HaSpeeDe 2: Comparative Analysis of Feature Extraction Models and Machine Learning Classifiers for Hate Speech Detection

Mariangela Panunzio
University of Bari "Aldo Moro"
m.panunzio13@studenti.uniba.it

Abstract

The rise of the internet and social media platforms has enabled the rapid spread of online hate speech, leading to significant consequences, including discrimination, hostility, and violent incitation. Machine learning techniques and natural language processing have a crucial role in addressing this issue. EVALITA 2020 Hate Speech Detection (HaSpeeDe2) challenge's Task A is a binary classification task aimed at determining the presence or the absence of hateful content in the text. Hence, this case study aims to address this task by combining several feature extraction methods with machine learning models to evaluate their performances in detecting online hate speech.

1 Introduction and Motivations

Free speech is essential and reflects a healthy society, facilitating differences of opinion while respecting tolerance of diversity and creativity. However, the relationship between extremism, tolerance, and free speech is complex. Hate speech challenging contemporary society is mainly fed by the power, speed, and reach of the Internet. (Guiora and Park, 2017) Specifically, the internet offers freedom of communication and opinion expression, but the current social media is regularly being misused to spread violent messages and hateful speech. (Zhang and Luo, 2019)

Online hate speech ca be defined as any communication that disparages a person or a group on the basis of characteristics such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or political affiliation. Its main consequences involve harm against social groups by creating an environment of prejudice and intolerance, fostering discrimination and hostility, and in severe cases facilitating violent acts. As a matter of fact, it has been linked to a global increase in violence toward immigrants and other minorities, including mass shootings, lynchings, and ethnic cleansing. (Gagliardone et al., 2015; Laub, 2019) However, hate speech detection is not an easy task. First, this task is characterized by a high degree of subjectiveness: what is considered acceptable for some might not be for others. Moreover, since there is a fine line between freedom of expression and censorship and illegal discrimination, one of the great challenges is to determine the degree at which online speech can be tolerated before being considered hateful and dangerous for the society. (Guiora and Park, 2017)

Nowadays, social media are the main carrier of hate speech. Thus, some measures have been taken in order to reduce this phenomenon; one of these initiatives is the code of conduct [1] signed by some companies (YouTube, Facebook, Twitter), aimed at monitor and remove within 24 hours this type of publication. Still, the automated detection of such content is a crucial problem to solve in order to address and limit this phenomenon, introducing more safety and security in social media.

Machine learning techniques, along with natural language processing, are the mail tool being used to automate this activity and identify this type of speech more accurately. As a matter of fact, in the last few years, there has been a proliferation of contributions on this topic (Caselli et al., 2020; Jurgens et al., 2019; Fortuna et al., 2019), corpora and lexica (De Pelle and Moreira, 2017; Sanguinetti et al., 2018; Bassignana et al., 2018), dedicated workshops, and shared tasks within national (GermEval 2, HASOC 3, IberLEF 4) and international (SemEval 5) evaluation campaigns, aimed at developing different ways to address the problem. Specifically, the evaluation campaign of

[1] 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

Natural Language Processing and Speech Tools for Italian (EVALITA) started in 2007 and aims to promote the development of language resources for Italian language. In its 2018 edition, a task (HaSpeeDe) was proposed to identify hate speech on Facebook and Twitter (Bosco et al., 2018). Its promising results stimulated the development of the second edition of the event (HaSpeeDe2) at EVALITA 2020 (Manuela et al., 2020; Basile et al., 2020); HaSpeeDe2 aimed to detect whether the Italian language on Twitter contains hate language in order to reduce the spread of hate speeches and online harassment. (Waseem and Hovy, 2016)

The goal of this case study is to compare the performance of different feature extraction methods, including Word2Vec embeddings, TF-IDF vectors, CountVectorizer vectors, coupled with traditional machine learning classifiers, such as Support Vector Machines (SVM), Random Forest (RF), Gradient Boosting (GB), as well as the BiLSTM model.

2 Related Work

One of the traditional ways to perform text classification is based on bag-of-words representation counting the number of occurrences of each word within text. It is often combined with term frequency-inverse document frequency (Spärck Jones, 2004) (TF-IDF) representation: TF-IDF allows the frequencies to be normalized according to how often the words appear in all documents. Specifically, with the rise of neural networks, word vectors have provided useful features for text classification tasks. Over time, several methods and approaches have been reported in the literature for hate speech detection, mainly proposed for English; later, systems have been developed to deal also with other languages, including Turkish, Arabic, Danish (Zampieri et al., 2020), German (Wiegand et al., 2018) and Spanish (Basile et al., 2019). Concerning Italian, the first Hate Speech Detection task (HaSpeeDe) for Italian was organized at EVALITA-2018 (Bosco et al., 2018), which consisted in automatically annotating messages from Twitter and Facebook, with values indicating the presence (or absence) of hate speech. The participating systems adopt a wide range of approaches, including bi-LSTM (De la Pena Sarracén et al., 2018), SVM (Santucci et al., 2018), ensemble classifiers (Polignano and Basile, 2018; Bai et al., 2018), RNN (Nunes et al., 2018), CNN and GRU (von Grünigen et al., 2018). The authors of the best-performing system, ItaliaNLP (Cimino et al., 2018), experiment with three different classification models: one based on linear SVM, another one based on a 1-layer BiLSTM and one based on a 2-layer BiLSTM, exploiting additional data from the 2016 SENTIPOLC task (Barbieri et al., 2016). The same training and test set released for HaSpeeDe has been used also for other types of evaluation, for example to compare classifier performance and settings across different languages (Corazza et al., 2020), confirming the importance of domain-specific language models and the effectiveness of deep learning approaches. In (Polignano et al., 2019b), the AlBERTo[2] monolingual Italian BERT-based language model was trained that outperformed the state-of-the-art on the HaSpeeDe 2018 evaluation task (Polignano et al., 2019a). Transformers were a popular choice in the second edition of this event (HaSpeeDe2) at EVALITA 2020: the participating models are characterized by different architectures that mainly exploit BERT-based models and linguistic features. (Lees et al., 2020; Klaus et al., 2020; Leonardelli et al., 2020; Lavergne et al., 2020) finetuned BERT, AlBERTo and UmBERTo [3] language models for both runs. (Ou and Li, 2020) exploited the pre-trained XLMRoBERTa [4] multi-language model as input of Neural Networks architecture. (Fontana and Attardi, 2020) developed a model that is an ensemble of fixed number of instances of two principal transformers (AlBERTo and DBMDZ [5]) and a combination of DBMDZ input and a dense layer. The DBMDZ is used also by (Deng et al., 2020) in a transfer learning approach. (Cisnero and Ortega Bueno, 2020), on the other hand, used a BiLSTM with the addition of linguistic features in the first run, while using the pre-trained DBMDZ model in the second one. (Gambino et al., 2020) experimented transformer encoders in the first run and depth-wise Separable Convolution techniques in the second one. Moreover, (da Silva and Roman, 2020; Ferraccioli and et al., 2020; Kruschwitz and Hoffmann, 2020; Bisconti and Montagnani, 2020) explored classical machine learning approaches. Finally, (Delmonte, 2020), based on the parser for

[2] https://github.com/marcopoli/AlBERTo-it

[3] https://github.com/musixmatchresearch/umberto

[4] https://huggingface.co/docs/transformers/model_doc/xlm-roberta

[5] https://huggingface.co/dbmdz

Italian ItGetaruns, applied six different rule-based classifiers.

3 Task

3.1 Task Description

HaSpeeDe 2 focused on three main phenomena relevant to online hate speech detection, proposing three different tasks. These tasks are shortly described as follows:

- **Task A - Hate Speech Detection (Main Task)**: a binary classification task aimed at determining the presence or the absence of hateful content in the text towards a given target (among Immigrants, Muslims or Roma people)
- **Task B - Stereotype Detection (Pilot Task 1)**: binary classification task aimed at determining the presence or the absence of a stereotype towards the same targets as Task A
- **- Task C - Identification of Nominal Utterances (Pilot Task 2)**: sequence labeling task aimed at recognizing NUs in data previously labeled as hateful

This case study takes into account only Task A (Main Task).

3.2 Data Description

The dataset being used was made available by the organizers of HaSpeeDe2. Specifically, data is provided in tab-separated values (TSV) in which mentions and URLSs were replaced with “@user” and “URL” placeholders.

The training set contains the Twitter portion of the data of HaSpeeDe 2018, namely tweets posted from October 2016 to April 2017, and a subset of the tweets posted between September 2018 and May 2019, gathered for an Italian hate speech monitoring project. In Figure 1 are shown the words in the Training set used to express a negative connotation.

As for the test set, it includes both in-domain and out-of-domain data, as well as from different time periods. Specifically, news headlines were introduced as cross-domain test data, while the Twitter test set intentionally contains tweets published in a different time frame than those in the training set to verify the systems’ ability to detect HS forms independently of biases. Hence, the test set can be distinguished into the Twitter dataset and News dataset. The Twitter dataset contains tweets posted between January and May 2019. The News dataset includes a corpus composed of newspaper headlines about immigrants retrieved between October 2017 and February 2018 from online newspapers and annotated within the context of a Master’s degree thesis discussed in 2018 at the Department of Foreign Languages at the University of Turin.

Dataset	**HS**	**NOT HS**	**Total**
Training Set	2766	4073	6839
Testing Set - Tweet	622	641	1263
Testing Set - News	181	319	500

Table 1: Distribution of Hate Speech labels among the available datasets.

Table 1 shows the distribution of Hate Speech labels among the datasets.

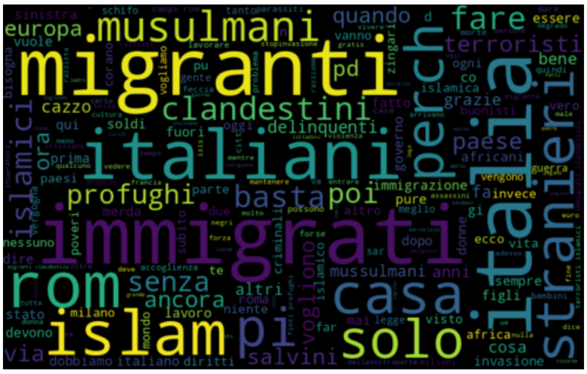


Figure 1: Negative words in the Training Set.

4 Experimental Setting

4.1 Data Pre-Processing

Data pre-processing is an essential step in natural language processing (NLP), particularly when dealing with data from social media platforms. The goal of data pre-processing is to transform raw text data into a suitable format for training a text classification model. In this specific case, a text pre-processing pipeline was implemented for both Twitter data and news headlines. The data pre-processing pipeline involves two steps, namely data cleansing and data normalization steps.

The data cleansing step involves several operations to clean and standardize the text by removing irrelevant patterns and characters. Specifically, it includes lower-casing the text to ensure uniformity, removing specific patterns (e.g., "@user", “URL”, “retweet”, "VIDEO", hashtags), eliminating repeated characters in words (e.g., "goooood" becomes "good"), replacing symbols and specific abbreviations with the corresponding word (e.g., "€" becomes "euro”, "x" with "per," "nn" with "non," "xk" and "xké" with "perchè"). Moreover, any non-alphanumeric character that does not serve a purpose for text classification is replaced with whitespace.

The data normalization step focuses on reducing words to their common base form. Italian stop words, which are commonly occurring words with little semantic value, are removed from each text. Furthermore, stemming is applied to transform words into their base form using the SnowballStemmer with the Italian language, finally returning the pre-processed text to the further text classification models.

In table 2 are shown examples of both original corpus and pre-processed corpus with respect to each dataset.

4.2 Text Classifiers

In this case study, the binary classification task employs a combination of traditional machine learning classifiers, namely Support Vector Machines (SVM), Random Forest (RF), and GradientBoosting (GB), (Buitinck et al., 2013) as well as a Bidirectional Long Short Term Memory (BiLSTM) model. However, every model is configured for binary classification with two labels: hate speech and non-hate speech.

Support Vector Machine (SVM)[6] is a popular machine learning algorithm used for classification tasks. SVM aims to maximize the margin between the decision boundary and the training data points. Hence, it works by mapping the input data into a higher-dimensional space using a kernel function and then finding the hyperplane that best separates the classes. Random Forest (RF)[7] is a learning method that combines multiple decision trees to make predictions. Each tree in the forest independently classifies the input data, and the final prediction is determined by majority voting. Gradient Boosting (GB)[8] is another learning method

Dataset	Layer	Output Shape
Train	#Salvini: In Italia troppi si sono montati la testa, io ringrazio Dio e voi per questi mesi straordinari. Vi raccontavano che su immigrazione non si poteva fare nulla, è bastato usare buonsenso e coraggio. #iocisono #piazzadelpopolo	salvin ital tropp mont test ringraz dio mes straordinar raccont immigr pot far null bast usar buonsens coragg iocis piazzadelpopol
Tweet Test	SALUTE #italiani A RISCHIO CAUSA #immigrazione AFRICANA !!!!!!!!!!! AUMENTO MALATTIE INFETTIVE E DIFFUSIVE !!!! #StopInvasione URL	sal italian risc caus immigr african aument malatt infett diffus stopinvasion
News Test	Pisa, il poster di Salvini con i migranti fatto dagli studenti, Ceccardi: «Va rimosso»	pis poster salvin migrant fatt student ceccard va rimoss

Table 2: Original and Preprocessed Text Data Samples in Training, Tweet Test, and News Test Datasets.

[6] https://scikit-learn.org/stable/modules/svm.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[8] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

that combines multiple weak prediction models, typically decision trees, in a sequential manner. Specifically, each subsequent model focuses on the previously misclassified data to improve the overall prediction accuracy. A Bidirectional Long Short-Term Memory (BiLSTM) model is implemented in addition to these traditional machine learning classifiers. Specifically, BiLSTM is a variant of the LSTM model, which is a type of recurrent neural network (RNN). RNNs are designed to process sequential data by considering the dependencies between previous and future elements in the sequence. BiLSTM incorporates information from both past and future timesteps, enabling a more comprehensive understanding of the input sequences: it leverages bidirectional processing to capture contextual information in both directions. In this case study, two versions of the Bidirectional Long Short-Term Memory (BiLSTM) model were implemented. In the first version, the first layer of the model is an embedding layer that maps each word index in the input sequence to a dense vector representation of 200 dimensions. The embedding layer is initialized with pre-trained word embeddings, namely the embedding matrix, and it is set to be non-trainable (i.e. the word embeddings are kept fixed during training). The Bidirectional LSTM layer allows the model to capture both past and future contextual information, while the subsequent dense layers with ReLU activation (Nair and Hinton, 2010) introduce non-linearity, letting the model learn complex relationships. Dropout layers are included to prevent overfitting, and the final dense layer with sigmoid activation (Nwankpa and et al., 2018) produces binary classification outputs. The sigmoid activation function ensures that the output values are within the range of [0, 1], representing the probabilities of each class, namley hate speech and non-hate speech. In Table 3 is shown the architecture of this BiLSTM model. In the second version, the embedding layer is replaced by an input layer since the input sequences are obtained from TF-IDF vectors. The remaining layers stay the same, including the Bidirectional LSTM, dense, and dropout ones. Specifically, this version is designed to work with the TF-IDF feature extraction method. In Table 4 is shown the architecture of this BiLSTM model.

Layer	Output Shape	Dropout
Embedding	(None, 100, 200)	
Bidirectional	(None, 128)	
Dense	(None, 128)	0.2
Dense	(None, 64)	0.2
Dense	(None, 16)	0.2
Dense	(None, 2)	

Table 3: Architecture of the first version of the BiLSTM model with the embedding layer.

Layer	Output Shape	Dropout
Bidirectional	(None, 128)	
Dense	(None, 128)	0.2
Dense	(None, 64)	0.2
Dense	(None, 16)	0.2
Dense	(None, 2)	

Table 4: Architecture of the second version of the BiLSTM model with the embedding layer.

4.3 Text Classification Approaches

Word2Vec. The first approach employs Word2Vec embeddings. To obtain Word2Vec embeddings, the training dataset is tokenized using the Keras Tokenizer[9], and the sequences of tokens are padded to ensure a consistent length. The Word2Vec model[10] is trained on the tokenized training data, and the trained Word2Vec model is then used to generate embeddings for each token in the vocabulary. For tokens that exist in the Word2Vec model's vocabulary, their corresponding embeddings are extracted and averaged to obtain a representative embedding for the tweet; if a token is out-of-vocabulary, it is assigned a zero vector. For the traditional machine learning classifiers, such as SVM, RF, and GB, the Word2Vec embeddings are directly used as features. In the case of the BiLSTM model, the Word2Vec embeddings are transformed into the LSTM input format. The tweet text from the training dataset is tokenized, and the embeddings for each token are retrieved from the Word2Vec model; the resulting embeddings are then reshaped into a 2D array of shape (number of samples, number of features) and the target labels are converted to categorical format.

TF-IDF. Firstly, a set of traditional machine learning classifiers, such as SVM, RF, and GB, is used. For each classifier, a TF-IDF vectorizer[11] is

[9] https://keras.io/api/keras_nlp/tokenizers/tokenizer/

[10] https://radimrehurek.com/gensim/models/word2vec.html

[11] https://scikit-learn.org/stable/

initialized with a maximum number of words and n-gram range of 1 to 2. The vectorizer is fitted on the training dataset, which consists of preprocessed tweet text and corresponding hate speech labels. Then, pipelines are created for each model, combining the TF-IDF vectorizer and the classifier. The training data is converted to string format, and the model is trained on this transformed data. Then, a Bidirectional Long Short Term Memory (BiLSTM) model is employed. The feature extraction is performed using the TF-IDF vectorization technique: the tweet text from the training dataset is transformed into TF-IDF matrices, while the target labels are converted into one-hot encoded format. The TF-IDF features are reshaped to match the input shape of the BiLSTM model, which consists of a 3D array with dimensions (number of samples, 1, number of features).

Word embeddings. The text data is tokenized using the Keras Tokenizer, which assigns a unique index to each word in the vocabulary. Then, the sequences of tokens are then generated from the tokenized text data, and they are padded to a fixed length to ensure consistency in the input data; this process allows for capturing the semantic meaning and contextual information within the text data. These processed data are used for training the Bidirectional LSTM model.

CountVectorizer. The pre-processed cleaned data is transformed into N-gram representations using the CountVectorizer[12]. The N-gram range is varied from unigrams (single words) to trigrams (three consecutive words): this allows capturing different levels of context and word combinations within the text data. The process involves looping through different N-gram ranges; specifically, for each N-gram range, the training data is converted to a bag of words representation using the CountVectorizer with the specified N-gram range, and the same CountVectorizer is then applied to transform the test data. Three different tradizional machine learning classifiers are employed (namely SVM, RF, and GB), and each model is trained on the N-gram representations of the training data.

4.4 BiLSTM Hyperparameters

In general, the BiLSTM model is trained with Adam optimizer and different learning rates, namely 0.001, 0.0001, 0.00001, binary cross-entropy as loss function, batch size of 32 words, and 10 epochs. However, whenever Word Embeddings is the feature extraction method the BiLSTM model is trained with a batch size of 64 words.

4.5 Performance Evaluation

The performances of the different were evaluated with Accuracy and macro F1-score. Accuracy is the proportion of correct predictions, both true positives and true negatives, among the total number of cases examined. Specifically, it is a statistical measure of how well a binary classification test correctly identifies or excludes a condition. (International Organization for Standardization (ISO), 1994) The formula for quantifying accuracy is the following one:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Evaluating model performance using precision and recall provides a comprehensive understanding of the model's ability to correctly classify instances and identify relevant patterns. The F1 score, which is the harmonic mean of precision and recall, offers a balanced measure that considers both aspects. In this case study, the macro-averaged F1 score is used to assess the overall performance of the models. Specifically, this scoring method calculates the arithmetic mean of the per-class F1 scores, treating all classes equally regardless of their support values. Hence, by this specific score, the evaluation takes into account the performance across all classes, providing a single value that summarizes the model's effectiveness in handling the binary classification task. (Sasaki, 2007) The formula for quantifying F1 score anche macro F1 score is the following one:

$$(F1 - score)_i = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{2}$$

$$MacroF1 - score = \frac{1}{N} \sum_{i=1}^{N} (F1 - score)_i \tag{3}$$

modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

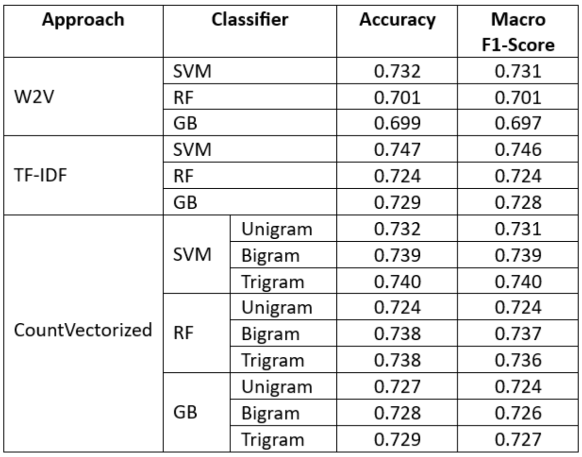
[12] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

5 Experimental Results

5.1 Tweets Test Dataset

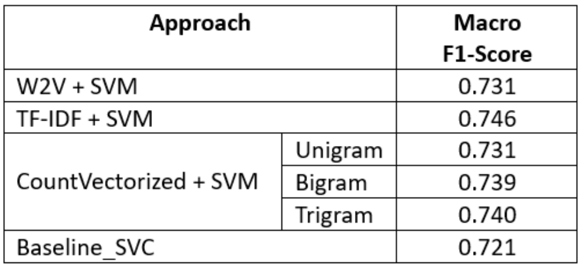
Figure 2 shows the performance of different feature extraction methods and machine learning classifiers for hate speech detection. For the Word2Vec (W2V) approach, the Support Vector Machine (SVM) classifier achieves an accuracy of 0.732 and a macro F1-score of 0.731. The Random Forest (RF) and Gradient Boosting (GB) classifiers achieve slightly lower accuracy and F1-scores. Moving to the TF-IDF approach, the SVM classifier performs slightly better with an accuracy of 0.747 and a macro F1-score of 0.746. The RF and GB classifiers also show good results but slightly lower than SVM. When using the CountVectorizer approach, different n-gram ranges are considered (unigram, bigram, and trigram). The SVM classifier achieves consistent accuracy and F1-scores across the considered n-gram, ranging from 0.732 to 0.740. The RF classifier also performs well, with accuracy and F1-scores ranging from 0.724 to 0.738. Similarly, the GB classifier shows competitive performance across the n-gram ranges, with accuracy and F1-scores ranging from 0.727 to 0.729. Comparing the overall results, the TF-IDF approach with the SVM classifier achieves the highest macro F1-score of 0.746, indicating strong performance in hate speech detection. The CountVectorized approach with the SVM classifier, using unigram, bigram, and trigram n-gram ranges, also demonstrates competitive results with macro F1-scores ranging from 0.731 to 0.740. Moreover,as shown in Figure 3, the Baseline_SVC model achieves a macro F1-score of 0.721, providing a reference point for comparison made available by the task's organizers; thus, every approach with SVM as text classifier overcomes this baseline.

Figure 4 shows the performance of different feature extraction methods and BiLSTM model configurations for hate speech detection. Specifically, each BiLSTM model configuration is characterized by a different learning rate. For the Word2Vec (W2V) approach, the accuracy ranges from 0.701 to 0.712, while the macro F1-score ranges from 0.700 to 0.710. Specifically, with this feature extraction method the highest accuracy and macro F1-score are achieved with a learning rate of 0.0001. Moving to the TF-IDF approach, the accuracy ranges from 0.690 to 0.702, while the macro F1-score ranges from 0.690 to 0.702; in this case, the highest accuracy and macro F1-score are achieved with a learning rate of 0.00001. For the Word Embeddings approach, the accuracy is 0.692, and the macro F1-score is 0.706, both achieved with a learning rate of 0.00001. Overall, the BiLSTM model with Word2Vec approach achieves the highest accuracy and macro F1-score.



Approach	Classifier		Accuracy	Macro F1-Score
W2V	SVM		0.732	0.731
	RF		0.701	0.701
	GB		0.699	0.697
TF-IDF	SVM		0.747	0.746
	RF		0.724	0.724
	GB		0.729	0.728
CountVectorized	SVM	Unigram	0.732	0.731
		Bigram	0.739	0.739
		Trigram	0.740	0.740
	RF	Unigram	0.724	0.724
		Bigram	0.738	0.737
		Trigram	0.738	0.736
	GB	Unigram	0.727	0.724
		Bigram	0.728	0.726
		Trigram	0.729	0.727

Figure 2: Performance Comparison of Feature Extraction Methods and Traditional Machine Learning Classifiers for Hate Speech Detection on the Tweets Test Dataset.

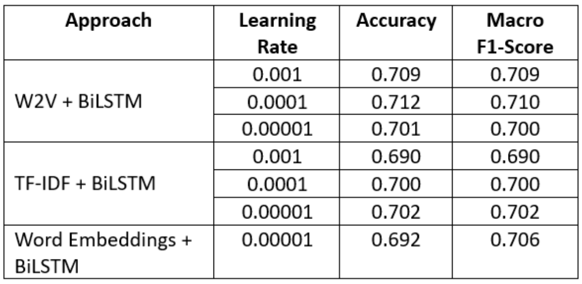


Approach		Macro F1-Score
W2V + SVM		0.731
TF-IDF + SVM		0.746
CountVectorized + SVM	Unigram	0.731
	Bigram	0.739
	Trigram	0.740
Baseline_SVC		0.721

Figure 3: Performance Comparison of different Feature Extraction Methods and SVM classifier with the baseline for this classifier on the Tweets Test Dataset.

5.2 News Test Dataset

Figure 5 shows the performance of different feature extraction methods and machine learning classifiers for hate speech detection. For the Word2Vec (W2V) approach, the SVM classifier achieves an accuracy of 0.730 and a macro F1-score of 0.660. The RF classifier shows an accuracy of 0.714 and a macro F1-score of 0.620, while the GB classifier achieves an accuracy of 0.744 and a macro F1-score of 0.680. As for the TF-IDF approach, the SVM classifier achieves the highest performance with an accuracy of 0.754 and a macro F1-score of 0.689. The RF classifier achieves an accuracy of 0.682 and a macro F1-score of 0.552, while the GB classifier

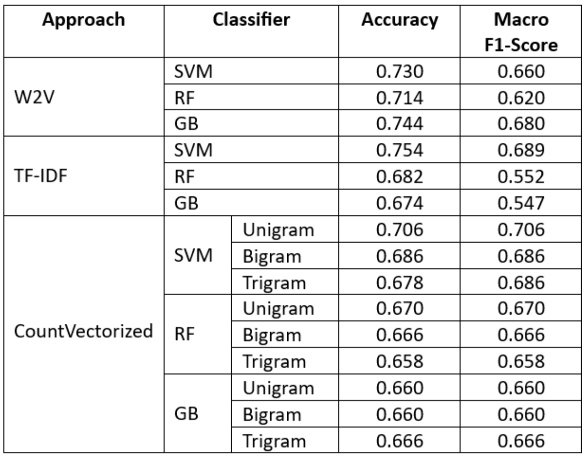


Approach	Learning Rate	Accuracy	Macro F1-Score
W2V + BiLSTM	0.001	0.709	0.709
	0.0001	0.712	0.710
	0.00001	0.701	0.700
TF-IDF + BiLSTM	0.001	0.690	0.690
	0.0001	0.700	0.700
	0.00001	0.702	0.702
Word Embeddings + BiLSTM	0.00001	0.692	0.706

Figure 4: Performance Comparison of Feature Extraction Methods and BiLSTM model configurations for Hate Speech Detection on the Tweets Test Dataset.

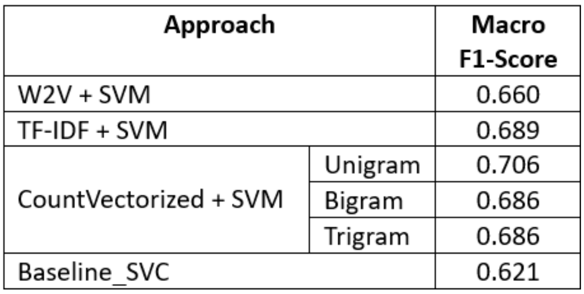
achieves an accuracy of 0.674 and a macro F1-score of 0.547. As for the CountVectorizer approach, considering different n-gram ranges, the SVM classifier consistently achieves accuracy and macro F1-scores ranging from 0.678 to 0.706. The RF classifier also achieves good performances, with accuracy and macro F1-scores ranging from 0.658 to 0.670. Similarly, the GB classifier maintains good performances across the n-gram ranges with accuracy and macro F1-scores of 0.660. Overall, the results indicate that the TF-IDF approach with the SVM classifier obtains the best results, achieving the highest accuracy and macro F1-score. The CountVectorizer approach also shows competitive results, particularly with the SVM classifier and the unigram n-gram range. However, the overall performance, especially in terms of macro F1-score, could be improved across all approaches and classifiers; the leading cause of these low-performance values is the News Testing Set since it is not well balanced in terms of hateful speech and non-hateful speech samples (as shown in Table 1). Moreover, as shown in Figure 6, the Baseline_SVC model achieves a macro F1-score of 0.621, providing a reference point for comparisonmade available by the task's organizers; thus, every approach having SVM as classifier overcomes this baseline.

Table 7 shows the performance of different feature extraction methods and BiLSTM model configurations for hate speech detection. Specifically, each BiLSTM model configuration is characterized by a different learning rate. For the Word2Vec (W2V) approach, the accuracy ranges from 0.714 to 0.748, while the macro F1-score ranges from 0.649 to 0.696; in this case, the highest accuracy and macro F1-score are achieved with a learning rate of 0.001. Moving to the TF-IDF approach, the accuracy ranges from 0.670 to 0.688, while the macro F1-score ranges from 0.608 to 0.631; for this feature extraction approach, the highest accuracy and macro F1-score are achieved with a learning rate of 0.00001. As for the Word Embeddings approach, the accuracy is consistently 0.682 across all learning rates, while the macro F1-score ranges from 0.457 to 0.670; in this case, the highest macro F1-score is achieved with a learning rate of 0.00001. Overall, Word2Vec is the approach that achieves the best performance: the higest accuracy is achieved with 0.001 as learning rate, while the highest macro F1-score is achieved with 0.00001 as learning rate. However, it is important to note that in this case the Word Embeddings approach does not perform as well as the other approaches, with a relatively low macro F1-score across all learning rates.



Approach	Classifier		Accuracy	Macro F1-Score
W2V	SVM		0.730	0.660
	RF		0.714	0.620
	GB		0.744	0.680
TF-IDF	SVM		0.754	0.689
	RF		0.682	0.552
	GB		0.674	0.547
CountVectorized	SVM	Unigram	0.706	0.706
		Bigram	0.686	0.686
		Trigram	0.678	0.686
	RF	Unigram	0.670	0.670
		Bigram	0.666	0.666
		Trigram	0.658	0.658
	GB	Unigram	0.660	0.660
		Bigram	0.660	0.660
		Trigram	0.666	0.666

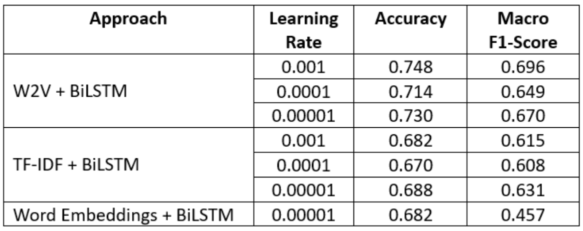
Figure 5: Performance Comparison of Feature Extraction Methods and Traditional Machine Learning Classifiers for Hate Speech Detection on the News Test Dataset.



Approach		Macro F1-Score
W2V + SVM		0.660
TF-IDF + SVM		0.689
CountVectorized + SVM	Unigram	0.706
	Bigram	0.686
	Trigram	0.686
Baseline_SVC		0.621

Figure 6: Performance Comparison of different Feature Extraction Methods and SVM classifier with the baseline for this classifier on the News Test Dataset.

The results shown in Figure 4 and Figure 7 imply that the BiLSTM model combined with different feature extraction approaches achieves different performances. Hence, the choice of learning rate has a notable impact on the accuracy and macro F1-



Approach	Learning Rate	Accuracy	Macro F1-Score
W2V + BiLSTM	0.001	0.748	0.696
	0.0001	0.714	0.649
	0.00001	0.730	0.670
TF-IDF + BiLSTM	0.001	0.682	0.615
	0.0001	0.670	0.608
	0.00001	0.688	0.631
Word Embeddings + BiLSTM	0.00001	0.682	0.457

Figure 7: Performance Comparison of Feature Extraction Methods and BiLSTM model configurations for Hate Speech Detection on the News Test Dataset.

score of the BiLSTM model, focusing on the significance of tuning hyperparameters to optimize the performance of the BiLSTM model for hate speech detection. Further experimentation and analysis may be needed to explore additional hyperparameter settings and enhance the overall performance of the approaches.

6 Discussion

In the tweets test dataset, the TF-IDF approach coupled with the SVM classifier consistently demonstrates the highest accuracy and macro F1-scores, outperforming other feature extraction methods such as Word2Vec and CountVectorizer. Similarly, in the news dataset, the TF-IDF approach with the SVM classifier also performs the best, achieving the highest accuracy and macro F1-score. The TF-IDF approach tends to achieve better results in this hate speech detection task due to its ability to capture the importance of words in a document relative to a corpus of documents. Specifically, TF-IDF considers both the frequency of a term within a document and its occurrence across the entire corpus: by giving higher weight to terms that appear frequently within a document but less frequently in the whole corpus,TF-IDF can effectively highlight words that are significant in the specific context of hate speech detection. This is crucial in distinguishing hate speech from other types of content, contributing to improve classification accuracy.

The BiLSTM model configurations, on the other hand, exhibit varying levels of performance across different feature extraction methods and learning rates. In both datasets, the Word2Vec approach combined with the BiLSTM model achieves the highest accuracy among the three approaches. As a matter of fact, Word2Vec represents words in a dense vector space and considers the neighbouring words surrounding a target word in a given context. This characteristic allows the model to capture the meaning of words within their specific linguistic context, taking into account the influence of nearby words on the target word's semantics. This is particularly useful for hate speech detection, as the context in which certain words appear can greatly impact their association with hate speech. Moreover, Word2Vec maps words to lower-dimensional vectors (in this case, 200 dimensions): this dimensionality reduction facilitates computational efficiency and reduces the risk of overfitting, allowing the model to better generalize from limited training data.

7 Conclusions

The rise of the internet and social media platforms has enabled the rapid spread of extremist and hateful messages, posing challenges to maintaining a healthy society. Online hate speech, which targets individuals or groups based on characteristics such as race, ethnicity, religion, or political affiliation, has significant consequences, including fostering discrimination, hostility, and even inciting violence. Detecting and addressing online hate speech is a challenging task; machine learning techniques and natural language processing has a crucial role in addressing this issue. The goal of this case study is to compare the performance of different feature extraction methods and machine learning classifiers to address Task A of the EVALITA 2020 Hate Speech Detection (HaSpeeDe2) challenge, a binary classification task aimed at determining the presence or the absence of hateful content in the text towards a given target. The TF-IDF approach coupled with the SVM classifier achieves the best performance in hate speech detection for both the tweets and news test datasets. Its ability to capture the importance of words within a document relative to a corpus is crucial to distinguish hate speech from other content, leading to higher accuracy and macro F1-scores. On the other hand, the Word2Vec approach combined with the BiLSTM model achieves the highest accuracy among the other feature extraction methods. Hence, Word2Vec's representation of words in a dense vector space and consideration of contextual information allow it to capture the semantics of hate speech-related terms effectively; moreover, its ability to handle unseen words and dimensionality reduction contribute to improved generalization and mitigated overfitting. Hyperparameter tuning, particularly the learning rate, plays a crucial role in optimizing the performance of the BiLSTM model: finding the optimal learning rate significantly influences the accuracy and macro F1-scores achieved by the BILSTM model. These findings provide insights into the effectiveness of different feature extraction methods and classifiers for hate speech detection, which can be the basis for the development of more robust and accurate models in addressing the spread of hate speech on online platforms.

References

Xiaoyu Bai, Flavio Merenda, Claudia Zaghi, Tommaso Caselli, and Malvina Nissim. 2018. Rug@ evalita 2018: Hate speech detection in italian social media. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:245.

Francesco Barbieri, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, Viviana Patti, et al. 2016. Overview of the evalita 2016 sentiment polarity classification task. In *CEUR Workshop Proceedings*, volume 1749. CEUR-WS.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 54–63.

Valerio Basile, M Di Maro, D Croce, L Passaro, et al. 2020. Evalita 2020: Overview of the 7th evaluation campaign of natural language processing and speech tools for italian. In *CEUR WORKSHOP PROCEEDINGS*, volume 2765. CEUR-ws.

Elisa Bassignana, Valerio Basile, Viviana Patti, et al. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *CEUR Workshop proceedings*, volume 2253, pages 1–6. CEUR-WS.

Elia Bisconti and Matteo Montagnani. 2020. Montanti@ HaSpeeDe2 EVALITA 2020: Hate speech detection in online contents. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, volume 2765, Online event.

Cristina Bosco, Dell'Orletta Felice, Fabio Poletto, Manuela Sanguinetti, Tesconi Maurizio, et al. 2018. Overview of the evalita 2018 hate speech detection task. In *Ceur workshop proceedings*, volume 2263, pages 1–9. CEUR.

Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. 2013. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, Inga Kartoziya, and Michael Granitzer. 2020. I feel offended, don't be abusive! implicit/explicit messages in offensive and abusive language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6193–6202.

Andrea Cimino, Lorenzo De Mattei, and Felice Dell'Orletta. 2018. Multi-task learning in deep neural networks at evalita 2018. *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, pages 86–95.

Mariano Jason Rodriguez Cisnero and Reynier Ortega Bueno. 2020. UO@ HaSpeeDe2: Ensemble Model for Italian Hate Speech Detection. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020*, page 148.

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. 2020. A multilingual evaluation for online hate speech detection. *ACM Transactions on Internet Technology (TOIT)*, 20(2):1–22.

Adriano dos SR da Silva and Norton T. Roman. 2020. No Place For Hate Speech@ HaSpeeDe 2: Ensemble to Identify Hate Speech in Italian. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020*, page 154.

Gretel Liz De la Pena Sarracén, Reynaldo Gil Pons, Carlos Enrique Muniz Cuza, and Paolo Rosso. 2018. Hate speech detection using attention-based lstm. *EVALITA evaluation of NLP and speech tools for Italian*, 12:235.

Rogers Prates De Pelle and Viviane P Moreira. 2017. Offensive comments in the brazilian web: a dataset and baseline results. In *Anais do VI Brazilian Workshop on Social Network Analysis and Mining*. SBC.

Rodolfo Delmonte. 2020. Venses@ HaSpeeDe2 & SardiStance: Multilevel Deep Linguistically Based Supervised Approach to Classification. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020*, page 121.

Tao Deng, Yang Bai, and Hongbing Dai. 2020. By1510@ haspeede 2: Identification of hate speech for italian language in social media data. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020*, page 116.

Federico Ferraccioli and et al. 2020. TextWiller@ SardiStance, HaSpeede2: Text or Con-text? A smart use of social network data in predicting polarization. In *Proceedings of the 7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2020)*. CEUR-WS.org.

Michele Fontana and Giuseppe Attardi. 2020. Fontana-unipi@ haspeede2: Ensemble of transformers for the hate speech task at evalita. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020*, page 136.

Paula Fortuna, Joao Rocha da Silva, Leo Wanner, Sérgio Nunes, et al. 2019. A hierarchically-labeled portuguese hate speech dataset. In *Proceedings of the third workshop on abusive language online*, pages 94–104.

Iginio Gagliardone, Danit Gal, Thiago Alves, and Gabriela Martinez. 2015. *Countering online hate speech*. Unesco Publishing.

Giuseppe Gambino, Roberto Pirrone, and D. Ingegneria. 2020. CHILab@ HaSpeeDe 2: Enhancing hate speech detection with part-of-speech tagging. In *CEUR Workshop Proc.*, volume 2020.

Amos Guiora and Elizabeth A. Park. 2017. Hate speech on social media. *Philosophia*, 45(3):957–971.

International Organization for Standardization (ISO). 1994. Accuracy, I.S.O. "of measurement methods and results—part 1: General principles and definitions". Technical report, International Organization for Standardization, Geneva, Switzerland.

David Jurgens, Eshwar Chandrasekharan, and Libby Hemphill. 2019. A just and comprehensive strategy for using nlp to address online abuse. *arXiv preprint arXiv:1906.01738*.

Svea Klaus, Anna-Sophie Bartle, and Daniela Rossmann. 2020. Svandiela@ HaSpeeDe: Detecting Hate Speech in Italian Twitter Data with BERT. *EVALITA Evaluation of NLP and Speech Tools for Italian-December 17th, 2020*, page 159.

Udo Kruschwitz and Julia Hoffmann. 2020. UR_NLP@ HaSpeeDe 2 at EVALITA 2020: Towards Robust Hate Speech Detection with Contextual Embeddings.

Zachary Laub. 2019. Hate speech on social media: Global comparisons. *Council on foreign relations*, 7.

Eric Lavergne, Rajkumar Saini, György Kovács, and Killian Murphy. 2020. Thenorth@ haspeede 2: Bert-based language model fine-tuning for italian hate speech detection. In *7th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop, EVALITA*, volume 2765.

Alyssa Lees, Jeffrey Sorensen, and Ian Kivlichan. 2020. Jigsaw@ ami and haspeede2: Fine-tuning a pre-trained comment-domain bert model. In *EVALITA*.

Elisa Leonardelli, Stefano Menini, and Sara Tonelli. 2020. Dh-fbk@ haspeede2: Italian hate speech detection via self-training and oversampling. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, volume 2765.

Sanguinetti Manuela, Comandini Gloria, Elisa Di Nuovo, Simona Frenda, Marco Antonio Stranisci, Cristina Bosco, Caselli Tommaso, Viviana Patti, Russo Irene, et al. 2020. Haspeede 2@ evalita2020: Overview of the evalita 2020 hate speech detection task. In *Proceedings of the seventh evaluation campaign of natural language processing and speech tools for Italian. Final Workshop (EVALITA 2020)*, pages 1–9. CEUR.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Sérgio Nunes, P Fortuna, and I Bonavita. 2018. Merging datasets for hate speech classification in italian.

Chigozie Nwankpa and et al. 2018. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

Xiaozhi Ou and Hongling Li. 2020. Ynu_oxz@ haspeede 2 and ami: Xlm-roberta with ordered neurons lstm for classification task at evalita 2020. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 2765:102–109.

Marco Polignano and Pierpaolo Basile. 2018. Hansel: Italian hate speech detection through ensemble learning and deep neural networks. *EVALITA Evaluation of NLP and Speech Tools for Italian*, 12:224.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, Valerio Basile, et al. 2019b. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CEUR Workshop Proceedings*, volume 2481, pages 1–6. CEUR.

Marco Polignano, Pierpaolo Basile, Marco De Gemmis, Giovanni Semeraro, et al. 2019a. Hate speech detection through alberto italian language understanding model. In *NL4AI@ AI* IA*, pages 1–13.

Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An italian twitter corpus of hate speech against immigrants. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*.

Valentino Santucci, Stefania Spina, Alfredo Milani, Giulio Biondi, Gabriele Di Bari, et al. 2018. Detecting hate speech for italian language in social media. In *CEUR WORKSHOP PROCEEDINGS*, volume 2263.

Yutaka Sasaki. 2007. The truth of the F-measure. *Teach tutor mater*, 1(5):1–5.

Karen Spärck Jones. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 60(5):493–502.

Dirk von Grünigen, Fernando Benites, Pius von Däniken, Mark Cieliebak, and Ralf Grubenmann. 2018. spmmmp at germeval 2018 shared task: Classification of offensive content in tweets using convolutional neural networks and gated recurrent units.

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

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the germeval 2018 shared task on the identification of offensive language.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

Ziqi Zhang and Lei Luo. 2019. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semantic Web*, 10(5):925–945.