Hate speech detection in Italian tweets

Group 4

1 INTRODUCION

1.1 Motivation

Although hate speech is a phenomenon that existed long before the advent of the Web, communication via the Internet has allowed this phenomenon to spread and develop with peculiar characteristics tied to the nature of online media, such as anonymity on one side, and velocity and breadth of proliferation on the other. Given the relevance and dangerous potential of this social phenomenon, there is widespread interest in recognizing and detecting hate speech online, also on an institutional level.

In May 2016 the European Commission, citing the chilling effect of hate speech on the democratic discourse on online platforms, agreed with Facebook, Microsoft, Twitter and YouTube on a "Code of conduct on countering illegal hate speech online" [2]. In this document the major IT Companies made a public commitment to:

1. Have in place clear and effective processes to review notifications regarding hate speech on their platforms;

2. review the majority of valid notifications of hate speech in less than 24 hours.

The most recent evaluation campaign of the Code of Conduct [5] assessed in general a worse performance of the IT Companies especially in this last respect (only 64.4% of notifications were reviewed in less than 24 hours), confirming a troubling downward trend already observed in 2021.

Accurate, precise and efficient automated systems of hate speech detection are necessary for combating the phenomenon both with proactive action and in the process of reviewing user notification of occurrences of such violations.

Moreover, such systems should prove valuable for providing the data that is still needed to investigate the ecosystem of hate speech online and the magnitude of the phenomenon.

1.2 Project goal

The goal of this project is to identify characteristics in the text which allow for the detection of Italian tweets containing hate speech (this category is meant to include expressions of racism, xenophobia and terrorist propaganda). So, we can define this task as a binary classification.

A secondary goal is to evaluate how insights obtained by our model can generalize across textual domains. In order to do that, we would like to test our model, trained exclusively on tweets, against Italian newspaper headlines.

The project is based on the main task of the HaSpeeDe 2 shared task presented at EVALITA2020 [6].

2 METHODS

2.1 Data

We used the datasets made available[1] by the organizers of the Evalita 2020 shared task HaSpeeDe 2 [6].

The train set consists in 6839 Italian Tweets posted between October 2016 and May 2019 annotated for presence of hate speech. The test set includes both a corpus of 1263 Italian Tweets posted between January and May 2019 and a corpus of 500 Italian newspapers' headlines retrieved between October 2017 and February 2018 from the online editions of *La Stampa*, *La Repubblica*, *Il Giornale* and *Liberoquotidiano*. This second corpus will allow us to appraise how well the application generalizes across textual domains.

Table 1 shows the distribution of hate speech labels in the training set and in each of the test sets.

The data sets are available in TSV format and contain three features for each record:

- `id`: numeric identifier for each document
- `text`: the body of the document

1 https://github.com/msang/haspeede/tree/master/2020

Table 1: Distribution of Hate Speech labels.

	HS	Not HS	Total
Train	2766	4073	6839
Test Tweets	622	641	1263
Test News	181	319	500

- `hs`: boolean value, whether the document contains HS (1) or not (0).
- `stereotype`: boolean value, whether the document contains a stereotype (1) or not (0) [might not be useful].

As part of the preprocessing performed by the organizers of the task, mentions and URLs recurring in the original documents were replaced with `@user` and `URL` placeholders.

We will also make use of the Italian Twitter embeddings [1] lexicon computed by ItaliaNLP Lab on a corpus of 50,000,000 tweets using the word2vec[2] toolkit. This lexicon consists of embeddings of 128 features for 1,188,949 tokens that were computed with the CBOW model with a symmetric context window of 5 tokens.

The embeddings were made available[3] by the ItaliaNLP Lab as a SQLite database containing a single table `store` with 130 columns:

- `key`, representing the token;
- one column for each dimension of the embedding;
- `ranking` storing the frequency rank of the token.

2.2 Automatic text annotation

[4]

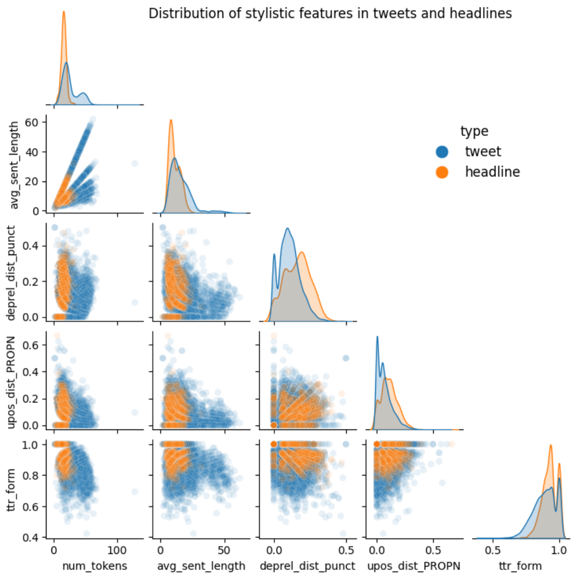
2.3 Linguistic profiling

2.4 Ngrams extraction and modelling

In this section, we describe the process used to extract n-grams from tweets to exploit them as features. The process begins with a pre-processing step applied to the raw training dataset, implemented through the `clean_df` function, which consists of three inner functions.

The first inner function, `count_uppercase`, calculates the number of uppercase characters in the dataset. When we considered the stylistic features, we successfully replaced this simple counter with a function that computes ratio of all uppercase words in a document.

Figure 1: Comparative stylistic analysis of tweets and scraped headlines.



Additionally, we defined a custom punctuation set that extends the standard `string.punctuation` by including special characters encountered in tweets. After applying this function, the dataset includes the original columns, `text` and `hs`, along with a new column, `uppercase_count`.

The second inner function, `split_hashtags`, utilizes the WordSegment library to split words within hashtags. This function is stochastic, meaning it may split hashtags differently during each execution. Based on our observations, the function performs well in identifying and splitting longer words within hashtags but struggles with stopwords and shorter words. This second inner function is incorporated into the `process_hashtags` function, which tokenizes hashtags when more than one word is present. An additional condition was added to handle words longer than 10 characters, although this could occasionally split valid long words incorrectly. However, given the rarity of such words in the dataset, this approach is effective for identifying incorrectly joined words. At this stage, the dataset includes a new column, `text_processed`.

Subsequently, all text is converted to lowercase, mentions, URLs, and numbers are removed, and the tweets are tokenized using *Stanza* [4]. Tokens are then further processed by removing stopwords and punctuation. After tokenization, two addi-

2 http://code.google.com/p/word2vec/
3 http://www.italianlp.it/download-italian-twitter-embeddings/

tional columns are added: one reporting the final number of tokens per tweet and another counting the occurrence of bad words, if any. The bad words were identified using a list of 500 Italian bad words retrieved from GitHub[4].

Once the tokenized version of the tweets was prepared, we lemmatized the tokens to ensure higher consistency and reduce the number of final n-grams. Given the brevity of tweets, we limited the n-grams to unigrams and bigrams.

The final dataset consisted of 6837 rows and 6366 features, including the original tweet, tokenized version, lemmatized version, the number of uppercase characters, the count of bad words, and extracted unigrams and bigrams. Before extracting additional features and embeddings, we conducted initial experiments by training models directly on this dataset. Several models were tested, including Random Forest, SVM, XGBoost, and Logistic Regression. After applying random search for hyperparameter optimization, the best performance was achieved with XGBoost, yielding a macro average F1 score of 0.759.

The same preprocessing steps were applied to both test sets: one containing tweets and the other consisting of 500 headlines from Italian newspapers. To ensure consistency between the test datasets and the training set, we used the dictionary, the Bigram model, and the TF-IDF model trained on the training set. As a result, the final tweets test dataset has the same number of features as the training set, minus one (the target class). This ensures that the model trained on the training dataset can be correctly applied to the test set. Specifically, if an n-gram is present in both the training and test sets, the corresponding column will report non-zero values; otherwise, the column will consist entirely of zeros. Additionally, by using the same TF-IDF model for both training and testing, the weight of each n-gram remains consistent. The final dataset consisted of 6837 rows and 6366 features, including the original tweet, tokenized version, lemmatized version, the number of uppercase characters, the count of bad words, and extracted unigrams and bigrams. Before extracting additional features and embeddings, we conducted initial experiments by training models directly on this dataset. Several models were tested, including Random Forest, SVM, XGBoost, and Logistic Regression. After applying random search for hyperparameter optimization, the best performance was achieved with XGBoost, yielding a macro average F1 score of 0.759.

4 https://github.com/napolux/paroleitaliane/tree/master/paroleitaliane

2.5 Embeddings

[3] [7] [1]

3 RESULTS

4 DISCUSSION

5 CONCLUSION

REFERENCES

[1] Andrea Cimino, Lorenzo De Mattei, and Felice Dell'Orletta. "Multi-task Learning in Deep Neural Networks at EVALITA 2018". In: *EVALITA Evaluation of NLP and Speech Tools for Italian*. Ed. by Tommaso Caselli et al. Torino: Accademia University Press, 2018, pp. 86–95. ISBN: 978-88-319-7842-2 978-88-319-7869-9. DOI: 10.4000/books.aaccademia.4527. URL: https://books.openedition.org/aaccademia/4527 (visited on 06/04/2024).

[2] European Commission. *The EU Code of conduct on countering illegal hate speech online*. URL: https://commission.europa.eu/strategy-and-policy/policies/justice-and-fundamental-rights/combatting-discrimination/racism-and-xenophobia/eu-code-conduct-countering-illegal-hate-speech-online_en (visited on 10/11/2024).

[3] Marco Polignano et al. "AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets". In: *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*. Vol. 2481. CEUR, 2019. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074851349&partnerID=40&md5=7abed946e06f76b3825ae5e294ffac14.

[4] Peng Qi et al. "Stanza: A Python Natural Language Processing Toolkit for Many Human Languages". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 2020. URL: https://nlp.stanford.edu/pubs/qi2020stanza.pdf.

[5] Didier Reynders. *7th evaluation of the Code of Conduct*. Fact-sheet. European Commission, Nov. 2022. URL: https://commission.europa.eu/document/download/5dcc2a40-785d-43f0-b806-f065386395de_en?filename=Factsheet%20-%207th%20monitoring%20round%20of%20the%20Code%20of%20Conduct.pdf.

[6] Manuela Sanguinetti et al. "HaSpeeDe 2@ EVALITA2020: Overview of the EVALITA 2020 Hate Speech Detection Task". In: *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*. Ed. by Valerio Basile et al. Online: CEUR.org, 2020.

[7] Stefan Schweter. *Italian BERT and ELECTRA models*. Version 1.0.1. Nov. 2020. DOI: 10.5281/zenodo.4263142. URL: https://doi.org/10.5281/zenodo.4263142.