A Multilingual Evaluation for Online Hate Speech Detection

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, Serena Villata

To cite this version:

Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, Serena Villata. A Multilingual Evaluation for Online Hate Speech Detection. ACM Transactions on Internet Technology, Association for Computing Machinery, 2020, 20 (2), pp.1-22. 10.1145/3377323 . hal-02972184

HAL Id: hal-02972184
https://hal.archives-ouvertes.fr/hal-02972184

Submitted on 20 Oct 2020

A Multilingual Evaluation for Online Hate Speech Detection

Michele Corazza, Università di Bologna, Bologna, Italy
Stefano Menini, Fondazione Bruno Kessler, Trento, Italy
Elena Cabrio, Université Côte d'Azur, Inria, CNRS, I3S, France
Sara Tonelli, Fondazione Bruno Kessler, Trento, Italy
Serena Villata, Université Côte d'Azur, Inria, CNRS, I3S, France

Abstract

The increasing popularity of social media platforms like Twitter and Facebook has led to a rise in the presence of hate and aggressive speech on these platforms. Despite the number of approaches recently proposed in the Natural Language Processing research area for detecting these forms of abusive language, the issue of identifying hate speech at scale is still an unsolved problem. In this paper, we propose a robust neural architecture which is shown to perform in a satisfactory way across different languages, namely English, Italian and German. We address an extensive analysis of the obtained experimental results over the three languages to gain a better understanding of the contribution of the different components employed in the system, both from the architecture point of view (i.e., Long Short Term Memory, Gated Recurrent Unit, and bidirectional Long Short Term Memory) and from the feature selection point of view (i.e., ngrams, social network specific features, emotion lexica, emojis, word embeddings). To address such in-depth analysis, we use three freely available datasets for hate speech detection on social media on English, Italian and German.

1 Introduction

The use of social media platforms such as Twitter, Facebook and Instagram has enormously increased the number of online social interactions, connecting billions of users, favouring the exchange of opinions and giving visibility to ideas that would otherwise be ignored by traditional media. However, this has led also to an increase of attacks targeting specific groups of users based on their religion, ethnicity or social status, and individuals often struggle to deal with the consequences of such offenses.

This problem affects not only the victims of online abuse, but also stakeholders such as governments and social media platforms. For example, Facebook, Twitter, YouTube and Microsoft have recently signed a code of conduct[1],

[1] http://ec.europa.eu/justice/fundamental-rights/files/hate_speech_code_of_

proposed by the European Union, pledging to review the majority of valid notifications for removal of illegal hate speech in less than 24 hours.

Within the Natural Language Processing (NLP) community, there have been several efforts to deal with the problem of online hate speech detection, since the computational analysis of language can be used to quickly identify offenses and ease the removal of abusive messages. Several workshops [61, 27] and evaluation campaigns [26, 12, 66, 7, 68] have been recently organised to discuss existing approaches to hate speech detection, propose shared tasks and foster the development of benchmarks for system evaluation. These have led to the creation of a number of datasets for hate speech detection in different languages, that have been shared within the NLP research community. Recent advances in deep learning approaches to text classification have then been applied also to deal with this task, achieving for some languages state-of-the-art results [17, 29, 31]. These systems are usually tailored to deal with social media texts by applying pre-processing, using domain-specific embeddings, adding textual features, etc. Given the number of configurations and external resources that have been used by systems for hate speech detection, it is rather difficult to understand what makes a classifier robust for the task, and to identify recommendations on how to pre-process data, what kind of embeddings should be used, etc. This is indeed the main contribution of the current paper: after identifying a deep learning architecture that is rather stable and well-performing across different languages, we evaluate the endowments of several components that are usually employed in the task, namely the type of embeddings, the use of additional features (text-based or emotion-based), the role of hashtag normalisation and that of emojis. We perform our comparative evaluation on English, Italian and German, focusing on freely available Twitter datasets for hate speech detection. Our goal is to identify a set of recommendations to develop hate speech detection systems, possibly going beyond language-specific differences.

The article is organised as follows: in Section 2 we present past work related to hate speech detection. In Section 3, we describe the neural architecture adopted in our experiments, while in Section 4 we present both the datasets used to train and test our classifier, and the external resources fed to the system. In Section 5, the experimental setup is presented, with details on the pre-processing step and the selection of hyperparameters. Finally, Section 6 reports on the evaluation results and discusses suggestions for the development of robust hate speech detection systems. In Section 7, we summarise our findings.

NOTE: This paper contains examples of language which may be offensive to some readers. They do not represent the views of the authors.

conduct_en.pdf

2 Related work

2.1 Hate speech detection on English data

Given the well-acknowledged rise in the presence of toxic and abusive speech on social media platforms like Twitter and Facebook, an increasing number of approaches has been proposed to detect such a kind of messages in English. Automated systems for the detection of abusive language range from supervised machine learning models built using a combination of manually crafted features such as n-grams [67], syntactic features [47], and linguistic features [23], to more recent neural networks that take word or character sequences from comments and learn abusive patterns without the need for explicit feature engineering. The recent trend of using neural network-based approaches has been particularly evident for English, since several training datasets are available for this language, enabling more data-hungry approaches. Indeed, organisers of the 2019 Semeval task on Offensive Language Identification [68] report that 70% of the participants adopt a deep learning approach. However, also simpler classification systems using logistic regression have been successfully applied to the task [62, 22]. Among the neural network-based approaches, different algorithms have been presented, such as Convolutional Neural Network using pre-trained word2vec embeddings [69], bi-LSTM with attention mechanism [1] and bidirectional Gated Recurrent Unit network [40]. More recently, also the combination of different neural newtorks, capturing both the message content and the Twitter account metadata, has been proposed [29]. In a comparative study of various learning models on the *Hate and Abusive Speech on Twitter* dataset built by Founta et al. [30], Lee et al. [40] show that, in the classification of tweets as "normal", "spam", "hateful" and "abusive" a bidirectional Gated Recurrent Unit network trained on word-level features is the most accurate model. Instead, in the binary task of offensive language detection, Liu et al. [41] achieve the best performance at Semeval 2019 by fine-tuning a bidirectional encoder representation from transformer [24].

In this paper, we propose a robust neural classifier for the hate speech binary classification task which is performing well across different languages (English, Italian and German), and we study the impact of each feature and component on the results across these languages. Our recurrent neural architecture shares some elements with the above approaches, namely the use of a Long Short Term Memory and a Gated Recurrent Unit. Embeddings, textual and social network specific features are employed. As in [38], we do not use metadata related to the social media accounts. The obtained results are compared in a more detailed way in Section 6.

2.2 Hate speech detection on languages different from English

While most approaches to hate speech detection have been proposed for English, other systems have been developed to deal with the task in German, Italian and Spanish, thanks to recent shared tasks. The 2018 GermEval Shared Task on the *Identification of Offensive Language*[2] deals with the detection of offensive comments from a set of German tweets. The tweets have to be classified into the two classes *offense* and *other*, where the *offense* class covers abusive language, insults, as well as profane statements. Different classifiers are used by the participants, ranging from traditional feature-based supervised learning (i.e., SVMs for the top performing system TUWienKBS [48]) to the more recent deep learning methods. Most top performing systems in both shared tasks employed deep learning (e.g., spMMMP [60], uhhLT [63], SaarOffDe [25], InriaFBK [19]). For example, SaarOffDe employs Recurrent Neural Networks and Convolutional Neural Networks produced top scores, while other systems (e.g., spMMMP, uhhLT) employ transfer learning. The usage of ensemble classification seems to often improve the classification approaches (e.g., Potsdam [54], RuG [5], SaarOffDe, TUWienKBS, UdSW [64]). Concerning the features, several systems include a combination of word embeddings, character n-grams and some forms of (task-specific) lexicon. Both the HaUA and the UdSW systems report that high performance scores can be achieved with a classifier solely relying on a lexicon.

In 2018, the first *Hate Speech Detection* (HaSpeeDe) task for Italian has been organized at EVALITA-2018[3]. The task consists in automatically annotating messages from Twitter and Facebook, with a boolean value indicating the presence (or not) of hate speech. Similar to Germeval 2018 submissions, also in this case the participating systems adopt a wide range of approaches, including biLSTM [39], SVM [53], ensemble classifiers [52, 4], RNN [28], CNN and GRU [60]. The authors of the best-performing system, ItaliaNLP [17], experiment with three different classification models: one based on linear SVM, another one based on a 1-layer BiLSTM and a newly-introduced one based on a 2-layer BiLSTM which exploits multi-task learning with additional data from the 2016 SENTIPOLC task[4].

Concerning Spanish, the IberEval 2018 edition[5] has proposed the *Aggressiveness Detection* task [13] applied to Mexican Spanish, aiming at providing a classification of aggressive / non-aggressive tweets. A variety of systems is proposed, exploiting content-based (bag of words, word n-grams, term vectors, dictionary words, slang words) and stylistic-based features (frequencies, punctuation, POS, Twitter specific elements). Most of the systems rely on neural networks (CNN, LSTM and others). The top ranked team was INGEOTEC [32]: the system is based on MicroT, a text classification approach supported by a lexicon-based model that takes into account the presence of aggressive and affective words, and a model based on the Fasttext representation of texts. More recently, a task for the detection of hate speech against immigrants and women on Twitter has been organised at Semeval 2019 [7], providing an English and

[2] `https://www.oeaw.ac.at/fileadmin/subsites/academiaecorpora/PDF/GermEval2018_Proceedings.pdf`

[3] `http://www.evalita.it/2018`

[4] `http://www.di.unito.it/~tutreeb/sentipolc-evalita16/index.html`

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

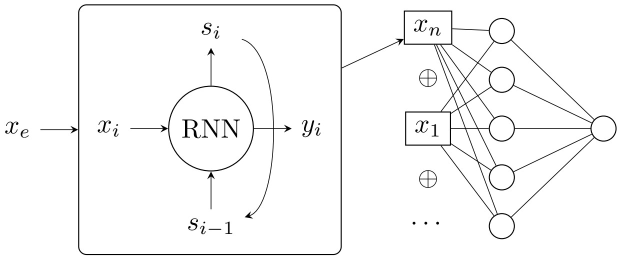


Figure 1: The modular neural architecture

Spanish dataset annotated according to the same guidelines. While for both languages a number of neural network approaches has been proposed, the best systems for hateful content detection still rely on SVM and embedding-based features [37, 50, 2].

Looking at the descriptions of the systems participating in the above tasks, as well as at most recent hate speech detection classifiers for English, we observe that deep learning approaches usually share a number of features, such as word embeddings, the use of emotion or sentiment lexica, as well as specific pre-processing steps. Many exploit also other features related to the tweets (e.g. message length, punctuation marks, etc.). Nevertheless, more emphasis is usually put on the architecture, and no insight is given into the role played by variants of the above features and by the selected pre-processing strategy. Also, no attempt to understand differences across different languages has been made. This motivates the experiments presented in the remainder of this paper.

3 Classification framework

Since our goal is to compare the effect of various features, word embeddings and pre-processing techniques on hate speech detection, we use a modular neural architecture for binary classification that is able to support both word-level and message-level features. The components are chosen to support the processing of social-media specific language. The neural architecture and the features are detailed in the following subsections.

3.1 Modular Neural Architecture

We use a modular neural architecture (see Figure 1) in Keras [15]. The architecture that constitutes the base for all the different models uses a single feed-forward hidden layer of 100 neurons, with a ReLu activation and a single output with a sigmoid activation. The loss used to train the model is binary cross-entropy. We choose this particular architecture because we used it to participate to two shared tasks for hate speech detection, EVALITA HaSpeeDe 2018 [18] for Italian and Germeval 2018 [19] for German, and it proved to be effective and robust for both languages, also across different social media platforms [20]. In particular, in our original submissions the same architecture was ranked fourth in the Twitter EVALITA subtask (-1.56 F1 compared to the first ranked) and seventh in the Germeval coarse-grained classification task (-2.52 F1 from the top-ranked one).

The architecture is built to support both word-level (i.e. embeddings) and tweet-level features. In particular, we use a recurrent layer to learn an encoding (x_n in the Figure) derived from word embeddings, obtained as the output of the recurrent layer at the last timestep. This encoding gets then concatenated with the other selected features, obtaining a vector of tweet-level features.

Since the models derived from using different features are different both in terms of number of parameters and in terms of layers, we decided to keep the size of the hidden layer fixed. This allows us to compare different features, as the latent representation learned by the hidden layer that is ultimately used to classify the tweets has the same size regardless of the number and kind of features.

More formally, given an input represented as the set of features $X = \{x_j | x_j \in X_m\}$, where X_M is the set of all features supported by a model M (see Section 3.2) and s is the sum of the dimensions of all the features, we compute a function:

$$\begin{gathered} M(X) = s(W_o H(X) + b_o) \quad W_o \in R^{1\times 100} \quad H(X) \in R^{100} \quad b_o \in R^1 \\ s(X) = (\sigma(x_1), \ldots, \sigma(x_n)) \quad x \in R^n \\ \sigma(x) = \frac{1}{1+e^{-x}} \end{gathered}$$

where W_o and b_o are the learned weights for the output layer and $\sigma(x)$ is the sigmoid activation function (note that in all the models we used $n = 1$ as we only have one binary output), and:

$$\begin{gathered} H(X) = g(W_h C(X) + b_h) \quad W_h \in R^{100\times s} \quad C(X) \in R^s \quad b_h \in R^{100} \\ g(X) = (f(x_1), \ldots, f(x_n)) \quad x \in R^n \\ f(x) = max(0, x) \quad x \in R \end{gathered}$$

where $H(X)$ represents the application of a hidden layer of size 100 and learned weights W_h and b_h and $g(x)$ is the ReLU activation function. Additionally:

$$C(X) = \bigoplus_{x_i \in X} R(x_i)$$

where $\bigoplus$ denotes the concatenation of all vectors along their axes. For example, if we have a set of vectors $X = [x_1, x_2, x_3]$, then:

$$\bigoplus_{x_i \in X} x_i \in R^{a+b+c} \quad x_1 \in R^a, x_2 \in R^b, x_3 \in R^c$$

Finally:

$$R(x) = \begin{cases} x & \text{if } x \text{ is a tweet-level feature} \\ RNN(x) & \text{if } x \text{ is a word-level feature} \end{cases}$$

where RNN is the function returning the output by a recurrent layer at the last timestep.

3.2 Features

In our experiments, we use the following features, with the goal of evaluating their impact on a hate speech detection model:

- **Word Embeddings** (x_e in Figure 1): multiple word embeddings from various sources have been tested (for a full description of the different embeddings see Section 4.2). We evaluate in particular the contribution of word embeddings extracted from social media data, therefore belonging to the specific domain of our classification task, compared with the performance obtained using generic embedding spaces, like Fasttext [11], which are widely used across different NLP tasks because of their good coverage.

- **Emoji embeddings**: emojis are a peculiar element of social media texts. They are often used to emphasize or reverse the literal meaning of a short message, for example in ironic or sarcastic tweets [36]. It is therefore very important for hate speech detection to understand which is the best way to represent them and to include them in the embedding space. We compare different ways to embed emoji information in our classifier: *i)* we use embedding spaces created from social media data, where each emoji is also represented through a word embedding, or *ii)* in case of generic embedding spaces, where emojis are not present, we include emoji embeddings through the alignment of different spaces following the approach presented in [58], or *iii)* in order to cope with the low coverage of emojis, they are replaced by their description in plain text as suggested in [56].

- **Ngrams**: unigrams (x_1 in Figure 1) and bigrams derived from the tweets are also included as features. We first tokenize and lemmatize the tweets by using Spacy [35], then normalize the tweet-level ngram occurrence vector by using tf-idf. Our intuition is that these features should capture lexical similarities between training and test data, therefore they should be predictive when training and test set deal with the same type of offenses. Higher-level ngrams are not considered, as we expect them to be very sparse especially in social media, where tweets do not follow standard writing conventions.

- **Social-network specific features**: The character limit imposed by some social media platforms like Twitter affects the style in which messages are written: function words tend to be skipped, texts are very concise while punctuation and uppercase words are used to convey effective messages despite their brevity. Therefore, all these linguistic indicators can be used to identify the presence of hateful messages. We consider in particular the number of hashtags and mentions, the number of exclamation and question marks, the number of emojis, the number of words that are written in uppercase at the tweet-level. These features are then normalized by subtracting their mean and dividing them by their standard deviation.

- **Emotion lexica**: several emotion lexica have been (manually or automatically) created and used in classification tasks to represent the emotional content of a message [45, 44, 9, 59]. While the importance of emotion information to hate speech detection may seem evident [3], it is also true that an embedding space which is large and representative enough of the domain may make additional emotion features redundant. We therefore evaluate the contribution of emotion information using two freely available, multilingual emotion lexica, namely EmoLex and Hurtlex. Emolex [45, 44] is a large list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive), manually annotated with Amazon's Mechanical Turk. The creators of the lexicon have additionally made available a multilingual version of the resource, that was created by translating each word with Google translate in 2017. We therefore use the German and Italian translations as well as the English one. Using EmoLex, we extract two sentiment-related features and eight emotion-related features for each tweet by summing all the sentiment and emotion scores assigned to the words in a tweet and normalizing them by using tf-idf. The second resource, i.e., Hurtlex [9], is a multilingual lexicon of hate words created starting from the Italian hate lexicon developed by the linguist Tullio De Mauro, organized in 17 categories. It has been expanded through the link to available synset-based computational lexical resources such as MultiWordNet [51] and Babelnet [46], and evolved in a multilingual perspective by semi-automatic translation and expert annotation. Since Hurtlex may contain the same word multiple times with different Part-of-Speech tags, we performed a union operation over the categories in order to represent all the categories that a word can belong to, independently of the POS. Using HurtLex, we assign with the same strategy a score for *negative stereotypes*, one for *hate words and slurs* and one for *other insults* to each tweet.

4 Data and linguistic resources

In the following, we present the datasets used to train and test our system for English, Italian and German and their annotations (Section 4.1). Then, we describe the word embeddings (Section 4.2) we have used in our experiments.

4.1 Datasets

English We use the dataset described in [62], containing 16k English tweets manually annotated for hate speech. More precisely, 1,924 are annotated as containing racism, 3,082 as containing sexism, while 10,884 tweets are annotated as not containing offensive language. We merge the sexist and racist tweets in a single class, so that 5,006 tweets are considered as positive instances of hate speech, as in Example 1.

1. Annotation: hateful.
   Since 1/3 of all #Islam believes that people who leave the religion should be murdered where are the moderate Muslim.

Italian We use the Twitter dataset released for the HaSpeeDe (Hate Speech Detection) shared task organized at Evalita 2018, the evaluation campaign for NLP and speech processing tools for Italian[6]. This dataset includes a total amount of 4,000 tweets [12], comprising for each tweet the respective annotation, as can be seen in Example 2. The two classes considered in the annotation are “hateful post” or “not”.

2. Annotation: hateful.
   altro che profughi? sono zavorre e tutti uomini (EN: Are they really refugees? they are ballast and all men).

German We use the dataset distributed for the shared task on the Identification of Offensive Language organized at Germeval 2018, a workshop in a series of shared tasks on German processing[7]. The dataset provided for task 1, where offensive comments are to be detected from a set of German tweets (binary classification), consists of 5,009 German tweets manually annotated at the message level [66] with the labels “offense” (abusive language, insults, and profane statements) and “other” (i.e. not offensive). More specifically, 1,688 messages are tagged as “offense” (see Example 3), while 3,321 messages as “other”.

3. Annotation: Offense.
   @Ralf_Stegner Oman Ralle..dich mag ja immer noch keiner. Du willst das die Hetze gegen dich aufhort? Geh in Rente und verzichte auf die 1/2deiner Pension (EN: @Ralf_Stegner Oman Ralle... still, nobody likes you. You want to stop hate against you? Retire and give up half of your pension).

Table 1 summarizes the main statistics on the datasets. The reported values show that, although the datasets have different sizes, the distribution between positive and negative examples is similar. We also manually investigated data samples and the annotation schemes of the English, German and Italian datasets. Although the developers of the English and the Italian corpus focus on hate speech, while the Germeval organisers claim to target offensive language, the kind of messages they annotate as belonging to their respective ‘positive’ class largely overlap. The targets are different, i.e. the Italian messages focus on immigrants, Muslim and Roma, the English ones on sexist and racial offenses, while the German one has no specific targets, and includes both offensive messags towards groups and towards individuals. However, the types of offenses,

[6] `http://www.di.unito.it/~tutreeb/haspeede-evalita18`

[7] `https://www.oeaw.ac.at/ac/konvens2018/workshop/`

Dataset	# hate speech/offensive (%)	# other (%)	# total
English	5,006 (32%)	10,884 (68%)	16,000
Italian	1,296 (32%)	2,704 (68%)	4,000
German	1,688 (34%)	3,321 (66%)	5,009

Table 1: Statistics on the datasets

both explicit and implicit, including sarcastic messages, rhetorical questions and false claims based on prejudices make them in our view comparable. The only difference is the set of messages labeled as 'Profanity' and included among the 'Offensive' ones in the German dataset, which covers slurs without a specific target. However, they account only for 1.4% messages in this training set.

4.2 Word Embeddings

In our experiments we test several embeddings, with the goal to compare generic with social media-specific ones. In order to have a high coverage of emojis, we also experiment with aligned embedding spaces obtained by aligning the English, Italian and German ones. Another element we take into account is the access to the binary Fasttext model that originates the embedding space. When using that binary model, it is possible to greatly mitigate the problem of out-of-vocabulary words, since the system is able to provide an embedding for unknown words by using subword unit information [42]. The binary model is often made available together with the standard model when pre-trained embeddings are released. When available, we always use this version. The tested embeddings, summarised in Table 2, are the following:

- **Fasttext embeddings for German and Italian**: we use embedding spaces obtained directly from the Fasttext website[8] for German and Italian. In particular, we use the Italian and German embeddings trained on Common Crawl and Wikipedia [33] with size 300. A binary Fasttext model is also available and was therefore used;

- **English Fasttext Crawl embeddings**: English embeddings trained by Fasttext[9] on Common Crawl, with an embedding size of 300. A binary Fasttext model is provided;

- **English Fasttext News embeddings**: English embeddings trained on Wikipedia 2017 using subword information, UMBC web base corpus and statmt.org and released by Fasttext [10], with an embedding size of 300. The available binary Fasttext model was used;

- **Italian Twitter embeddings**: we trained Fasttext embeddings from a sample of Italian tweets [8], with embedding size of 300. We used the binary version of the model;

[8] `https://fasttext.cc/docs/en/crawl-vectors.html`

[9] `https://fasttext.cc/docs/en/english-vectors.html`

[10] `https://fasttext.cc/docs/en/english-vectors.html`

- **German Twitter embeddings**: trained by Spinning Bytes[11] from a sample of German tweets [16]. We used the model with embeddings of size 300. A binary Fasttext model was not provided, we therefore used the word-based version;
- **English Twitter embeddings**: English Fasttext embeddings from Spinning Bytes[12], trained on an English Twitter sample [16] with an embedding size of 200. Since a binary Fasttext model was not provided, we used the word-based version;
- **Aligned embeddings**: since Fasttext embeddings for Italian and German do not contain emojis, we extend them by aligning them with an English embedding space containing emojis [6], following the alignment approach presented in [57]. All embeddings and the resulting aligned spaces have a size of 300.

EMBEDDINGS	LANGUAGE	ALGORITHM	SIZE	FASTTEXT BINARY MODEL
Fasttext En CCrawl	EN	Fasttext	300	YES
Fasttext En News	EN	Fasttext	300	YES
Twitter English	EN	Fasttext	200	NO
Fasttext It CCrawl & Wiki	IT	Fasttext	300	YES
Twitter Italian	IT	Fasttext	300	YES
Fasttext De CCrawl & Wiki	DE	Fasttext	300	YES
Twitter German	DE	Fasttext	300	NO
Aligned	EN,IT,DE	Fasttext	300	NO

Table 2: Overview of the different embeddings used in our experiments

In summary, we were able to use a binary model for all the official Fasttext monolingual datasets and the Italian Twitter embeddings that we trained. For the remaining embedding spaces, we only had access to a dictionary-like structure, that contains the embedding for each word in the vocabulary.

5 Experiments

In this section, we detail the setup of our experiments, including the preprocessing step, the selection of hyperparameters and the combination of features and configurations tested for each language.

5.1 Preprocessing

Since hashtags, user mentions, links to external media and emojis are common in social media interactions, it is necessary to carefully preprocess the data,

[11] https://www.spinningbytes.com/resources/wordembeddings/

[12] https://www.spinningbytes.com/resources/wordembeddings/

in order to normalize the text as much as possible while retaining all relevant semantic information. For this reason, we first replace URLs with the word “url” and “@” user mentions with “username” by using regular expressions. Since hashtags often provide important semantic content, we wanted to test how splitting them into single words would impact on the performance of the classifier. To this end, we use the Ekphrasis tool [10] to do hashtag splitting and evaluate the classifier performance with and without splitting. Since the aforementioned tool only supports English, it has been adapted to Italian and German by using language-specific Google ngrams.[13]

Another pre-processing step we evaluate in our experiments is the description of emojis in plain text, that proved to benefit tweet classification [56] but was evaluated so far only on English. In order to map each emoji with a description, we first retrieve an emoji list using the dedicated Python library[14] and replace each emoji with its English description according to the website of the Unicode consortium[15]. We then translate the descriptions using Google Translate and fix any mistakes by hand. In this way we create a list of emojis with the corresponding transcription in three languages (available at `https://github.com/dhfbk/emoji-transcriptions`).

5.2 Hyperparameters

In order to keep our setting robust across languages, we base our model on a configuration that performed consistently well on all subtasks of Evalita hate speech detection [18], both on Facebook and on Twitter data, even if it was not the best performing configuration on the single tasks. In particular, our model uses no dropout and no batch normalization on the outputs of the hidden layer. Instead, a dropout on the recurrent units of the recurrent layers is used. We select a batch size of 32 for training and a size of 200 for the output (and hidden states) of the recurrent layers. We also test the impact of different recurrent layers, namely long short-memory (LSTM) [34], gated recurrent unit (GRU) [14] and bidirectional LSTM (BiLSTM) [55].

5.3 Settings

In our experiments, we perform a series of tests on the aforementioned modular neural model, concerning the following aspects:

- For each language, we test the corresponding embeddings presented in Section 4.2.
- We test all possible combination of features: embeddings, unigrams, bigrams, social features, EmoLex and Hurtlex.

[13] `http://storage.googleapis.com/books/ngrams/books/datasetsv2.html`

[14] `https://github.com/carpedm20/emoji`

[15] `https://www.unicode.org/emoji/charts/full-emoji-list.html`

- We test three possible recurrent layers, namely LSTM, GRU and Bidirectional LSTM.
- We train models with and without hashtag splitting.
- We test models that replace emojis with their description and models that do not. For Italian and German Fasttext embeddings that do not contain emojis, we also test the model performance after using emoji embeddings resulting from alignment with an English embedding space (see details in Section 4.2)

Overall, we compare 1,800 possible configurations for English, 1,080 for Italian and 1,224 for German. The difference is due to the availability of more embedding spaces for English, which increase the amount of possible settings and feature combinations to be tested.

Concerning the dataset splits into training and test instances, for the English dataset - since no standardized split is provided - we randomly selected 60% of the dataset for training, 20% for validation and 20% for testing. Since we want our experiments to be reproducible, we use the *train_test_split* function from scikit-learn [49] to shuffle and split the dataset 60%/40%. The remaining 40% was then split in half to obtain the validation and test set, respectively. We use 42 as a seed value for the random number generator used for shuffling.

The German dataset was already randomly split by the GermEval task organizers into training and test set, containing 5,009 and 3,532 messages respectively. For our experiments we keep the same split as proposed in the challenge, but we use 20% of the training set as validation set, obtained by invoking *train_test_split* from scikit learn with 42 as seed. Similarly, the Italian dataset was randomly split by the HaSpeeDe task organizers into training and test set of 3,000 and 1,000 messages respectively. Again, in our experiments we keep the same split as proposed in the challenge, but we used 20% of the training set as validation set applying the same function as for the German dataset.

For each language, the validation test is used to evaluate the classifier performance over 20 training epochs and select the best performing model in terms of macro averaged F1 score. The selected model is then used to evaluate performance on the test set.

6 Evaluation

In this section, we report a selection of the most relevant results from the pool of settings described in Section 5.3. In particular, the first row of Table 3, 4 and 5 reports the best run over all the configurations tested for English, Italian and German respectively, while the other rows show how the best performance changes when modifying one parameter at a time. We also provide an evaluation of the effectiveness of different configurations by comparing the three languages after downsampling the training sets. As comparison we provide a baseline obtained running a SVM (linear kernel) with a bag of word approach using tf-idf as weight.

6.1 Multilingual evaluation on the complete datasets

For **English**, the best result (0.823 F1) is obtained using an LSTM network and the Fasttext embeddings trained on Common Crawl. Table 3 shows how adding or removing single features from the best configuration affects the result: adding unigram and bigram-based features to the classifier leads to the largest drop in performance, while changing other features the impact is lower. This confirms the findings in [62], in which character n-grams outperform word n-grams in the classification of racial, sexist and not-offensive tweets. Overall we find that, although the best result is obtained using an LSTM network, replacing LSTM with Bi-LSTM keeping the same features achieves similar results, with a difference of F1 of 0.1-0.2% F1. This shows that having both forward and backward information when dealing with tweets is probably not needed because of the limited length of the messages. The use of hashtag normalization to split the hashtags into words improves the system performance in every configuration, increasing the coverage of the embeddings. Overall, the coverage of Fasttext embeddings trained on CommonCrawl is sufficient to deal with Twitter data, therefore adding specific embeddings or pre-processing them is not necessary. Also, the SVM baseline suffers from lower recall compared to the best neural configuration, especially when dealing with the hate category, that has less training instances.

EMBED.	TEXT FEATS	SOCIAL	EMOTIONS	EMOJI	NETWORK	HASH. SPLIT	F1 NO HATE	F1 HATE	P AVG	R AVG	F1 AVG
Fasttext CCrawl	emb	NO	NO	NO	lstm	YES	0.885	0.760	0.820	0.825	**0.823**
Fasttext CCrawl	emb	NO	NO	NO	lstm	**NO**	0.879	0.745	0.811	0.814	0.812
Fasttext CCrawl	emb	NO	NO	**transcr.**	lstm	YES	0.886	0.756	0.821	0.821	0.821
Fasttext CCrawl	emb	**YES**	NO	NO	lstm	YES	0.883	0.757	0.817	0.823	0.820
Fasttext CCrawl	emb	**YES**	**EMOLEX**	NO	lstm	YES	0.887	0.751	0.823	0.815	0.819
Fasttext CCrawl	emb	NO	**EMOLEX**	NO	lstm	YES	0.885	0.751	0.820	0.817	0.818
Fasttext CCrawl	emb	NO	**HURTLEX**	NO	lstm	YES	0.884	0.744	0.818	0.810	0.814
Fasttext CCrawl	emb	**YES**	**HURTLEX**	NO	lstm	YES	0.883	0.742	0.816	0.809	0.812
Fasttext CCrawl	**emb+uni+bi**	NO	NO	NO	lstm	YES	0.881	0.719	0.815	0.790	0.800
Fasttext CCrawl	**emb+uni**	NO	NO	NO	lstm	YES	0.871	0.711	0.796	0.787	0.791
Fasttext CCrawl	**emb+bi**	NO	NO	NO	lstm	YES	0.873	0.694	0.800	0.773	0.784
SVM baseline							0.875	0.682	0.808	0.763	0.778

Table 3: Best performing configuration on English data (Macro AVG). EMOJI = ‘NO’ means that no specific processing of emoji was applied

For **Italian**, the best result (0.805 F1) is obtained with a configuration using a LSTM network and the word embeddings we trained on a large corpus of Italian tweets. In Table 4 we show to what extent the different features affect the performance obtained with the best configuration. On Italian, differently from English and German, the use of unigrams in addition to word embeddings is beneficial to the classifier performance. The best result is obtained using the emoji transcription, but their impact is not significant (0.805 F1 using them vs. 0.804 not using them). The same trend can be found also with different configurations not reported in the table. Considering all runs with all configurations, the use of embeddings trained on the same domain of the dataset (Italian Tweets) always leads to better results compared with the use of more generic embeddings as the ones from Fasttext (trained on Common Crawl and Wikipedia). Almost all the best performing configurations take advantage of the use of hashtag splitting. BiLSTM performs generally worse than LSTM. Like in the English evaluation, the SVM baseline achieves a remarkably lower performance on the hate class, and shows recall issues.

EMBED.	TEXT FEATS	SOCIAL	EMOTIONS	EMOJI	NETWORK	HASH. SPLIT	F1 NO HATE	F1 HATE	P AVG	R AVG	F1 AVG
Twitter	emb+uni	NO	NO	transcription	lstm	YES	0.867	0.736	0.803	0.806	**0.805**
Twitter	emb+uni	YES	NO	transcription	lstm	YES	0.872	0.737	0.803	0.806	**0.805**
Twitter	emb+uni	NO	NO	**NO**	lstm	YES	0.871	0.736	0.802	0.805	0.804
Twitter	emb+uni	NO	**HURTLEX**	transcription	lstm	YES	0.867	0.728	0.795	0.800	0.797
Twitter	emb+uni	NO	NO	transcription	lstm	**NO**	0.861	0.727	0.789	0.800	0.794
Twitter	emb+uni	**YES**	**HURTLEX**	transcription	lstm	YES	0.863	0.723	0.790	0.796	0.793
Twitter	emb+uni	**YES**	**EMOLEX**	transcription	lstm	YES	0.864	0.718	0.790	0.792	0.791
Twitter	emb+uni	NO	**EMOLEX**	transcription	lstm	YES	0.858	0.719	0.784	0.794	0.788
Twitter	**emb**	NO	NO	transcription	lstm	YES	0.862	0.697	0.785	0.775	0.779
aligned	emb+uni	NO	NO	**embeddings**	lstm	YES	0.872	0.676	0.809	0.758	0.774
Twitter	**emb+bi**	NO	NO	transcription	lstm	YES	0.860	0.660	0.783	0.747	0.760
Twitter	**emb+uni+bi**	NO	NO	transcription	lstm	YES	0.847	0.690	0.766	0.771	0.768
SVM baseline							0.855	0.593	0.781	0.707	0.724

Table 4: Best performing configuration on Italian data (Macro AVG).

Table 5 reports the results obtained on **German** data. The best result is achieved with a GRU network, using the standard Fasttext embeddings (trained on Common Crawl and Wikipedia). Similar to English, adopting unigrams and bigrams as feature leads to a decrease in performance (0.05 points F1). Considering all the experiments run on German data, the results confirm that also for this language emoji transcriptions perform better than the emoji vectors obtained through multilingual alignment, but for the best configuration no specific emoji processing is needed. Hashtag splitting, which is included in the best performing configuration for English and Italian, is instead not beneficial to German tweet classification. Our intuition is that, since German is rich in compound words, Ekphrasis hashtag normalization approach based on Google n-grams tends to split terms also when it is not needed. Although social and emotion features are not used in the best output, they appear to help in most of the other configurations. Also for this language, the SVM baseline achieves a lower recall and less accurate classification on the hate speech class than the neural model.

EMBED.	TEXT FEATS	SOCIAL	EMOTIONS	EMOJI	NETWORK	HASH. SPLIT	F1 NO HATE	F1 HATE	P AVG	R AVG	F1 AVG
Fasttext	emb	NO	NO	NO	GRU	NO	0.829	0.686	0.754	0.762	**0.758**
Fasttext	emb	NO	NO	NO	GRU	**YES**	0.843	0.640	0.765	0.730	0.741
Fasttext	emb	NO	NO	**transcription**	GRU	NO	0.839	0.644	0.758	0.732	0.741
aligned	emb	NO	NO	**embeddings**	GRU	NO	0.835	0.652	0.752	0.738	0.744
Fasttext	emb	**YES**	**HURTLEX**	NO	GRU	NO	0.834	0.671	0.755	0.751	0.753
Fasttext	emb	**YES**	NO	NO	GRU	NO	0.836	0.671	0.759	0.751	0.754
Fasttext	emb	**YES**	**EMOLEX**	NO	GRU	NO	0.836	0.657	0.755	0.741	0.747
Fasttext	emb	NO	**EMOLEX**	NO	GRU	NO	0.840	0.655	0.760	0.740	0.748
Fasttext	emb	NO	**HURTLEX**	NO	GRU	NO	0.843	0.654	0.764	0.739	0.748
Fasttext	**emb+bi**	NO	NO	NO	GRU	NO	0.806	0.606	0.710	0.703	0.706
Fasttext	**emb+uni+bi**	NO	NO	NO	GRU	NO	0.821	0.590	0.726	0.697	0.706
Fasttext	**emb+uni**	NO	NO	NO	GRU	NO	0.819	0.586	0.722	0.694	0.702
SVM baseline							0.807	0.374	0.692	0.597	0.591

Table 5: Best performing configuration on German data (Macro AVG).

Beside the aforementioned experiments, we perform an additional evaluation using a character-based RNN. Indeed, character-based representations have been recently used in several NLP tasks including abusive language detection [43] with promising results thanks to their ability to effectively handle rare and unseen words. We use the best performing systems for the three languages, replacing word-based RNN with a character-based one. In order to learn a dense representation for characters, we used a learned embedding layer with size 10. The results of this set of experiments are reported in Table 6, and show that using a character-based RNN the performance of the system drops significantly in all three languages compared to word-based RNNs, probably because Fasttext embeddings already account for subword information. We therefore decided not to perform further tests with this configuration.

LANG	TEXT FEATS	SOCIAL	EMOTIONS	EMOJI	NETWORK	HASH. SPLIT	F1 NO HATE	F1 HATE	P AVG	R AVG	F1 AVG
EN	char	NO	NO	NO	lstm	YES	0.821	0.489	0.697	0.645	0.655
IT	char+uni	NO	NO	transcription	lstm	YES	0.845	0.540	0.763	0.677	0.692
DE	char	NO	NO	NO	GRU	NO	0.771	0.212	0.555	0.524	0.491

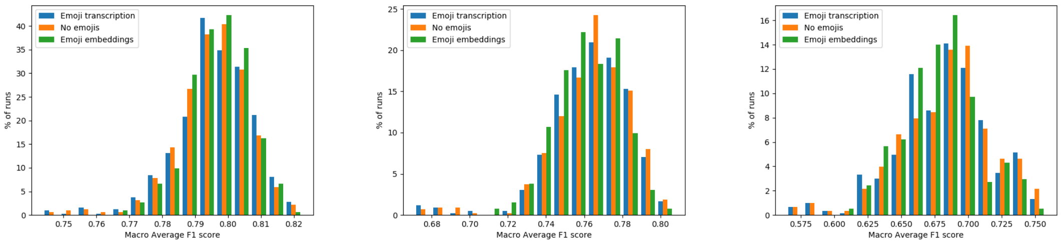
Table 6: Results of character based RNN using the best configurations for the three languages.

Language	Emoji	AVG F1	Max F1	Standard deviation F1	Number of runs
EN	NO	0.796	0.823	0.034	612
EN	YES	0.797	0.821	0.009	576
EN	Transcription	0.796	0.821	0.034	612
IT	NO	0.764	0.804	0.021	468
IT	YES	0.761	0.798	0.016	144
IT	Transcription	0.763	0.805	0.021	468
DE	NO	0.684	0.758	0.034	468
DE	YES	0.678	0.745	0.028	288
DE	Transcription	0.682	0.754	0.034	468

Table 7: Mean and Standard deviation of macro averaged F1 scores without any specific processing of emojis ('NO'), using emojis obtained through alignment ('YES') and transcribing them ('TRANSCRIPTION')

6.2 Contribution of social and emotion information

In order to better understand the contribution of specific features or pre-processing steps on all the system runs, we present a comparative evaluation of the classifier performance with or without emoji transcription (in Figure 2) and with or without social and emotion features (Figure 3). This analysis is done with the goal of focusing not only on the best performing configuration, but also on general trends that could not be included in the previous tables. In particular, we plot the distribution of runs achieving different macro average F1 scores.



(a) English (total: 1800 runs)

(b) Italian (total: 1080 runs)

(c) German (total: 1224 runs)

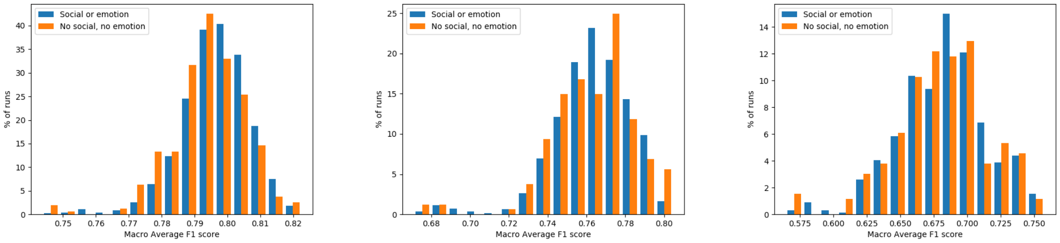
Figure 2: Results distribution with and without emoji transcription and using aligned emoji embeddings over the three languages.

Figure 2 shows that transcribing emojis yields the best performance for English but not for the two other languages. Nevertheless, this distinction is not clear-cut, since no clear trend can be associated with this feature. More details on the different configurations are shown in Table 7, confirming the above findings. Figure 3 analyses in a similar way the contribution of social network specific features (i.e. tweet length, punctuation marks, uppercase, etc.) and emotion features (i.e. based on EmoLex and Hurtlex). It shows that, while

Language	Social & emotion features	Mean F1	Max F1	Standard Deviation F1	Number of Runs
EN	NO	0.794	0.823	0.011	300
EN	YES	0.797	0.821	0.010	1500
IT	NO	0.763	0.805	0.020	180
IT	YES	0.763	0.805	0.021	900
DE	NO	0.680	0.758	0.035	204
DE	YES	0.682	0.754	0.033	1020

Table 8: Mean and Standard deviation of macro averaged F1 scores with and without social and emotion features

for English and Italian the best results are obtained without these two groups of features, other runs achieving on average a slightly lower performance make use of this information. For German, the improvement due to social and emotion features appears to be more consistent, even if it does not apply to all runs. Also, the averaged results summarised in Table 8 confirm that, like for emojis, the differences are not clear-cut.



(a) English (total: 1800 runs)

(b) Italian (total: 1080 runs)

(c) German (total: 1224 runs)

Figure 3: Results distribution with and without social network and emotion features over the three languages.

Comparing the results across the three languages, we summarize the main findings from the evaluation as follows:

- Using subword information has a positive impact on our task, since it can deal with the high language variability and creativity in the social media domain as well as with typos.
- Creating specific embeddings that cover well the domain of interest is beneficial to the task performance. If possible, a large amount of Twitter data should be collected to create embeddings when dealing with online hate speech classification. If not, pretrained Fasttext embeddings trained on CommonCrawl or similar are recommended, provided that it is possible to access the binary model

- If the above domain-specific embeddings are available, where emojis are also present, our experiments show that it is not needed to pre-process emojis in specific ways (e.g. transcribe, add emoji embeddings through alignment)

- Hashtag normalization is useful to classify hate speech in English and Italian, but current approaches to hashtag splitting may not perform well on languages that are rich in compounds like German, which in turn may affect classification

- Using domain-specific embeddings with a good coverage make emotion lexica redundant in our experiments. The fact that such lexica may be manually or semi-automatically created does not play a major role in classification performance

- Given the limited length of tweets, LSTM yielded better results than BiLSTM

6.3 Multilingual evaluation on downsampled datasets

We perform an additional set of experiments to investigate to what extent the size of the dataset affects the results. Therefore, we downsample both the German and the English datasets to match the size of the Italian Twitter dataset, the smallest one. In order to improve our ability to compare the results, we use the same distribution of labels (hate speech, non hate speech) as the Italian dataset for the two downsampled ones. We then replicate some of the best performing configurations presented in the previous tables, and report the results in Table 9. As expected, reducing the training data both for English and for German leads to a drop in performance (from 0.823 F1 to 0.782 for English, from 0.758 F1 to 0.713 for German). On all the runs, the classifier achieves a lower performance on German than on the other two languages, while the results on Italian and English are comparable. Our experiments suggest that German is more challenging to classify, partly because of inherent characteristics of the language (for example the presence of compound words that makes hashtag splitting ineffective), partly because of the way in which the Germeval dataset was built. Namely, the organisers report that they sampled the data starting from specific users and avoiding keyword-based queries, so to obtain the highest possible variability in the offensive language. They also manually checked and enriched the data so to cover all the political spectrum in their offenses, and avoid user overlaps between training and test data. This led to the creation of a very challenging dataset, where lexical overlap between training and test data is limited (therefore unigram and bigram features do not work well) and where hate speech is not associated with specific topics or keywords.

EMBED.	TEXT FEATS	SOCIAL	EMOTIONS	EMOJI	NETWORK	HASH. SPLIT	LANG	F1 NO HATE	F1 HATE	P AVG	R AVG	F1 AVG
Fasttext	emb+uni	NO	NO	transcription	LSTM	YES	EN	0.847	0.683	0.763	0.769	0.765
							IT	0.863	0.739	0.794	0.811	**0.801**
							DE	0.822	0.578	0.726	0.690	0.700
Fasttext	emb+uni	NO	NO	NO	LSTM	YES	EN	0.844	0.696	0.763	0.780	0.770
							IT	0.862	0.723	0.789	0.796	0.793
							DE	0.814	0.563	0.712	0.680	0.689
Fasttext	emb	NO	NO	NO	GRU	NO	EN	0.846	0.701	0.767	0.783	0.773
							IT	0.857	0.708	0.780	0.785	0.783
							DE	0.827	0.598	0.736	0.703	0.713
Fasttext	emb	NO	NO	transcription	LSTM	YES	EN	0.857	0.713	0.780	0.792	**0.785**
							IT	0.849	0.684	0.767	0.766	0.767
							DE	0.824	0.611	0.732	0.710	**0.718**
Fasttext	emb	NO	NO	NO	LSTM	YES	EN	0.853	0.711	0.776	0.791	0.782
							IT	0.837	0.683	0.755	0.767	0.760
							DE	0.830	0.596	0.741	0.702	0.713

Table 9: Performance evaluation on data sets of comparable size in English, Italian and German

While our main goal is not to develop a system achieving state-of-the-art results, it is interesting to compare our performance with the best systems dealing with hate speech detection. For Italian and German our approach can be easily compared to other existing classifiers using the same training and test split, since we relied on the official data released in two shared tasks. These results, however, were obtained in the context of the shared task, therefore the authors could not use information about the test set performance as we did. The comparison is still interesting, but it should be noted that we are reporting the best results on the test set, not on the development set.

On Italian, we observe that our best system configuration achieves state-of-the-art results (F1 0.805). The best performing system in the EVALITA shared task [17] reached 0.800 F1 on the development set using an SVM-based classifier with rich linguistic features, while the best score obtained on the test set (0.799 F1) was yielded by a two-layer BiLSTM in a multi-task learning setting. Similar to our best setting, they also use embeddings extracted from social media data, and observe that using sentiment-based lexica does not increase system performance.

On German, the best performing system participating in Germeval [66] achieved 0.768 F1 [48] and was a stacked ensemble system that combined maximum entropy and random forest classifiers and relied on five groups of features. However, the system performance in 10-fold cross-validation using only the training set reached 0.817 F1. Our best configuration on the task test set yields 0.758 F1 with a much simpler architecture, using only Fasttext and no other features except for word embeddings.

As for English, it is more difficult to draw a similar comparison because the dataset we use [62] was originally annotated with three classes (i.e. racism, sexism and none), thus most systems using the same data perform multiclass classification. Besides, they are run using ten-fold cross-validation like in the original paper [62]. One of the few attempts to distinguish between hate and non-hate speech on the same English data is described in [38], where the authors present a classifier combining word-based CNN and character-based CNN. They report 0.734 F1 on the binary task in ten-fold cross-validation. Other works using the same data set for three-class classification report much higher results (0.783 F1 in [31] using CNN, 0.86 F1 in [38] using a multi-layer perceptron). Interestingly, as shown in [38], multi-class classification seems generally easier than the binary one on this specific data set, since sexist and racist tweets present lexical-based discriminating features that are easy to capture.

6.4 Qualitative evaluation

In our experiments we tested more that 1,000 configurations for each language, and it is therefore difficult to manually evaluate and compare the results, since each configuration may make specific mistakes and the distribution of false positives and negatives on the test split would change. In order to gain some insights into the specificity of each language and dataset, however, we focus on the output of the best performing configuration for each language, and we manually check the wrongly classified instances. In most of the cases, it is not possible to assign a category to the mistakes done by the classifier, since the false negative tweets are clearly hateful and the false positive ones are unambiguously non-hateful. These cases are prevalent in all the datasets, so they are independent from the language and also from the dataset size. The opaque mechanisms with which deep learning classifiers assign labels make it difficult to explain why these apparently trivial cases were misclassified, but we plan to exploit information conveyed by attention mechanisms to shed light into this issue [21].

Among the broad mistake categories found across the inspected datasets, there are some cases of implicit abuse. Such messages do not contain abusive words but rather convey their offensive nature through sarcasm, jokes, the usage of negative stereotypes or supposedly objective statements implying some form of offense.

We report few examples of false negatives for the hate speech class below:

4. *It's not about any specific individuals, but about an ideology that will always produce terrorists.*

5. *Molti ancora non vedono, ma quando attraversano un parco, se popolato da immigrati, si tengono stretta la borsa.* (EN: Many do not see it, but when they cross a park populated with immigrants they hold their bag close).

6. *Schau doch Pornos wenn du mehr Redeanteil von Frauen hören willst* (EN: Watch porn if you want to hear more women talk).

We also observe that sentences with a complex syntactic structure, containing for example more than one negation, or questions, are frequent both among the false positives and the false negatives (see Sentence 7, which was wrongly classified as 'Not hate'). The same happens for tweets that contain anaphoric elements that hint at mentions probably present in previous messages, and for tweets which require some form of world knowledge to be understood. In some cases, a link to external media contributed to the hateful meaning of a tweet, as in Sentence 8. However, since we remove urls in the pre-processing step this information was not exploited for classification.

7. *No. You have proven your ignorance here to anyone who isn't as dumb as you. It's there for all to see but you don't know it..*

8. *A quanto pare, il corano si può usare anche per questo. Ma pare non funzioni molto bene..... `http://t.co/DcOSHfmfxK`* (EN: It seems that Quran can be used also for this. But apparently it does not work very well...`http://t.co/DcOSHfmfxK`).

Among false positives, the inspected examples confirm the remarks in [65] concerning the English dataset, and we observe a similar behaviour also for Italian tweets: since these datasets were collected starting from keywords concerning potential hate targets such as women, Roma and Muslims and then extended with not offensive tweets, classifiers tend to associate target mentions to hate speech, even if such messages are not offensive. This phenomenon is less evident on the German data, which indeed was created in a different way, starting from a list of users. Two examples of false positive are reported below. In (9) the message is probably classified as hateful because of the mention of 'Jewish'. In (10) it may depend on the mention of 'migration'.

9. *Fine by me. I had five Jewish friends in college. None ever went to a Synagogue.*

10. *l'immigrazione è un problema x tutti! Ma servono iniziative non comunicati* (EN: Migration is a problem for everybody! But we need initiatives, not press releases).

Finally, we noted few mistakes in the gold standard annotation of the test sets, which were correctly classified by our system.

7 Conclusions

Targeting the hate speech detection task in social media messages, in this paper we have first identified a recurrent neural architecture that is rather stable and well-performing across different languages (i.e., English, German and Italian), and then we have evaluated the contribution of several components that are usually employed in the task, namely the type of embeddings, the use of additional features (text-based or emotion-based), the role of hashtag normalisation and that of emojis. Our comparative evaluation has been carried out on English, Italian and German available Twitter datasets for hate speech detection (annotated as either containing hate speech/offensive language or not). More precisely, in our detailed study we have compared 1,800 possible configurations for English, 1,080 for Italian and 1,224 for German. This allowed us to propose a set of findings, listed in Section 6, that could guide researchers in the design of hate speech detection systems, especially for languages different from English. To be exhaustive, we have also performed an additional set of experiments to investigate to what extent the size of the dataset affects the results.

8 Acknowledgements

Part of this work was funded by the CREEP project[16], a Digital Wellbeing Activity supported by EIT Digital in 2018 and 2019. This research was also supported by the HATEMETER project[17] within the EU Rights, Equality and Citizenship Programme 2014-2020, and by the ANR-19-P3IA-0002 - 3IA Côte d'Azur - Nice - Interdisciplinary Institute for Artificial Intelligence.

[16] `http://creep-project.eu/`

[17] `http://hatemeter.eu/`

References

[1] Sweta Agrawal and Amit Awekar. Deep learning for detecting cyberbullying across multiple social media platforms. In Gabriella Pasi, Benjamin Piwowarski, Leif Azzopardi, and Allan Hanbury, editors, *Advances in Information Retrieval - 40th European Conference on IR Research, ECIR 2018, Grenoble, France, March 26-29, 2018, Proceedings*, volume 10772 of *Lecture Notes in Computer Science*, pages 141–153. Springer, 2018.

[2] Luis Enrique Argota Vega, Jorge Carlos Reyes-Magaña, Helena Gómez-Adorno, and Gemma Bel-Enguix. MineriaUNAM at SemEval-2019 task 5: Detecting hate speech in twitter using multiple features in a combinatorial framework. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 447–452, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[3] Pinar Arslan, Michele Corazza, Elena Cabrio, and Serena Villata. Overwhelmed by Negative Emotions? Maybe You Are Being Cyber-bullied! In *SAC 2019 - The 34th ACM/SIGAPP Symposium On Applied Computing*, Limassol, Cyprus, April 2019.

[4] Xiaoyu Bai, Flavio Merenda, Claudia Zaghi, Tommaso Caselli, and Malvina Nissim. Rug @ EVALITA 2018: Hate speech detection in italian social media. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy.*, 2018.

[5] Xiaoyu Bai, Flavio Merenda, Claudia Zaghi, Tommaso Caselli, and Malvina Nissim. Rug at germeval: Detecting offensive speech in german social media. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 2018.

[6] Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *Language Resources and Evaluation conference, LREC*, Portoroz, Slovenia, May 2016.

[7] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 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, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[8] Valerio Basile and Malvina Nissim. Sentiment analysis on italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107, Atlanta, 2013.

[9] Elisa Bassignana, Valerio Basile, and Viviana Patti. Hurtlex: A multilingual lexicon of words to hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6. CEUR-WS, 2018.

[10] Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. DataStories at SemEval-2017 Task 4: Deep LSTM with Attention for Message-level and Topic-based Sentiment Analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[11] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[12] Cristina Bosco, Felice Dell'Orletta, Fabio Poletto, Manuela Sanguinetti, and Maurizio Tesconi. Overview of the EVALITA 2018 hate speech detection task. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy.*, 2018.

[13] Miguel Ángel Álvarez Carmona, Estefanía Guzmán-Falcón, Manuel Montes-y-Gómez, Hugo Jair Escalante, Luis Villaseñor Pineda, Verónica Reyes-Meza, and Antonio Rico Sulayes. Overview of MEX-A3T at ibereval 2018: Authorship and aggressiveness analysis in mexican spanish tweets. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 74–96, 2018.

[14] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734. Association for Computational Linguistics, 2014.

[15] François Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[16] Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli. A twitter corpus and benchmark resources for german sentiment analysis. In *5th International Workshop on Natural Language Processing for Social Media, Boston, MA, USA*, pages 45–51. Association for Computational Linguistics, 2017.

[17] Andrea Cimino, Lorenzo De Mattei, and Felice Dell'Orletta. Multi-task learning in deep neural networks at EVALITA 2018. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech*

Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy., 2018.

[18] Michele Corazza, Stefano Menini, Pinar Arslan, Rachele Sprugnoli, Elena Cabrio, Sara Tonelli, and Serena Villata. Comparing different supervised approaches to hate speech detection. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy.*, 2018.

[19] Michele Corazza, Stefano Menini, Pinar Arslan, Rachele Sprugnoli, Elena Cabrio, Sara Tonelli, and Serena Villata. Inriafbk at germeval 2018: Identifying offensive tweets using recurrent neural networks. In *GermEval 2018 Workshop*, 2018.

[20] Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. Cross-platform evaluation for italian hate speech detection. In *Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019*, 2019.

[21] Michele Corazza, Stefano Menini, Elena Cabrio, Sara Tonelli, and Serena Villata. Inriafbk drawing attention to offensive language at germeval2019. In *Proceedings of the 15th Conference on Natural Language Processing, KONVENS 2019, Erlangen, Germany, October 9-11, 2019*, 2019.

[22] Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada, May 15-18, 2017.*, pages 512–515, 2017.

[23] Liangjie Hong Brian D Davison April Kontostathis Lynne Edwards Dawei Yin, Zhenzhen Xue. Detection of harassment on web 2.0. In *Proceedings of the Content Analysis in the Web*, pages 1–7, 2009.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[25] Polina Stadnikova Dietrich Klakow Dominik Stammbach, Azin Zahraei. Offensive language detection with neural networks for germeval task 2018. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 2018.

[26] Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. Overview of the task on automatic misogyny identification at ibereval 2018. In *IberEval@SEPLN*, volume 2150 of *CEUR Workshop Proceedings*, pages 214–228. CEUR-WS.org, 2018.

[27] Darja Fišer, Ruihong Huang, Vinodkumar Prabhakaran, Rob Voigt, Zeerak Waseem, and Jacqueline Wernimont. Proceedings of the 2nd workshop on abusive language online (alw2). In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Association for Computational Linguistics, 2018.

[28] Paula Fortuna, Ilaria Bonavita, and Sérgio Nunes. Merging datasets for hate speech classification in italian. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018.*, 2018.

[29] Antigoni-Maria Founta, Despoina Chatzakou, Nicolas Kourtellis, Jeremy Blackburn, Athena Vakali, and Ilias Leontiadis. A unified deep learning architecture for abuse detection. *CoRR*, abs/1802.00385, 2018.

[30] Antigoni-Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Proceedings of the Twelfth International Conference on Web and Social Media, ICWSM 2018, Stanford, California, USA, June 25-28, 2018.*, pages 491–500, 2018.

[31] Björn Gambäck and Utpal Kumar Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90. Association for Computational Linguistics, 2017.

[32] Mario Graff, Sabino Miranda-Jiménez, Eric Sadit Tellez, Daniela Moctezuma, Vladimir Salgado, José Ortiz-Bejar, and Claudia N. Sánchez. INGEOTEC at MEX-A3T: author profiling and aggressiveness analysis in twitter using μtc and evomsa. In *Proceedings of the Third Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018) co-located with 34th Conference of the Spanish Society for Natural Language Processing (SEPLN 2018), Sevilla, Spain, September 18th, 2018.*, pages 128–133, 2018.

[33] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[35] Matthew Honnibal and Ines Montani. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 2017.

[36] Tianran Hu, Han Guo, Hao Sun, Thuy-vy Thi Nguyen, and Jiebo Luo. Spice up your chat: The intentions and sentiment effects of using emojis. In *Proceedings of the Eleventh International Conference on Web and Social Media, ICWSM 2017, Montréal, Québec, Canada*, pages 102–111, 2017.

[37] Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Shrivastava, Nikhil Chakravartula, Manish Gupta, and Vasudeva Varma. FERMI at SemEval-2019 task 5: Using sentence embeddings to identify hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 70–74, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[38] Rohan Kshirsagar, Tyrus Cukuvac, Kathy McKeown, and Susan McGregor. Predictive embeddings for hate speech detection on twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 26–32. Association for Computational Linguistics, 2018.

[39] Gretel Liz De la Peña Sarracén, Reynaldo Gil Pons, Carlos Enrique Muñiz-Cuza, and Paolo Rosso. Hate speech detection using attention-based LSTM. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy.*, 2018.

[40] Younghun Lee, Seunghyun Yoon, and Kyomin Jung. Comparative studies of detecting abusive language on twitter. *CoRR*, abs/1808.10245, 2018.

[41] Ping Liu, Wen Li, and Liang Zou. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[42] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[43] Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. Neural character-based composition models for abuse detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 1–10, Brussels, Belgium, October 2018. Association for Computational Linguistics.

[44] Saif M Mohammad and Peter D Turney. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34. Association for Computational Linguistics, 2010.

[45] Saif M Mohammad and Peter D Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[46] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217 – 250, 2012.

[47] Chikashi Nobata, Joel R. Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, pages 145–153, 2016.

[48] Joaquin Padilla Montani and Peter Schüller. Tuwienkbs at germeval 2018: German abusive tweet detection. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 09 2018.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[50] Juan Manuel Pérez and Franco M. Luque. Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 64–69, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[51] Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. Multiwordnet: developing an aligned multilingual database. In *Proceedings of the First International Conference on Global WordNet*, January 2002.

[52] Marco Polignano and Pierpaolo Basile. Hansel: Italian hate speech detection through ensemble learning and deep neural networks. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy.*, 2018.

[53] Valentino Santucci, Stefania Spina, Alfredo Milani, Giulio Biondi, and Gabriele Di Bari. Detecting hate speech for italian language in social media. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy.*, 2018.

[54] Tatjana Scheffler, Erik Haegert, Santichai Pornavalaia, and Mino Lee Sasse. Feature explorations for hate speech classification. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 2018.

[55] Mike Schuster, Kuldip K. Paliwal, and A. General. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 1997.

[56] Abhishek Singh, Eduardo Blanco, and Wei Jin. Incorporating emoji descriptions improves tweet classification. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[57] Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. *CoRR*, abs/1702.03859, 2017.

[58] Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. 2017.

[59] Jacopo Staiano and Marco Guerini. Depeche mood: a lexicon for emotion analysis from crowd annotated news. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 427–433, 2014.

[60] Dirk von Grunigen, Ralf Grubenmann, Fernando Benites, Pius Von Daniken, and Mark Cieliebak. spmmmp at germeval 2018 shared task: Classification of offensive content in tweets using convolutional neural networks and gated recurrent units. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 2018.

[61] Zeerak Waseem, Wendy Hui Kyong Chung, Dirk Hovy, and Joel Tetreault. Proceedings of the first workshop on abusive language online. In *Proceedings of the First Workshop on Abusive Language Online*. Association for Computational Linguistics, 2017.

[62] Zeerak Waseem and Dirk Hovy. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *SRW@HLT-NAACL*, 2016.

[63] Gregor Wiedeman, Eugen Ruppert, Raghav Jindal, and Chris Biemann. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 2018.

[64] Michael Wiegand, Anastasija Amann, Tatiana Anikina, Aikaterini Azoidou, Anastasia Borisenkov, Kirstin Kolmorgen, Insa Kroger, and Christine Schafer. Saarland university's participation in the germeval task

2018 (udsw) – examining different types of classifiers and features. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 2018.

[65] Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. Detection of Abusive Language: the Problem of Biased Datasets. 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)*, pages 602–608, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[66] Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. Overview of the germeval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*, 2018.

[67] Ellery Wulczyn, Nithum Thain, and Lucas Dixon. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1391–1399, 2017.

[68] Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[69] Robinson D. Zhang, Z. and J. Tepper. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *ESWC 2018: The semantic web Conference Proceedings*, pages 745–760. Springer Verlag, 2018.