2, and looking at their co-text, we observed that nFNS talk about TV shows and related (2, 3, 4, 7, 11, 28, 29, 36, 43), fashion and related (12, 17, 20, 24, 25, 27), and invite to action (6, 18, 31). FNS, conversely, write about politics (2, 3, 4, 5, 6, 7, 13, 21, 23, 24, 30, 37, 40), famous people and gossip (1, 9, 12, 14, 17, 27, 28, 31, 35, 39), entertainment (19-28, 29), but also warnings about FN (8, 11, 15).

Table 2. Spanish and English corpora—Keywords.

Spanish Corpus First 50 Keywords of nFNS—Corpus 0 as Focus and Corpus 1 as Reference									
1	T	11	PRECAUCIÓN	21	qué	31	seguridad	41	información
2	HASHTAG	12	tuit	22	to	32	PodemosCMadrid	42	esa
3	Buenos	13	Albacete	23	added	33	hemos	43	Mancha
4	Android	14	bulos	24	Castilla-La	34	han	44	sociales
5	h	15	Google	25	Pues	35	usuarios	45	Os
6	he	16	artículo	26	sí	36	servicio	46	cómo
7	sentido	17	Xiaomi	27	Albedo	37	RT	47	Nuevos
8	RECOMENDACIONES	18	León	28	algo	38	datos	48	pruebas
9	Samsung	19	móvil	29	pantalla	39	os	49	Gracias
10	Galaxy	20	Cs_Madrid	30	disponible	40	playlist	50	creo

Table 2. *Cont.*

Spanish Corpus First 50 Keywords of FNS—Corpus 1 as Focus and Corpus 0 as Reference									
1	Unete	11	Lapiz	21	Dominicana	31	OLVIDES	41	Concierto
2	VIDEO	12	Vida	22	Fuertes	32	Joven	42	Acaba
3	Video	13	Conciente	23	Follow	33	Años	43	Muere
4	Clasico	14	DESCARGAR	24	DE	34	COMPÁRTELO	44	Hombre
5	ESTRENO	15	Mozart	25	Su	35	IMAGENES	45	Secreto
6	MINUTO	16	De	26	Descargar	36	Le	46	ft
7	ULTIMO	17	Ft	27	añadido	37	IMPACTANTE	47	Preview
8	Mayor	18	Imagenes	28	FUERTES	38	Accidente	48	lista
9	Alfa	19	Official	29	Don	39	Miguelo	49	Republica
10	Oficial	20	reproducción	30	Del	40	Remedios	50	Omega
English Corpus First 50 Keywords of nFNS—Corpus 0 as Focus and Corpus 1 as Reference									
1	Via	11	Synopsis	21	Tie	31	Check	41	isabelle
2	Promo	12	Styles	22	qua	32	Academy	42	AAPL
3	Review	13	Lane	23	Bayelsa	33	Ankara	43	fashion
4	Episode	14	GQMagazine	24	du	34	rabolas	44	Date
5	PHOTOS	15	Mariska	25	Robe	35	PhD	45	esme
6	Read	16	Hargitay	26	NYFA	36	Spoilers	46	isla
7	Actor	17	Nigerian	27	Tendance	37	DE	47	Marketing
8	TrackBot	18	READ	28	Supernatural	38	story	48	Link
9	RCN	19	br	29	Film	39	Draw	49	prinny
10	AU	20	beauty	30	Bilson	40	University	50	your
English Corpus first 50 Keywords of FNS—Corpus 1 as Focus and Corpus 0 as Reference									
1	Jordyn	11	ALERT	21	Schiff	31	tai	41	Price
2	realDonaldTrump	12	Grande	22	InStyle	32	Him	42	Says
3	Trump	13	Biden	23	Democrats	33	Her	43	post
4	Donald	14	Meghan	24	Trump's	34	Twitter	44	About
5	Hillary	15	NEWS	25	His	35	Markle	45	rally
6	Obama	16	published	26	After	36	Jonas	46	BUY
7	Clinton	17	Ariana	27	Reveals	37	border	47	Bernie
8	FAKE	18	Webtalk	28	Snoop	38	Khloe	48	Tristan
9	Woods	19	Viral	29	Thrones	39	Scandal	49	tweet
10	RelNews	20	added	30	Border	40	Pelosi	50	FBI

Differently from what emerged from keyword analysis in the Spanish corpus, in the English corpus it is not predictable to which class the first 50 keywords belong. In addition, tweets about FN alerts should not be in FNS data.

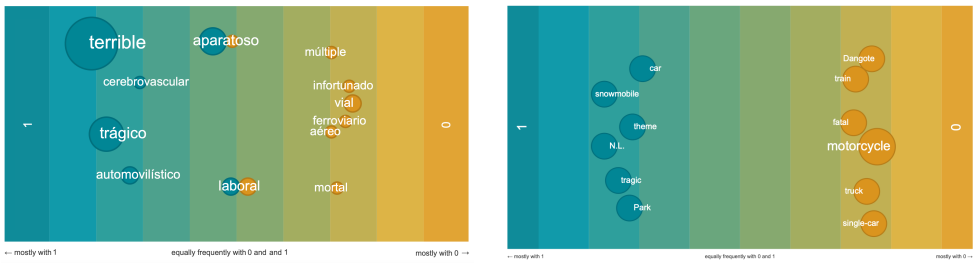
3.2.3. Word Sketch Difference

One of the original features of Sketch Engine is the possibility of outlining the behaviour of a word in a corpus using the Word Sketch facility. With its extension, called Word Sketch Difference, it is possible to compare two words observing differences in use or to compare how the same word is used in two different corpora. We used Word Sketch Difference to see how the same word is used by the two groups (i.e., FNS and nFNS) in the two corpora (i.e., English and Spanish datasets). We looked at the modifiers of the word *accident*, *accidente* in Spanish, because it is a word occurring in the two corpora and in the two classes, and because we expected a different use by the two groups which should not be due just to frequency. In Table 3 we report all the modifiers associated to *accidente* and *accident*, taken as lemma. In Figure 3, we show the distribution of their modifiers in the Spanish (Figure 3a) and English (Figure 3b) corpora. In both figures, on the left, the image shows the modifiers which are mostly associated with the selected lemma in FNS tweets; on the right, those associated to the lemma in nFNS tweets; in the middle, those employed by both groups (empty in the English corpus). The bigger the circle, the higher the frequency. In the Spanish corpus, even though in FNS tweets *accidente* occurs more, it

is associated mostly with two connotative modifiers (*terrible* and *trágico*). It is interesting to notice a correlation between the modifiers of *accident* in the English corpus with those used in the Spanish one: the use of *tragic* (in Spanish *trágico*) occurring in FNS subcorpus, while *fatal* (in Spanish *mortal*), and vehicles defining the type of accident, occurring in nFNS subcorpus. The presence of these modifiers might indicate that more subjective language is used in FNS data—as *trágico*, *terrible* and *tragic* suggest—while, in nFNS, the news about the accident seems to be reported in a more objective way.

Table 3. Modifiers of *ACCIDENTE* and *ACCIDENT* in the corpora.

Spanish Corpus			English Corpus		
Modifiers	**nFNS**	**FNS**	**Modifiers**	**nFNS**	**FNS**
vial	2	0	single-car	1	0
infortunado	1	0	Dangote	1	0
ferroviario	1	0	motorcycle	2	0
mortal	1	0	truck	1	0
aéreo	1	0	train	1	0
múltiple	1	0	fatal	1	0
grave	1	0	car	0	1
laboral	2	2	theme	0	1
aparatoso	1	5	Park	0	1
propio	0	2	tragic	0	1
cerebrovascular	0	1	snowmobile	0	1
automovilístico	0	2	N.L.	0	1
trágico	0	8			
terrible	0	19			



(**a**) (**b**)

Figure 3. Visualization of modifiers of *accidente* and *accident* in the Spanish and English corpora, respectively. (**a**) Spanish corpus. (**b**) English corpus.

3.3. Experimental Setup

All our tests were performed on Google Colab with TensorFlow on NVIDIA Tesla K80 GPUs. For the pretrained models we used the *Simple Transformers* (https://simpletransformers.ai/about/, accessed on 15 August 2022). Each transformer implemented come from the transformers library presented here [62]. Batch size is equal to 1 for all models. We fine-tune the pre-trained models for 10 epochs performing early stopping accordingly to the binary accuracy on the test set. The non-pretrained deep models (CNN and MultiCNN) are trained for 100 epochs. For the evaluation of each model performances we adopt the protocol used in [48], i.e., we execute five random initialisation (with early stopping for each run) reporting the median as model result. With this protocol we were able to select the best result obtained by each model for each run, reporting the median as the most representative value for future and

generic executions of our code. The results of our experiments can be explored looking at the raw files of the notebook hosted on GitHub. For each model we already reported references to the original implementation with the experimental setup of every architecture. The dataset we used is presented in [16] and available under request.

4. Results and Discussion

In accordance to the official metric used for the shared task, in this study we evaluate each model using binary accuracy defined as follows:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{2}$$

where *tp* stands for *true positive* over the positive class (i.e., FNS). Since the dataset is class-balanced, accuracy defined as in (2) is an effective metric for the binary classification task of classifying a user as FNS or nFNS.

In Figure 4 is reported the average accuracy of each model evaluated. AVG accuracy is calculated averaging the accuracy in each language (Spanish and English). In Table 4 detailed results of the experiments are reported. As already discussed, the binary accuracies reported are the median over five random initialzations on the test set, except for SVM and Naive Bayes, because their deterministic nature allow to report accuracy in a single run. For the same reason standard deviation of these two models is not reported in the table. The standard deviation is calculated using the five accuracies over the five random initialisations. This metric further provides information on the ability of each model to replicate constant results over several runs. As hypothesised during our preliminary linguistic analysis, performances over the English test set are worse compared to the Spanish test set for all the models evaluated. The results indicate the shallow CNN as the best performing model on both test sets (English and Spanish), and the one achieving the smallest standard deviation on the Spanish test set. The smallest standard deviation over the English test set is obtained by the Multi-CNN. Transformer-based models generally perform worst in terms of standard deviation. It is interesting that the linear SVM is able to outperform any Transformers on the Spanish test set, but ELECTRA. On the other hand, Naive Bayes on the English Test Set, is able to perform better (or equal, compared to RoBERTa) of any Transformer evaluated. Given the size of each sample in the dataset, results are in line with those reported in other studies, e.g., [63]. As far as Longformer is concerned, we expected a better performance from it. It is worth bearing in mind that each sample within train and test sets contains a feed of the last 100 tweets of a single user. This size would perfectly fit the information that a Longformer can manage. However, the results suggest that this is not enough to capture relevant user features based on the whole thread of each user. These low performances could be motivated by the fact that the user is not represented with a long consistent text. The 100-tweet sequence per user cannot be considered as a text (i.e., a coherent stretch of language), because each tweet is not always and directly linked to the previous and to the next one. To this content fragmentation could be due the poor performances of the Longformer. Contrarily to Transformers, for the CNN, fragmentation could be a positive feature. In fact, the 100-tweet thread per user could be seen as a picture composed by 100 different parts, each representing an aspect—represented in 280 pixels (because their longest sequence in tweets is 280 characters)—of the full picture. Some users are more diverse than others, depending on the variety in their feed. Since CNN filters are able to scan each content window and focus on the relevant features, it is not surprising that they are able to cope well with image classification/recognition, which in our opinion is comparable to the content fragmentation we have in this task. Apart from this, from our experimental evaluation it emerges that non-deep models are not a second choice compared to Transformers. In fact, a simple ensemble of Naive Bayes and SVM models could achieve better performances than Transformers on this and on similar binary classification tasks as reported in Section 2.

In Table 5 the results of the PAN@CLEF2020 task are reported. As already discussed in Section 2 none of the winners implemented deep models and as reported in [16] a very small number of participants submitted Transformer-based models. It is worth noting that the best performing model evaluated here (CNN) over the five random initialisations is able to reach, on the best run, a binary accuracy of 0.760 for English and 0.820 for Spanish, outperforming any other submitted model at PAN@CLEF2020.

Table 4. Models accuracies and standard deviation. For non-deterministic models accuracy is the median over five runs. In the table, the best results are shown in bold.

	English		Spanish	
	Acc	σ	**Acc**	σ
CNN	**0.715**	0.022	**0.815**	**0.005**
Multi-CNN	0.545	**0.004**	0.670	0.013
BERT	0.625	0.036	0.735	0.018
RoBERTa	0.695	0.014	0.735	0.024
ELECTRA	0.630	0.016	0.760	0.015
DistilBERT	0.645	0.016	0.725	0.014
XLNet	0.675	0.020	0.710	0.070
Longformer	0.685	0.041	0.695	0.007
Naive Bayes	0.695	-	0.695	-
SVM	0.630	-	0.755	-

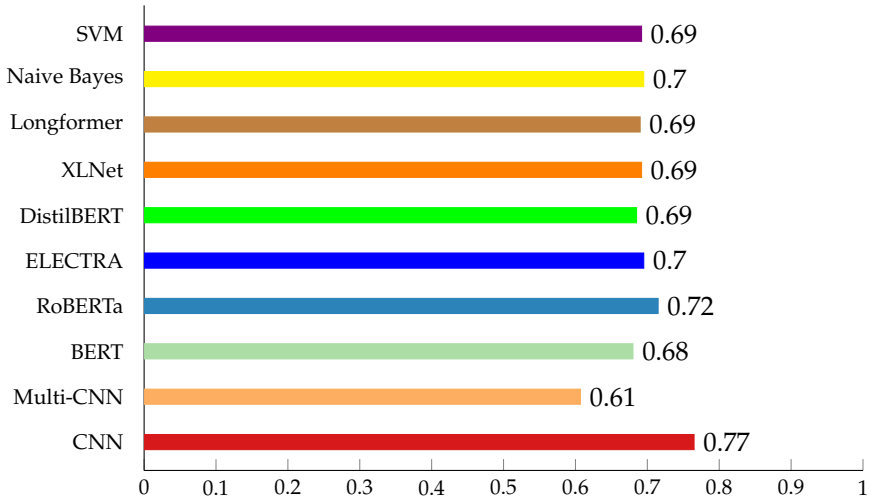


Figure 4. Average accuracy of each model evaluated. AVG accuracy is calculated averaging the accuracy in each language (Spanish and English).

Table 5. PAN@CLEF2020 Results (https://pan.webis.de/clef20/pan20-web/author-profiling.html, accessed on 15 August 2022). Firsts three positions plus baselines provided by organizers.

Position	Team	English	Spanish	AVG
1	bolonyai20	0.750	0.805	0.777
1	pizarro20	0.735	0.820	0.777
-	SYMANTO (LDSE)	0.745	0.790	0.767
3	koloski20	0.715	0.795	0.755
3	deborjavalero20	0.730	0.780	0.755
3	vogel20	0.725	0.785	0.755
-	SVM + c nGrams	0.680	0.790	0.735
-	NN + w nGrams	0.690	0.700	0.695
-	LSTM	0.560	0.600	0.580
-	RANDOM	0.510	0.500	0.505

5. Post-Hoc Model Analysis

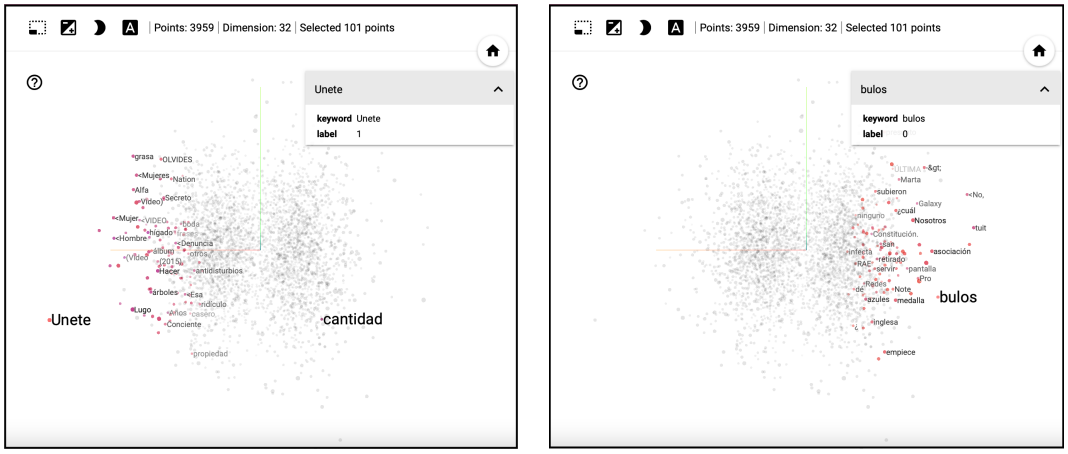
In Section 3.2, we observe that keywords are good indicators to distinguish the two FNS and nFNS classes, as corroborated also by the results of the Bayesian model reported in Table 4. However, the CNN-based model must go beyond these frequency differences as its results suggest. In this section, we provide a post-hoc analysis of intermediate model outputs, devised to shed light on the CNN behaviour. In particular, we analyse the outputs of three hidden layers: embedding layer, convolutional layer and global average pooling layer. These are the model layers that, can be analysed by relating the outputs to the inputs to better understand the overall classification decision. Although hybrid approaches have been exploited for eXplainable AI [64], the CNN tested here can be defined as a shallow neural model. Thus, it can be analyzed mapping each layer outputs to its inputs.

5.1. Word Embedding Layer Output

After the training, we visualised in the Embedding projector two clearly distinguishable clusters, as reported in Figure 5a. To verify how these two clusters are related to the two classes, we labelled the words represented there. To do so, we extracted 3959 keywords using our Bayesian model—precisely, we extracted 1980 most frequent tokens in corpus 0 and 1979 most frequent tokens in corpus 1—and labelled them accordingly. Then, we visualised them in the embedding space of the trained CNN model, as shown in Figure 5b. Note that we used key tokens retrieved by the Bayesian model and not those obtained using Sketch Engine, because the former has the same tokenization of the CNN model. We excluded tokens occurring in both corpora 0 and 1. Figure 5b confirms that the two clouds are closely related to the two task classes. Red dots refers to FNS, blue dots to nFNS. Exploring these clouds, we can find some of the keywords also identified using Sketch Engine Keywords (cfr. Table 2). In Figure 6a,b, we highlighted *Unete* as FNS keyword and *bulos* as nFNS keyword. Apart from *Unete*, in Figure 6a we can find other keywords individuated in the preliminary analysis conducted in Section 3.2. Of course, since Sketch Engine tokenization differs from that of the CNN model, we cannot have a one to one correspondence. While, for example, following the standard tokenization in Sketch Engine we can distinguish cased and uncased letters, it is not the case with punctuation, which is always kept apart. In the embedding space, we can notice that the tokens with a higher keyness score are positioned farther than the other cluster (see for example *Unete* in Figure 6a). Thus, this could suggest that in the embedding space tokens are located according to their keyness score.

(**a**) (**b**)

Figure 5. Word embedding as visualised in a 3-dimensional space. (**a**) Unlabelled word embedding space (75,999 points). (**b**) Labelled word embedding space (3959 points).



(**a**) (**b**)

Figure 6. Visualization of FNS and nFNS keywords in the labelled embedding space. (**a**) Label 1. (**b**) Label 0.

5.2. Convolutional Layer Output

The output of each filter of the convolutional layer was searched for finding maximum and minimum values in the output tensor. Our hypothesis is that these values correspond to some tweets, captured by the filter window, showing some of the relevant linguistic features we found during our preliminary analysis. Reverse mapping the input tokens corresponding to the filter window, we identified the 32-token windows with maximum and minimum values assigned. The 32-token windows receiving the maximum value are those considered important by the convolutional layer filters and, consequently, pass also the max pooling layer. Thus, we randomly downloaded 15 samples per class (10% of the train) together with the 32-token windows with the maximum and minimum values assigned. These 32-token windows consist of maximum three complete tweets. We noticed that the majority of the 32 filters outputted a maximum or minimum value for the same windows of tokens (with a variation of a few tokens) per author sample. This behavior suggests that a lower number of filters could have been enough for capturing the token patterns which are more relevant for classifying an author as FNS or nFNS. We observed that giving as input the whole collection of 100 tweets per author produced two or three peaks in the filter output that are clearly distinguishable from the other local maximum values. An example of output corresponding to the complete filtering of a reference author found by the first convolutional filter is graphically shown in Figure 7. The document—i.e., the author's 100 tweets—consists of about 2000 tokens, then it is padded up to 4060. The output of this filter shows a global maximum in position 1739, indicating that in that 32-token window there are relevant features. To see what this window contains, we looked at our vocabulary and did a reverse mapping. We applied this procedure to all the windows with maximum and minimum valued tokens, allowing an analysis of the linguistic features that the best performing model considers most or less important when classifying the sample.

Analysing FNS and nFNS 32-token windows considered important by the filters, we found some patterns, reported below, corresponding to specific topics and tweet style, such as the usage of the first person or the formulation of a question.

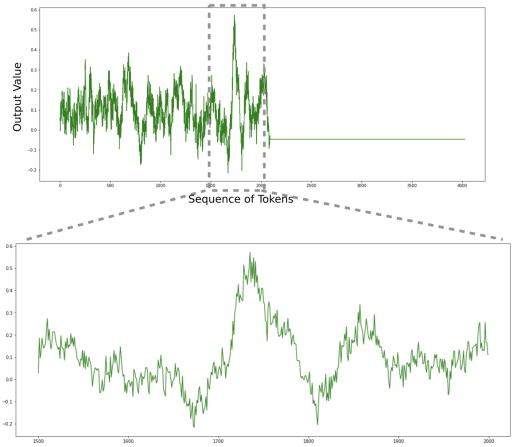


Figure 7. Output of the first convolutional layer after convolving one of the 32 filters over the input provided. The maximum value corresponds to the token in position 1739 and the minimum corresponds to the token in position 1673. The sample shown consists of less than 1500 tokens, hence the document is padded up to 4060.

FNS Patterns. We found both features in accordance with our preliminary analysis and not. On the one hand, in these windows of FNS samples, we found information about: 1. tricks, miracle foods or home remedies (e.g., *El truco para secar la ropa sin necesidad de tenderla—VÍDEO #URL#*, 'The trick to drying clothes without hanging them out to dry'); 2. sensitive (o strong) images or videos (e.g., *FUERTE VÍDEO—Matan Hombre Por Violar Niñas #URL# #URL#*, 'STRONG VIDEO—Man Killed For Raping Girls'); 3. music (e.g., *Chimbala anuncia union entre algunos dembowseros para cambiar el sonido musical de ese genero!!! #URL# Unete #USER#>*, 'Chimbala announces union between some dembowsers to change the musical sound of this genre!!!'). On the other hand, we found also tweets containing: 1. personal opinions (e.g., *no te vas a poner a dialogar sobre la cosntruccion de un nuevo pais, sobre aristotles, pitagoras o engels.*, 'you are not going to start a dialogue about the construction of a new country, about Aristotle, Pythagoras or Engels.'); 2. political news we do not know if they are fake or not (e.g., *El nuevo Gobierno boliviano detendrá a diputados del partido de Morales por [UNK] y sedición" #URL#*, 'The new Bolivian government will arrest deputies from Morales' party for [UNK] and sedition').

nFNS Patterns. We noticed in nFNS sample windows: 1. complete questions (e.g., *¿Por qué se nos riza el pelo? ¿Por qué crece pero las pestañas y el vello no? #URL# vía #USER#*); 2. series of mentions (from three up; two mentions in a row are also present in FNS sample data) (e.g., *#USER# #USER# #USER# #USER# #USER# #USER# #USER# Quería poner tocaros, no tocarlos...*); 3. politics (e.g., *Se ha visto Srª #USER# en estas imágenes, a mi me da verguenza, una diputada del congreso*, 'It has been seen Miss #USER in these images, it gives me shame, a deputy of the congress'); 4. emojis (almost absent in FNS maximum outputs).

This analysis suggests that the CNN model might consider important the features highlighted in the preliminary analysis of the dataset. However, what emerges is also that this CNN model might be biased towards some topics (e.g., music for FNS and politics for nFNS).

5.3. Global Average Pooling Output

Figure 8 shows the output of the global average pooling layer when the training set is provided as input. On x-axis we represent the 32 units of the layer, on the y-axis the values associated to each unit. For every sample of the set, a line is drawn connecting the 32 output values of each unit of the level. Blue lines represent FNS, while green nFNS. Similarly, Figure 9 shows values of the 32-GAP-output units when test set samples are provided to the CNN. In this case, some of the lines near to 0 values output fall outside

their actual area. This might suggest that wrongly predicted samples are similar to the opposite class, hence confusing our classifier when making predictions.

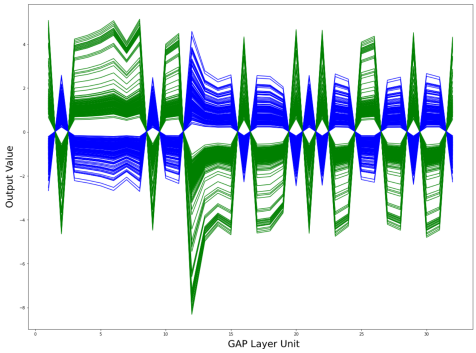


Figure 8. Global average pooling layer output providing the training test as model input. For both classes (i.e., FNS and nFNS) every sample is correctly classified. In this case no overlapped lines are visible between the two groups of lines (i.e., green or blue). Each line corresponds to an author.

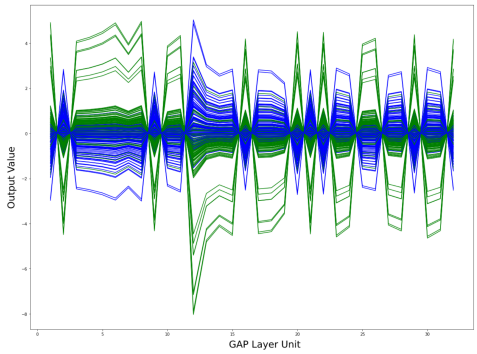


Figure 9. Global average pooling layer output providing the full test set as model input. In this case some errors are visible (i.e., green lines in blue-line zone and vice versa). It is worth noting that errors in detection are often near to 0 values output. This might suggest that the 0.0 threshold value used to separate the classes is small and this could possibly explain model mistakes.

Thus, we extracted 2 documents per class selecting one document whose 32-GAP-output values are far from the 0 threshold and one near it, because we imagined that highly characterised documents (i.e., documents which contain a high number of features characteristic of their class) should be far from 0. As expected, the features highlighted in the preliminary analysis are in a higher number in those documents whose 32-GAP-output values are far from 0. In particular, 52% of tweets in the far-from-0 FNS document start with *VIDEO*, *DE ULTIMO MINUTO* 'breaking news', *ESTRENO*, *IMPACTANTE*, or *DESCARGAR*, 76% contain *Unete* at the end of the tweet (i.e., contain keywords of FNS as reported in Table 2). Similarly, in the far-from-0 nFNS document, 19% of the total number of tokens is made of *#HASHTAG#*, in addition to other keywords reported also in Table 2 such as *Samsung*, *bulos*, *qué*, *informacíon*, but also complete questions (starting with ¿ and ending with ?) as emerged as important feature analysing the first convolutional layer output. In the two documents whose 32-GAP-output values are near to 0, we found a similar tweet (nFNS: *He publicado una foto nueva en Facebook #URL#*, 'I have posted a new photo on Facebook #URL#.', and FNS: *He publicado un vídeo nuevo en Facebook #URL#.*, 'I have posted a new video on Facebook #URL#.') repeated more than once, 33 and 7 times out of 100 in nFNS and FNS, respectively. This, not only, reduces the variety of features available for classifying each document, but also it is a similar behaviour shared by the two opposite-class authors. In addition, in both documents at least a quarter of tweets are retweets (25% and 29% in nFNS and FNS, respectively), though different in nature. In

particular, the analysed nFNS author retweeted mostly users' personal opinion (e.g., about politics), whilst the FNS author retweeted mostly crime news.

5.4. Qualitative Error Analysis

The CNN tested, in the best performing run on the Spanish dataset, reaches an accuracy of 0.82 and fails to recognize 19 FNS and 17 nFNS authors, confirming that FNS are slightly more difficult to identify than nFNS. Since it is worst to mistakenly label a nFNS as a FNS, we decided to analyse the features of wrongly and correctly identified nFNS—following the suggestion by [65] who propose that error analysis should also be done on correctly identified samples to verify why the system performs well, especially when using black-box models.

Since we suppose that the CNN model considers important for the classification the distribution of keywords, we selected three nFNS samples—one wrongly identified as FNS and two correctly identified as nFNS—containing keywords identified as a good predictor of FNS. We found that the CNN model is able to distinguish different usages of the same keyword. In Table 6, we show three different examples in which the lemma *remedio* (cfr. Table 2) is used in two different ways. Examples 1 and 3 are similar to what can be found in FNS tweets. Example 2 is very different from FNS authors' usage. Since the model does not make its decision based only on one tweet (the first convolutional layer takes 32-token windows corresponding to maximum three complete tweets), we can suppose that the presence of the tweets reported in Example 1 and 3 are not enough to assign the FNS label to a nFNS. The author sharing the tweet in Example 1 was wrongly labelled as FNS by the CNN model. The authors sharing the tweets in Examples 2 and 3 were correctly identified as nFNS by the CNN model.

Table 6. Examples drawn from the nFNS Spanish test set.

Example	Tweet Text
1	RT #USER#: Remedio casero para limpiar las juntas del azulejo. #URL#
2	La venta de medicamentos con receta bajó todos los años entre 2016 y 2019. Además, en 2018 la mitad de los hogares pobres de CABA y el Conurbano debieron dejar de comprar remedios por problemas económicos. Más info en esta nota #URL# de #USER#. #URL#
3	Poderoso remedio casero para eliminar el colesterol de los vasos sanguíneos y perder peso -... #URL#

The author that shared the tweet in Example 2, shared also several features in line with what we found in the preliminary analysis for nFNS. This author, indeed, always publishes where to find information concerning what they say and also shares information on how to counteract misinformation, thus we might suppose that the CNN model pays more attention to these features and not on the presence of that precise tweet containing a keyword of FNS. Then, the question is why the authors sharing Example 1 and 3 are not both wrongly—or correctly—predicted. The author who shared the tweet in Example 1, not only uses the keyword *remedio* (used by many FNS), but also contains several variants of one of the highly discriminant keywords pinpointed both by Sketch Engine and by the Bayesian model, i.e., *video*. Conversely, the author sharing the tweet in Example 3, apart from sharing *powerful remedies*, they ask many questions (and we saw in Section 5.2 that the convolutional filters consider questions as good predictors of nFNS) and publishes personal opinions in both explicit and implicit form (e.g., *yo opino*, 'I think'; *yo digo*, 'I say'; *yo comento*, 'I comment').

Hence, we supposed that the CNN model is able to discriminate also on the basis of the presence of overtly expressed personal pronouns. We checked if FNS and nFNS use the first person pronoun *yo* differently. We performed a Welch t test and found a statistically significant p value of 0.0194 when inspecting together test and train, a p value not quite statistically significant looking only at train data (0.0833), and a p value of 0.1158 in test data, which is not statistically significant. Thus, we might suppose that since it was trained only on train data, this difference should not be so discriminant. Then, we wanted to measure if nFNS use more first-person verbs and pronouns than FNS (both singular and plural). To obtain this type of information, we automatically parsed the dataset using the AnCora pretrained model with UDPipe (https://lindat.mff.cuni.cz/services/udpipe/, accessed on 15 August 2022) The linguistic annotation confirmed that nFNS tweets contain more first-person tokens than FNS. Hence, we performed a Welch t. test to determine if this difference is statistically significant. We found a p value less than 0.0001, thus extremely statistically significant. We also investigated if also the second and the third person features were significantly different and found a p value less than 0.0001 for each person (1, 2 and 3, taken singularly) and also when aggregated. This last result suggests that these two classes use differently verbs (and auxiliaries) and pronouns—the only parts of speech that can have this morphological feature (i.e., person).

6. Conclusions

In this article, we presented a comparative evaluation of SotA models, with a special focus on transformers, to address the task of FNS detection. We exploited corpus linguistics techniques to guide the analysis of the multilingual dataset used as case study. As the analysis of the dataset suggested, all the compared models performed better on the Spanish test set. From the comparative evaluation, it emerges that attention-based models are not the optimal solution for the analysed task. In fact, as far as FNS are concerned, deterministic models, such as Naive Bayes and SVM, proved to perform better on the multilingual dataset proposed at PAN@CLEF2020.

On the best performing model—a shallow CNN—we carried out a post-hoc analysis of layers output. We observed similarities between the keywords dataset analysis and the embedding space generated by the CNN. Two clearly-distinguishable clusters representing the two different classes were outlined in the embedding space, and their position seems correlating with their keyness score. The higher the keyness score, the farther the tokens from the other cluster. Mapping the convolutional layer outputs to inputs, we analysed the token windows having maximum and minimum local values. We observed that the CNN filters assigned maximum values to different topics depending on the user class. It is not clear if it is a topic bias in the dataset (music for FNS and politics for nFNS) or if it applies also in real case scenarios.

In addition, a comparison between shallow and pre-trained models should be investigated. Probably the shallow CNN performs better because it can deal with variety in users feed. This tendency is noted also in a similar sentiment analysis task—i.e., profiling Hate Speech Spreaders on Twitter [55]—in which the best performing model is a CNN [54] which outperformed the participating Transformer-based models.

Author Contributions: Conceptualization, M.S., E.D.N., I.T. and M.L.C.; methodology, M.S.; software, M.S.; validation, M.S.; formal analysis, M.S. and E.D.N.; investigation, M.S.; data curation, M.S.; writing—original draft preparation, M.S. and E.D.N.; writing—review and editing, M.S., E.D.N., I.T. and M.L.C.; visualization, M.S. and E.D.N.; supervision, M.S., I.T. and M.L.C.; project administration, M.S., I.T. and M.L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the code developed to run our experiments is publicly available on GitHub at https://github.com/marco-siino/fake_news_spreaders_detection (Accessed on 15 August 2022).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FN	Fake News
FNS	Fake News Spreaders
nFNS	non-Fake News Spreaders
PFNSoT	Profiling Fake News Spreaders on Twitter

References

1. Vosoughi, S.; Roy, D.; Aral, S. The spread of true and false news online. *Science* **2018**, *359*, 1146–1151. [CrossRef] [PubMed]
2. Pian, W.; Chi, J.; Ma, F. The causes, impacts and countermeasures of COVID-19 "Infodemic": A systematic review using narrative synthesis. *Inf. Process. Manag.* **2021**, *58*, 102713. [CrossRef]
3. McGonagle, T. 'Fake news' False fears or real concerns? *Neth. Q. Hum. Rights* **2017**, *35*, 203–209. [CrossRef]
4. Zhou, X.; Zafarani, R. A survey of fake news: Fundamental theories, detection methods, and opportunities. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–40. [CrossRef]
5. Zhang, X.; Ghorbani, A.A. An overview of online fake news: Characterization, detection, and discussion. *Inf. Process. Manag.* **2020**, *57*, 102025. [CrossRef]
6. Lazer, D.M.; Baum, M.A.; Benkler, Y.; Berinsky, A.J.; Greenhill, K.M.; Menczer, F.; Metzger, M.J.; Nyhan, B.; Pennycook, G.; Rothschild, D.; et al. The science of fake news. *Science* **2018**, *359*, 1094–1096. [CrossRef] [PubMed]
7. Guo, Z.; Schlichtkrull, M.; Vlachos, A. A Survey on Automated Fact-Checking. *Trans. Assoc. Comput. Linguist.* **2022**, *10*, 178–206. [CrossRef]
8. Shu, K.; Sliva, A.; Wang, S.; Tang, J.; Liu, H. Fake news detection on social media: A data mining perspective. *Acm Sigkdd Explor. Newsl.* **2017**, *19*, 22–36. [CrossRef]
9. Rubin, V.L. On deception and deception detection: Content analysis of computer-mediated stated beliefs. *Proc. Am. Soc. Inf. Sci. Technol.* **2010**, *47*, 1–10. [CrossRef]
10. Moravec, P.L.; Minas, R.K.; Dennis, A. Fake News on Social Media: People Believe What They Want to Believe When it Makes No Sense At All. *Manag. Inf. Syst. Q.* **2019**, *43*, 1343–1360. [CrossRef]
11. Sharma, K.; Qian, F.; Jiang, H.; Ruchansky, N.; Zhang, M.; Liu, Y. Combating fake news: A survey on identification and mitigation techniques. *Acm Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–42. [CrossRef]
12. Southwell, B.G.; Thorson, E.A.; Sheble, L. The Persistence and Peril of Misinformation: Defining what truth means and deciphering how human brains verify information are some of the challenges to battling widespread falsehoods. *Am. Sci.* **2017**, *105*, 372–376.
13. Molina, M.D.; Sundar, S.S.; Le, T.; Lee, D. "Fake news" is not simply false information: A concept explication and taxonomy of online content. *Am. Behav. Sci.* **2021**, *65*, 180–212. [CrossRef]
14. Cambria, E.; Li, Y.; Xing, F.Z.; Poria, S.; Kwok, K. SenticNet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual, 19–23 October 2020; pp. 105–114.
15. Cambria, E.; Kumar, A.; Al-Ayyoub, M.; Howard, N. Guest Editorial: Explainable Artificial Intelligence for Sentiment Analysis. *Know.-Based Syst.* **2021**, *238*, 1–3. [CrossRef]
16. Rangel, F.; Giachanou, A.; Ghanem, B.; Rosso, P. Overview of the 8th Author Profiling Task at PAN 2020: Profiling Fake News Spreaders on Twitter. In Proceedings of the CLEF 2020–Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, 22–25 September 2020.
17. Derczynski, L.; Bontcheva, K.; Liakata, M.; Procter, R.; Hoi, G.W.S.; Zubiaga, A. SemEval-2017 Task 8: RumourEval: Determining rumour veracity and support for rumours. *arXiv* **2017**, arXiv:1704.05972.
18. Gorrell, G.; Kochkina, E.; Liakata, M.; Aker, A.; Zubiaga, A.; Bontcheva, K.; Derczynski, L. SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours. In Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, MN, USA, 6–7 June 2019; pp. 845–854.
19. Chakraborty, A.; Paranjape, B.; Kakarla, S.; Ganguly, N. Stop clickbait: Detecting and preventing clickbaits in online news media. In Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (Asonam), San Francisco, CA, USA, 18–21 August 2016; pp. 9–16.
20. Ghanem, B.; Rosso, P.; Rangel, F. An emotional analysis of false information in social media and news articles. *ACM Trans. Internet Technol. (TOIT)* **2020**, *20*, 1–18. [CrossRef]
21. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The rise of social bots. *Commun. ACM* **2016**, *59*, 96–104. [CrossRef]
22. Rangel, F.; Rosso, P. Overview of the 7th author profiling task at PAN 2019: Bots and gender profiling in Twitter. In Proceedings of the CLEF 2019–Conference and Labs of the Evaluation Forum, Lugano, Switzerland, 9–12 September 2019.

23. Lomonaco, F.; Donabauer, G.; Siino, M. COURAGE at CheckThat! 2022: Harmful Tweet Detection using Graph Neural Networks and ELECTRA. *CEUR Workshop Proc.* **2022**, *3180*, 573–583.
24. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv* **2017**, arXiv:1707.01926.
25. Pradhyumna, P.; Shreya, G.P.; Mohana. Graph Neural Network (GNN) in Image and Video Understanding Using Deep Learning for Computer Vision Applications. In Proceedings of the 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 4–6 August 2021; pp. 1183–1189.
26. Siino, M.; La Cascia, M.; Tinnirello, I. WhoSNext: Recommending Twitter Users to Follow Using a Spreading Activation Network Based Approach. In Proceedings of the 2020 International Conference on Data Mining Workshops (ICDMW), Sorrento, Italy, 17–20 November 2020; pp. 62–70.
27. Figuerêdo, J.S.L.; Maia, A.L.L.; Calumby, R.T. Early depression detection in social media based on deep learning and underlying emotions. *Online Soc. Netw. Media* **2022**, *31*, 100225. [CrossRef]
28. Siino, M.; Tinnirello, I.; La Cascia, M. T100: A modern classic ensemble to profile irony and stereotype spreaders. *CEUR Workshop Proc.* **2022**, *3180*, 2666–2674.
29. Croce, D.; Garlisi, D.; Siino, M. An SVM Ensamble Approach to Detect Irony and Stereotype Spreaders on Twitter. *CEUR Workshop Proc.* **2022**, *3180*, 2426–2432.
30. Mangione, S.; Siino, M.; Garbo, G. Improving Irony and Stereotype Spreaders Detection using Data Augmentation and Convolutional Neural Network. *CEUR Workshop Proc.* **2022**, *3180*, 2585–2593.
31. Patwa, P.; Bhardwaj, M.; Guptha, V.; Kumari, G.; Sharma, S.; Pykl, S.; Das, A.; Ekbal, A.; Akhtar, M.S.; Chakraborty, T. Overview of constraint 2021 shared tasks: Detecting english covid-19 fake news and hindi hostile posts. In *International Workshop on Combating On line Hostile Posts in Regional Languages during Emergency Situation*; Springer: Cham, Switzerland, 2021; pp. 42–53.
32. Al-Yahya, M.; Al-Khalifa, H.; Al-Baity, H.; AlSaeed, D.; Essam, A. Arabic Fake News Detection: Comparative Study of Neural Networks and Transformer-Based Approaches. *Complexity* **2021**, *2021*, 5516945. [CrossRef]
33. Mahir, E.M.; Akhter, S.; Huq, M.R.; Abdullah-All-Tanvir. Detecting fake news using machine learning and deep learning algorithms. In Proceedings of the 2019 7th International Conference on Smart Computing & Communications (ICSCC), Sarawak, Malaysia, 28–30 June 2019; pp. 1–5.
34. Bali, A.P.S.; Fernandes, M.; Choubey, S.; Goel, M. Comparative performance of machine learning algorithms for fake news detection. In *International Conference on Advances in Computing and Data Sciences*; Springer: Singapore, 2019; pp. 420–430.
35. Liu, Y.; Wu, Y.F. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In Proceedings of the AAAI conference on artificial intelligence, New Orleans, LA, USA, 4–6 February 2018; Volume 32.
36. Pérez-Rosas, V.; Kleinberg, B.; Lefevre, A.; Mihalcea, R. Automatic Detection of Fake News. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 20–26 August 2018; pp. 3391–3401.
37. Pizarro, J. Using N-grams to detect Fake News Spreaders on Twitter. In Proceedings of the CLEF 2020–Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, 22–25 September 2020.
38. Buda, J.; Bolonyai, F. An Ensemble Model Using N-grams and Statistical Features to Identify Fake News Spreaders on Twitter. In Proceedings of the CLEF 2020–Conference and Labs of the Evaluation Forum, Thessaloniki, Greece, 22–25 September 2020.
39. Leonardi, S.; Rizzo, G.; Morisio, M. Automated classification of fake news spreaders to break the misinformation chain. *Information* **2021**, *12*, 248. [CrossRef]
40. Cui, L.; Lee, D. Coaid: Covid-19 healthcare misinformation dataset. *arXiv* **2020**, arXiv:2006.00885.
41. Giachanou, A.; Ghanem, B.; Ríssola, E.A.; Rosso, P.; Crestani, F.; Oberski, D. The impact of psycholinguistic patterns in discriminating between fake news spreaders and fact checkers. *Data Knowl. Eng.* **2022**, *138*, 101960. [CrossRef]
42. Cervero, R.; Rosso, P.; Pasi, G. Profiling Fake News Spreaders: Personality and Visual Information Matter. In *International Conference on Applications of Natural Language to Information Systems*; Springer: Cham, Switzerland, 2021; pp. 355–363.
43. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
44. Cañete, J.; Chaperon, G.; Fuentes, R.; Ho, J.H.; Kang, H.; Pérez, J. Spanish Pre-Trained BERT Model and Evaluation Data. *PML4DC ICLR* **2020**, *2020*, 1–10.
45. Siino, M.; La Cascia, M.; Tinnirello, I. McRock at SemEval-2022 Task 4: Patronizing and Condescending Language Detection using Multi-Channel CNN, Hybrid LSTM, DistilBERT and XLNet. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Seattle, WC, USA, 14–15 July 2022; pp. 409–417. [CrossRef]
46. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
47. Abdaoui, A.; Pradel, C.; Sigel, G. Load What You Need: Smaller Versions of Mutililingual BERT. In Proceedings of the SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, Virtual, 20 November 2020; pp. 119–123.
48. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.

49. Gutiérrez-Fandiño, A.; Armengol-Estapé, J.; Pàmies, M.; Llop-Palao, J.; Silveira-Ocampo, J.; Carrino, C.P.; Gonzalez-Agirre, A.; Armentano-Oller, C.; Rodriguez-Penagos, C.; Villegas, M. Spanish Language Models. *arXiv* **2021**, arXiv:2107.07253.
50. Clark, K.; Luong, M.T.; Le, Q.V.; Manning, C.D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv* **2020**, arXiv:2003.10555.
51. Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The long-document transformer. *arXiv* **2020**, arXiv:2004.05150.
52. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv* **2019**, arXiv:1906.08237.
53. Chen, G.; Ma, S.; Chen, Y.; Dong, L.; Zhang, D.; Pan, J.; Wang, W.; Wei, F. Zero-shot Cross-lingual Transfer of Neural Machine Translation with Multilingual Pretrained Encoders. *arXiv* **2021**, arXiv:2104.08757.
54. Siino, M.; Di Nuovo, E.; Tinnirello, I.; La Cascia, M. Detection of Hate Speech Spreaders using Convolutional Neural Networks. *CEUR Workshop Proc.* **2021**, *2936*, 2126–2136.
55. Rangel, F.; Sarracén, G.; Chulvi, B.; Fersini, E.; Rosso, P. Profiling hate speech spreaders on twitter task at PAN 2021. In Proceedings of the CLEF 2021–Conference and Labs of the Evaluation Forum, Bucharest, Romania, 21–24 September 2021.
56. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *Acm Trans. Intell. Syst. Technol. (TIST)* **2011**, *2*, 1–27. [CrossRef]
57. McCallum, A.; Nigam, K. A comparison of event models for naive bayes text classification. In Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, Madison, WI, USA, 26–27 July 1998.
58. Kilgarriff, A.; Baisa, V.; Bušta, J.; Jakubíček, M.; Kovář, V.; Michelfeit, J.; Rychlý, P.; Suchomel, V. The Sketch Engine: Ten years on. *Lexicography* **2014**, *1*, 7–36. [CrossRef]
59. Kilgarriff, A. Comparing corpora. *Int. J. Corpus Linguist.* **2001**, *6*, 97–133. [CrossRef]
60. Demmen, J.E.; Culpeper, J.V. Keywords. In *The Cambridge Handbook of English Corpus Linguistics*; Biber, D., Reppen, R., Eds.; Cambridge University Press: Cambridge, UK, 2015; pp. 90–105.
61. Kilgarriff, A. Getting to know your corpus. In *International Conference on Text, Speech and Dialogue*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 3–15.
62. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. Transformers: State-of-the-Art Natural Language Processing. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Virtual, 16–20 November 2020; pp. 38–45.
63. Cambria, E.; Schuller, B.; Xia, Y.; Havasi, C. New avenues in opinion mining and sentiment analysis. *IEEE Intell. Syst.* **2013**, *28*, 15–21. [CrossRef]
64. Kenny, E.M.; Ford, C.; Quinn, M.; Keane, M.T. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies. *Artif. Intell.* **2021**, *294*, 103459. [CrossRef]
65. Bender, E.M.; Koller, A. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 June 2020; pp. 5185–5198. [CrossRef]