LING 573 Deliverable #3 Report

Mohamed Elkamhawy
University of Washington
mohame@uw.edu

Karl Haraldsson
University of Washington
kharalds@uw.edu

Alex Maris
University of Washington
alexmar@uw.edu

Nora Miao
University of Washington
norah98@uw.edu

Abstract

The paper describes our refined affect recognition system developed for SemEval-2019 Task 5, which focuses on identifying hate speech in tweets targeted at immigrants and women. The task is divided into two subtasks: the primary task involves hate speech detection in English tweets, while the adaptation task addresses Spanish tweets. We employed a binary classification approach, in which each given tweet is classified as either hate speech or non-hate speech. To build our system, we trained a Word2Vec model to generate word embeddings and then used them as input features for a Support Vector Machine (SVM) classification algorithm. As part of the enhancements for D3, we experimented with different preprocessing techniques, applied additional lexical resources, and fine-tuned hyperparameters on the development data set. Overall, our current system achieved a comparable level of performance as our initial system, with slight improvements in accuracy, precision, recall, and F1-score.

1 Introduction

In recent years, the proliferation of hate speech on social media platforms has become a critical societal issue receiving unprecedented attention. Not only does hate speech have detrimental effects on individuals' psychological well-being, but it also poses significant safety risks to our society at large (Saha et al., 2019). Hate speech can lead to devastating implications. For example, it can perpetuate and reinforce racism, biases, stereotypes, and discrimination against certain individuals, eliciting emotional distress. Moreover, studies have shown a link between hate speech and increased acts of violence and hate crimes against members of minority groups (Relia et al.). As online social networks continue to reach broader audiences and serve as a primary means of communication, the need for more effective hate speech detection algorithms has become particularly evident. Recent research in automatic hate speech detection using NLP techniques has shown some promising results[1]. To address the spread of hateful narratives online, we built an affect recognition system that aims to identify hate speech on Twitter against two specific minority groups: immigrants and women.

The rest of the paper is organized as follows. Section 2 provides an overview of the primary and adaptation tasks that we aim to address, along with a description of the dataset and the evaluation procedure. Section 3 presents the overall system architecture and briefly discusses the major design decisions. Next, Section 4 details the major subcomponents of the system. Section 5 outlines the key findings obtained through the evaluation of the system, while Section 6 interprets the findings and analyzes the results in detail. Additionally, this section includes error analysis and assessment of each component. Furthermore, in Section 7, we discuss the limitations and ethical considerations regarding the implementation and application of our system. Finally, Section 8 concludes the paper by summarizing the main findings and potential next steps for improvements.

2 Task Description

The chosen primary task is detecting hate speech in tweets, where the hate speech is against either women or immigrants, as described in SemEval-2019 Task 5 (Basile et al., 2019). Specifically, it is a binary classification task targeted at determining attitude–here whether a given tweet contains hate speech or not. The genre for this task is tweets and the modality is text. The target of this task is aspect-specific, and the language is English for the primary task. The adaptation task will be hate

[1] See, e.g., Asogwa et al. (2022), Kotarcic et al. (2023), and Schmidt and Wiegand (2017).

speech identification in Spanish tweets, also described in Basile et al. (2019). The only difference between the primary task and the adaptation task is the language, while all the other dimensions remain unchanged.

The data for the shared task was collected from July to September 2018 for the immigrant-targeted tweets (Basile et al., 2019). The data for the women-targeted tweets was collected from July to November 2017 (Fersini et al., 2018). The English language dataset contains 13,000 tweets, 9,000 of which are in the training set, 1,000 in the development set, and 3,000 in the test set. Of the 13,000 tweets, 7,530 are annotated for the negative class, and 5,470 are annotated for the positive class. The Spanish language dataset that will be used for the adaptation task consists of 6,600 tweets, 4,500 of which are for training, 500 for development, and 1,600 for testing. Of the 6,600 tweets, 3,861 are annotated for the negative class, and 2,739 are annotated for the positive class. The annotations were collected using the *Figure Eight* (F8) platform, where each tweet was annotated by at least three contributors, and then a relative majority label was assigned. Expert annotators were also utilized, such that the final label of a given tweet was determined by the majority label of the F8 annotation and two independent expert annotators (Basile et al., 2019). The evaluation is calculated using accuracy, precision, recall, and macro-averaged F1-scores to maintain class-size independence since the hate speech and non-hate speech class sizes are relatively balanced (Basile et al., 2019).

Table 1: Dataset Distribution

Language	Hateful	Non-Hateful	Total
English	3783	5217	9000
Spanish	1857	2643	4500
Total Train	**5160**	**8660**	**13820**
English	427	573	1000
Spanish	222	278	500
Total Dev	**649**	**851**	**1500**
English	1260	1740	3000
Spanish	660	940	1600
Total Test	**1920**	**2680**	**4600**
Total	**8209**	**11391**	**19600**

The dataset can be requested using this form, and the shared task is detailed at this page as well as in Basile et al. (2019).

3 System Overview

We developed an end-to-end system using `Python3` for the primary task. Our development workflow consists of six stages: data preprocessing, feature extraction, training, inference, evaluation, and deployment. To improve our model performance, we made several revisions specifically targeting three areas: data preprocessing, feature engineering, and hyperparameter tuning. Similar to our initial system, in the data preprocessing stage, we cleaned and prepared data using the Natural Language Toolkit (NLTK) library[2], and performed standard text processing such as removing special characters, tokenization, and lemmatization. However, in our updated system, we implemented advanced data preprocessing techniques, such as handling negation, processing emojis, and correcting misspellings, to improve the quality of our data.

In the feature extraction stage, we employed a Bag-of-Words approach for our baseline model. For our initially proposed models, we generated word embeddings using a pre-trained word2vec model from the Gensim library, which transformed text data into numerical representations. Additionally, we used the `Empath` package in conjunction with `Word2Vec` to generate lexical features. For our refined model, we added additional lexical features including n-grams and sentiment scores to our model to preserve contextual information and capture the relationships between words. These features can be selectively toggled on or off using our configuration files. This setup allowed us to explore various combinations of features and select the approach with the highest evaluation score.

Once the features were prepared, we used a support vector machine (or SVM, a supervised learning model) as our classifier and trained the model using labeled data in the training set. During training, we conducted hyperparameter fine-tuning to identify the optimal configuration of our model. The inference stage involves using the trained SVM model to predict the presence of hate speech in tweets from the development dataset. In the evaluation stage, we assessed the performance of our model using the evaluation metrics prescribed in Basile et al. (2019), including standard metrics

[2] Natural Language Toolkit (NLTK) is a popular open-source library for natural language processing (NLP) in Python. It provides a suite of tools for text-processing tasks such as classification, tokenization, stemming, tagging, and parsing. More information can be found at `https://www.nltk.org/`.

such as accuracy, precision, recall, and F1 score. Lastly, we integrated all the components into a comprehensive, functional system that can be run in the patas environment.

4 Approach

The system implements a series of binary classifiers for detecting hate speech in tweets. Our approach comprises four primary components: (1) data preprocessing; (2) feature extraction; (3) model training; (4) model inference; and (5) evaluation.

Our system can implement various modeling approaches, but we settled on four models. We created a baseline model, which relies on a "bag of words" representation of the features and a binary support vector machine (SVM) classifier (the model hereafter known as BOW-SVM). We propose four additional models. The first proposed model also uses a binary SVM classifier but relies on embeddings derived using `word2vec` (hereafter W2V-SVM). Our second proposed model uses additional lexical features derived using the `Empath` package (hereafter W2V-EMP-SVM). D3 attempts to improve on both of those models, which were originally a part of D2. Our third proposed model includes all of the features present in W2V-EMP-SVM, as well as n-gram counts for each instance (hereafter W2V-EMP-NG-SVM). Our fourth and final model agglomerates all of our proposed improvements, incorporating adjustments to text preprocessing, word2vec embeddings, lexical features from empath, n-gram vectors, and sentiment scores (hereafter KITCHNSNK-SVM).

4.1 Data preprocessing

The Shared Task data is provided in raw CSV format, already split into train, dev, and test sets. Each row in the CSV files represents a Tweet. The CSV includes a column with a numerical unique identifier (id), a column with the text of the tweet (text), and three target variable columns (HS, TR, and AG). Column HS is the target for Subtask A, and therefore this system. In that column, the value 1 indicates that hate speech is present. The value 0 indicates that it is not.

This system ingests that data, then performs several text preprocessing operations to ready the data for modeling. First, the added system removes hashtags and splits the text of the hashtags into distinct tokens using capitalization or underscore. Our model provides two options for handling mentions: they can either be removed altogether or processed in the same manner as hashtags. Next, we converted emojis in the tweets into words using `emoji.demojize()`. We also used regular expressions (regex) to capture some of the misspelled or censored curse words and converted them back to their corresponding original forms.

In addition to URLs and punctuations, our updated model also detected and removed HTML and Unicode characters using `str.encode()` as well as regex to further reduce noise in the data. Our system then tokenizes the text using a specialized tweet tokenizer from NLTK, which takes into account tweet-specific elements including hashtags, mentions, and emoticons.

To handle negation, we first expanded contractions using the `contractions` package. We defined a list of negation words such as "no", "not", and "never" and added a "NEG_" affix to the word following any negation word in the predefined list. Then, it lemmatizes these tokens using the WordNet lemmatizer[3] available from NLTK. Finally, our model considered English stopwords[4] as defined in NLTK.

Our revised system provided us with the flexibility to experiment with different combinations of preprocessing techniques by including boolean variables that can enable or disable certain preprocessing steps. This allowed us to quickly and easily evaluate the effects of a specific preprocessing step on our model performance. To minimize information loss, our default setup retained stopwords, numbers, and mentions.

The preprocessed steps mentioned above are applied consistently for every modeling approach, including the baseline BOW-SVM.

4.2 Feature extraction

Our system applies three distinct feature engineering approaches: one for the baseline model, another for the two models we developed specifically for this task in D2, and yet another for our newest model developed for D3. For the baseline

[3] More information about the WordNet lemmatizer can be found at https://www.nltk.org/api/nltk.stem.wordnet.html#nltk.stem.WordNetLemmatizer

[4] Stopwords are a set of commonly used words in a language that does not add any additional information or meaning to a sentence. They are generally filtered out during the data-preprocessing phase of many NLP tasks. The NLTK corpus provides a list of stopwords for multiple languages. More information can be found at https://www.nltk.org/book/ch02.html#stopwords_index_term.

model, we implement a simple BOW mechanism via scikit-learn's CountVectorizer() object.

For the two proposed models from D2 (W2V-SVM and W2V-EMP-SVM), we rely on word2vec (continuous bag of words). W2V-EMP-SVM includes the additional step of concatenating on the word2vec representation matrix a vector of lexical information for each instance, using the Empath package. Empath is a tool that is used to score text across 194 pre-identified categories that have been determined from topics and emotions, using dependency relationships from the ConceptNet knowledge base, and compiled seed terms that correspond to each concept. [5]. Our latest models, W2V-EMP-NG-SVM and KITCHNSNK-SVM, adds on to W2V-EMP-SVM. W2V-EMP-NG-SVM includes a vector representation of n-gram (unigrams, bigrams, trigrams, and four-grams) counts. KITCHNSNK-SVM includes all of the features from W2V-EMP-NG-SVM as well as a sentiment score of the text from the sentiment analysis package, VADER (vaderSentiment).

4.3 Model training

All of our models use an SVM classifier. We apply to our training set a basic grid search approach to identify the best hyperparameters per the development set.[6] Each model is scored based on its macro-averaged f1 score and the best model is saved and used for evaluation.

4.4 Model inference

Our system accepts at the time of inference both models trained during the same run and models saved from a prior run. In either case, the system applies the model to the preprocessed and engineered tweet text and outputs predictions for each dev or test instance.

4.5 Evaluation

The system applies the evaluation measures for Subtask A as described in Basile et al. (2019). Those measures are accuracy, precision, recall, and F1-score. Submissions are ranked by macro-averaged F1-score (Basile et al., 2019). As prescribed by the Shared Task, we calculate each using the precision_recall_fscore_support() method from sklearn.metrics. Those metrics are then written to a results file.

5 Results

This section describes the results of our initial system (D2), the results of our enhanced system (D3), and the differences between the two systems.

5.1 Initial System Results

Our system's preliminary results (D2) on the development dataset for the shared task are presented in Table 2.

Table 2: SemEval-2019 Task 5 (Subtask A) Preliminary Results (D2)

Model	Acc	Prec	Rec	F1-Macro
W2V-SVM	0.67	0.62	0.71	0.60
W2V-EMP-SVM	0.70	0.66	0.72	0.65
BOW-SVM *Baseline*	**0.73**	**0.72**	**0.73**	**0.72**
Basile SVC *Baseline*	-	-	-	0.45

Basile et al. (2019) provides a baseline of a support vector classifier (SVC) that uses default hyperparameters and a tf-idf approach. When evaluated on the test set, the F1-macro score for that model is 0.45. We've included it here for reference, but it is worth noting that our D2 results are based on performance against true labels in the development set, not the test set. We, therefore, establish a baseline model of our own as well. It implements a bag-of-words approach with a cross-validated support vector machine classifier (BOW-SVM), which achieved an accuracy of 0.73, a precision of 0.72, a recall of 0.73, and an F1-Macro score of 0.72.

Our SVM models that rely on word2vec representations had lower scores for each. The W2V-SVM model achieved an accuracy of 0.67, precision of 0.62, recall of 0.71, and an F1-Macro score of 0.60. Similarly, the W2V-EMP-SVM model achieved an accuracy of 0.70, precision of 0.66, recall of 0.72, and an F1-macro score of 0.65. These models have lower F1-macro scores than our own baseline BOW-SVM model, but they were more than 0.15 points over the baseline in Basile et al. (2019).

5.2 Enhanced System Results

Our system's enhanced results (D3) on the development dataset for the shared task are presented in

[5] See Fast et al. (2016) for more details

[6] Searching across all combinations of C {0.1, 1, 10} and kernel {linear, rbf, sigmoid}, the parameters with the highest F1-macro score were a C of 0.1 and the linear kernel

Table 3. The table compares the accuracy, precision, recall, and F1-Macro scores of six different models: W2V-SVM, W2V-EMP-SVM, W2V-EMP-NG-SVM, KITCHNSNK-SVM, and two versions of the BOW-SVM model, the D3 baseline and the D2 baseline. The results show that the BOW-SVM D3 baseline model performs the best with an accuracy and recall of 0.75 and precision and F1-Macro score of 0.74.

Table 3: SemEval-2019 Task 5 (Subtask A) Enhanced Results (D3)

Model	Acc	Prec	Rec	F1-Macro
W2V-SVM (alpha)	0.69	0.66	0.70	0.65
W2V-EMP-SVM (beta)	0.69	0.65	0.70	0.65
W2V-EMP-NG-SVM (gamma)	0.74	0.72	0.73	0.72
KITCHNSNK-SVM (delta)	0.73	0.72	0.73	0.72
BOW-SVM *D3 Baseline*	**0.75**	**0.74**	**0.75**	**0.74**
BOW-SVM *D2 Baseline*	0.73	0.72	0.73	0.72
Basile SVC *Baseline*	-	-	-	0.45

The enhanced system applies two changes to the D2 model (W2V-SVM, W2V-EMP-SVM, and BOW-SVM): adjusted text preprocessing and hyperparameter tuning on the development data set. As a result, W2V-SVM attained equivalent performance metrics to W2V-EMP-SVM, which did not change in a meaningful way. Per Table 4 W2V-SVM had the highest magnitude increase in accuracy (+0.02), precision (+0.04), and F1-Macro (+0.05) compared to D2. W2V-SVM saw slight decreases in accuracy (-0.01), precision (-0.01), and recall (-0.02), but none of them were substantial enough to materially change its F1-Macro score, which remained at 0.65. With these improvements, the evaluation metrics for W2V-SVM and W2V-EMP-SVM still lag behind the D2 Baseline, BOW-SVM.

The enhanced system also tests two modifications to feature extraction: Two new models, W2V-EMP-NG-SVM and KITCHNSNK-SVM, each achieves F1-macro scores of 0.72. They perform comparably to the D2 Baseline BOW-SVM F1-macro of 0.72 and only 0.02 points lower than the new D3 Baseline BOW-SVM.

Table 4: SemEval-2019 Task 5 (Subtask A) D3 Change vs. D2

Model	Acc	Prec	Rec	F1-Macro
W2V-SVM	**+0.02**	**+0.04**	-0.01	**+0.05**
W2V-EMP-SVM	-0.01	-0.01	-0.02	+0.00
BOW-SVM *D3 Baseline*	**+0.02**	+0.02	**+0.02**	+0.02

6 Discussion

Error analysis was conducted on the pre-processed dev tweets to identify if there were any attributes in common among the false negatives and false positives for each of the approaches, and across approaches, in the overlapping mislabeled documents as well. The dev data contains 573 tweets labeled for the negative class and 427 tweets labeled for the positive class.

Table 5: Errors on Development Set

Model	TP	TN	FP	FN
W2V-SVM	185	504	69	242
W2V-EMP-SVM	185	501	72	242
W2V-EMP-NG-SVM	258	478	95	169
KITCHSNK-SVM	252	482	91	175
BOV-SVM *D3 Baseline*	279	475	98	148

Similar to D2, the W2V approaches favor false negative errors to false positives. Notably, introducing the n-gram feature improved the rate of both types of errors. Of the false positive errors, of all the combinations of approaches, the two approaches with the least number of errors in common are W2V-SVM (alpha) and KITCHNSNK-SVM (delta) with 37 false positives in common (53.6% and 40.7% of the false positives for each approach respectively). The two approaches with the most number of errors in common are W2V-EMP-NG-SVM (gamma) and KITCHNSNK-SVM (delta) with 72 false positives in common (75.8% and 79.1% of the false positives for each approach respectively). This is likely due to the few features that KITCHNSNK-SVM contributes in addition to W2V-EMP-NG-SVM. Of the false negative errors, it is less clear which combination of approaches has the least number of errors in common (when considering both quantity and percentage). The two approaches with the most number of errors in common are W2V-SVM (alpha) and W2V-EMP-SVM (beta) with 219 false negatives in common (90.5% of the false negatives for both approaches).

Table 6: Classwise Model Performance on Hateful Tweets Only

Model	Precision	Recall	F1
BOW-SVM	**0.74**	**0.65**	**0.69**
W2V-SVM	0.73	0.43	0.54
W2V-EMP-SVM	0.72	0.43	0.54
W2V-EMP-NG-SVM	0.73	0.60	0.66
KITCHN-SINK-SVM	0.73	0.59	0.65

When we inspect the classwise precision and recall of the BOW-SVM model and the W2V-SVM model, we see that the BOW-SVM model recalls 65% of hateful tweets compared to just 43% for the W2V-SVM and W2V-EMP-SVM model. One reason for this may be that information present in a simple BOW approach is lost when embeddings are applied. That is, applying and averaging embeddings may have eliminated important signals from specific tokens. W2V-EMP-NG-SVM and KITCHNSNK-SVM, however, achieve recall scores of 60% and 59%, respectively. The primary difference between those two models and the initial D2 models above is that the feature vectors include counts of n-grams in the tweets. Thus, some of the information lost when using embeddings may have been recovered by adding n-gram counts.

As observed in D2, specific text features are observed frequently in mislabeled tweets, and serve as examples to illustrate this loss of information. For the W2V-EMP-SVM approach, out of 18 tweets that contain 'illegals', the model classified 11 (61.1%) as hate speech, while the true count is 16 (88.9%). It is likely that 'illegals' was not sufficiently used as a potential indicator of hate speech. The following tweet is clearly hate speech, but the model classified it incorrectly:

> **Original**: A Valedictorian DACA Illegal(is there any other kind?)took time out of her busy brain surgeon scheduleto SMUGGLE s'more4.0 Med Students &amp; Quantum Physicists ILLEGALS into the US.She musta heard that we were gonna use a merit-based immigration system!https://t.co/BpPfv8Har5
>
> **Pre-processed**: a valedictorian daca illegal is there any other kind took time out of her busy brain surgeon scheduleto smuggle s more 4 0 med student quantum physicist illegals into the u she musta heard that we were going to use a merit based immigration system

However, when introducing n-grams as features, the recall of the model is improved as more information is able to be captured in bi-, tri-, or 4-grams than unigrams. This improves the model's ability to achieve better correlations between features and hate speech classification. For example, the W2V-EMP-NG-SVM model did classify 16 (88.9%) of tweets containing 'illegals' as hate speech, which matches the true count (although there are 2 false positives among the 16, and thus 2 false negatives). The example tweet above was classified correctly by this model. Regarding empath, each of W2V-EMP-SVM, W2V-EMP-NG-SVM, and KITCHNSNK-SVM utilized the feature, but the values below reflect the W2V-EMP-SVM approach. When calculating the average empath scores over all lexical categories, the category of swearing terms had the largest difference in average scores between true positives and false positives, where the true positives averaged 0.0395 and the false positives averaged 0.0324. The category of childish had the largest difference in average scores between true negatives and false negatives, where the true negatives averaged 0.0046 and the false negatives averaged 0.0141. For reference, the largest scoring categories for each of true positives, false positives, true negatives, and false negatives respectively are swearing terms (0.0395), swearing terms (0.0324), crime (0.0101), and childish (0.0141), which are the same categories as for D2.

Lastly, the sentiment scores were averaged for each document that was labeled by the KITCHNSNK-SVM (delta) approach, for each type of error.

Table 7: Sentiment Analysis Results for Development Set

Sentiment Score	TP	TN	FP	FN
Avg. Neg.	0.228	0.133	0.231	0.135
Avg. Neu.	0.708	0.779	0.687	0.767
Avg. Pos.	0.064	0.089	0.082	0.097
Avg. Comp.	**-0.435**	**-0.094**	**-0.47**	**-0.066**

Based on the notable differences in the compound scores of TN and FN compared to TP and FP, the model seemingly tended to classify tweets that had a low compound score (and a higher negative score) as positive for hate speech, and a higher compound score as not hate speech, even though in practice, a tweet with a low compound sentiment score is not necessarily hate speech (as evidenced by the low-scoring false positives).

7 Ethical Considerations

Our current model has several limitations. Firstly, hate speech can change over time as language and culture evolve. As previously mentioned, the training data we used was collected in 2018. Therefore, our model may not be able to adequately capture the recent forms of hate speech as it lacks contextual understanding of the current world. Moreover, our existing system may not be particularly adept at identifying implicit hate speech, which is often expressed through sarcasm, metaphor, or other subtle forms of expression. Detecting implicit hate speech can be a challenging task as it requires context and relies less on lexical cues. As a result, our system needs to be constantly updated and fine-tuned in order to adapt to the nuanced and ever-changing patterns of hate speech.

The implementation and application of hate speech detection models against immigrants and women raise several ethical challenges that need to be considered. Firstly, the data was collected using a keyword-driven approach (Basile et al., 2019), which may lead to a biased and unrepresentative dataset. Furthermore, privacy is another ethical concern that is worth noting. The raw tweets from the dataset contain mentions and other information, which could potentially be used to re-identify the user who made the post and preferred to stay anonymous. Hate speech detection models may also amplify biases towards minority groups. For instance, certain terms may be considered offensive when used by a certain group of people but not necessarily by others. Hate speech depends on social and cultural context. To mitigate the potential harms associated with the deployment of hate speech detection models, it is imperative to build a system that is transparent, responsible, and accurate.

8 Conclusion

Our improved baseline BOW-SVM model achieved the best overall performance among all the models that we evaluated. Although our refined models did not outperform that model, they still achieved reasonable results. Indeed, each improved upon the Basile et al. (2019) baseline SVC model. Furthermore, the models that incorporated n-grams matched the F1-macro score of the prior system's best model. These results indicate that our efforts to improve the system have been successful. However, there is still room for further enhancements and future work should focus on refining existing preprocessing and feature extraction techniques, exploring ensemble methods, and deploying transformer-based models such as BERT, roBERTa, or DistilBERT.

References

Doris Chinedu Asogwa, Chiamaka Ijeoma Chukwuneke, CC Ngene, and GN Anigbogu. 2022. Hate speech classification using SVM and Naive BAYES. *arXiv preprint arXiv:2204.07057*.

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, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Ethan Fast, Binbin Chen, and Michael S Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 4647–4657.

Elisabetta Fersini, Paolo Rosso, and Maria Anzovino. 2018. Overview of the task on automatic misogyny identification at ibereval 2018. In *Proceedings of the 3rd Workshop on Evaluation of Human Language Technologies for Iberian Languages (IberEval 2018)*, volume 2150, pages 214–228.

Ana Kotarcic, Dominik Hangartner, Fabrizio Gilardi, Selina Kurer, and Karsten Donnay. 2023. Human-in-the-loop hate speech classification in a multilingual context.

Kunal Relia, Zhengyi Li, Stephanie H Cook, and Rumi Chunara. Race, ethnicity and national origin-based discrimination in social media and hate crimes across 100 us cities. In *Proceedings of the International AAAI Conference on Web and Social Media*.

Koustuv Saha, Eshwar Chandrasekharan, and Munmun De Choudhury. 2019. Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of the 10th ACM conference on web science*, pages 255–264.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.