**Literature Survey on Multilingual Undesired Speech Detection**

Hate crimes are unfortunately nothing new in society. However, social media and other means of online communication have begun playing a larger role in hate crimes. For instance, suspects in several recent hate-related terror attacks had an extensive social media history of hate-related posts, suggesting that social media contributes to their radicalization. Vast online communication forums, including social media, enable users to express themselves freely, at times, anonymously. While the ability to freely express oneself is a human right that should be cherished, inducing, and spreading hate towards another group is an abuse of this liberty. As such, many online forums such as Facebook, YouTube, and Twitter consider hate speech harmful, and have policies to remove hate speech content. Due to the societal concern and how widespread hate speech is becoming on the Internet, there is strong motivation to study automatic detection of hate speech. By automating its detection, the spread of hateful content can be reduced.

Detecting hate speech is a challenging task, however. First, there are disagreements in how hate speech should be defined. This means that some content can be considered hate speech to some and not to others, based on their respective definitions. We start by covering competing definitions, focusing on the different aspects that contribute to hate speech. We are by no means, nor can we be, comprehensive as new definitions appear regularly. Our aim is simply to illustrate variances highlighting difficulties that arise from such.

Automatic approaches for hate speech detection

Most social media platforms have established user rules that prohibit hate speech; enforcing these rules, however, requires copious manual labour to review every report. Some platforms, such as Facebook, recently increased the number of content moderators. Automatic tools and approaches could accelerate the reviewing process or allocate the human resource to the posts that require close human examination. In this section, we overview automatic approaches for hate speech detection from text.

A basic approach for identifying hate speech is using a keyword-based approach. By using an ontology or dictionary, text that contain potentially hateful keywords are identified. For instance, Hate base maintains a database of derogatory terms for many groups across 95 languages. Such well-maintained resources are valuable, as terminology changes over time. However, as we observed in our study of the definitions of hate speech, simply using a hateful slur is not necessarily enough to constitute hate speech.

Keyword-based approaches are fast and straightforward to understand. However, they have severe limitations. Detecting only racial slurs would result in a highly precise system but with low recall where precision is the percentage of relevant from the set detected and recall is the percent of relevant from within the global population. In other words, a system that relies chiefly on keywords would not identify hateful content that does not use these terms. In contrast, including terms that could but are not always hateful (e.g., “trash”, “swine”, etc.) would create too many false alarms, increasing recall at the expense of precision.

Furthermore, keyword-based approaches cannot identify hate speech that does not have any hateful keywords (e.g., figurative, or nuanced language). Slang such as “build that wall” literally means constructing a physical barrier (wall). However, with the political context, some interpret this is a condemnation of some immigrates in the United States.

### Machine learning classifiers

Machine learning models take samples of labelled text to produce a classifier that is able to detect the hate speech based on labels annotated by content reviewers. Various models were proposed and proved successful in the past. We describe a selection of open-sourced systems presented in the recent research.

#### **Content pre-processing and feature selection.**

To identify or classify user-generated content, text features indicating hate must be extracted. Obvious features are individual words or phrases (n-grams, i.e., sequence of n consecutive words). To improve the matching of features, words can be stemmed to obtain only the root removing morphological differences. Metaphor processing, e.g., Neuman, et. al., likewise, can extract features.

The bag-of-words assumption is commonly used in text categorization. Under this assumption, a post is represented simply as a set of words or n-grams without any ordering. This assumption certainly omits an important aspect of languages but nevertheless proved powerful in numerous tasks. In this setting, there are various ways to assign weights to the terms that may be more important, such as TF-IDF.For a general information retrieval review, see.

Besides distributional features, word embeddings, i.e., assigning a vector to a word, such as word2vec, are common when applying deep learning methods in natural language processing and text mining. Some deep learning architectures, such as recurrent and transformer neural networks, challenge the bag-of-words assumption by modelling the ordering of the words by processing over a sequence of word embeddings.

#### **Hate speech detection approaches and baselines.**

**Naïve Bayes, Support Vector Machine and Logistic Regression**. These models are commonly used in text categorization. Naïve Bayes models label probabilities directly with the assumption that the features do not interact with one another. Support Vector Machines (SVM) and Logistic Regression are linear classifiers that predict classes based on a combination of scores for each feature. Open-source implementations of these models exist, for instance in the well-known Python machine learning package sci-kit learn.

**Davidson, et al**. Davidson, et al. proposed a state-of-the-art feature-based classification model that incorporates distributional TF-IDF features, part-of-speech tags, and other linguistic features using support vector machines. The incorporation of these linguistic features helps identify hate speech by distinguishing between different usages of the terms, but still suffers from some subtleties, such as when typically, offensive terms are used in a positive sense.

**Neural Ensemble**. Zimmerman, et al. propose an ensemble approach, which combines the decisions of ten convolutional neural networks with different weight initializations. Their network structure is like the one proposed by, with convolutions of length 3 pooled over the entire document length. The results of each model are combined by averaging the scores, akin to.

**FastText** . Fast Text is an efficient classification model proposed by researchers in Facebook. The model produces embeddings of character n-grams and provides predictions of the example based on the embeddings. Over time, this model has become a strong baseline for many texts categorization tasks.

**BERT** . BERT is a recent transformer-based pre-trained contextualized embedding model extendable to a classification model with an additional output layer. It achieves state-of-the-art performance in text classification, question answering, and language inference without substantial task-specific modifications. When we experiment with BERT, we add a linear layer atop the classification token , and test all suggested tuning hyperparameters.

**C-GRU** . C-GRU, a Convolution-GRU Based Deep Neural Network proposed by Zhang, et al., combines convolutional neural networks (CNN) and gated recurrent networks (GRU) to detect hate speech on Twitter. They conduct several evaluations on publicly available Twitter datasets demonstrating their ability to capture word sequence and order in short text. Note, in the HatebaseTwitter dataset, they treat both Hate and Offensive as Hate resulting in binary label instead of its original multi-class label. In our evaluation, we use the original multi-class labels where different model evaluation results are expected.