LING 573: Affect Classification Report

Ben Cote
University of Washington
bpc23@uw.edu

Madhav Kashyap
University of Washington
madhavmk@uw.edu

Lindsay Skinner
University of Washington
skinnel@uw.edu

Keaton Strawn
University of Washington
kstrawn@uw.edu

Allan Tsai
University of Washington
yltsai@uw.edu

Abstract

This paper describes our initial system for Task 5 of SemEval-2019: HatEval: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. The main purpose of this shared task is to conduct hate speech detection on tweets, which mainly targets two specific groups of people, namely immigrants and women. To address this task, we developed a system that utilized word embeddings and the random forest classification method. We present the results obtained for both Subtask A and Subtask B for English tweets. Our initial system achieved an F1-score of 0.299 for English tweets in Task A.

1 Introduction

With the growing popularity of social media, microblogging platforms like Twitter provide a medium for people to communicate with each other using short texts. While these platforms can be used to share users' memorable personal events or constructive opinions on certain topics, some people may use them to propagate their hatred against an individual, a group, or a race. Hence, it becomes crucial to come up with automated and computational methods to identify hate speech on social media platforms.

While there is an increasing number of research dedicated to combatting the issue of hate speech on social media platforms, there are still a lot of problems that remain unsolved. This is mainly caused by the fact that there is no widespread agreement on what constitutes hate speech. One definition considers hate speech as "language that is used to express hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group" (Davidson et al., 2017). Another describes hate speech tweets as those containing racist or sexist comments (Waseem, 2016; Waseem et al., 2017).

The shared task 5 of the SemEval-2019 workshop (Basile et al., 2019) defines two subtasks for detecting hate speech against immigrants and women on Twitter. Both subtasks contain tweets in English and Spanish. In Subtask A, the system has to predict whether a tweet, with a given target, is hateful or not. In Subtask B, the system has to determine whether a given hateful tweet is aggressive or not and whether it targets an individual or a group.

The rest of the paper is structured as follows. After this introductory section, we provide a brief description of the subtasks in Section 2. In Section 3, we give an overview of our system. In Section 4, we provide a detailed approach to developing our system. In Section 5, we show the results of our initial system. Section 6 contains the discussion and summary. We conclude and describe future work in Section 8.

2 Task Description

Drawing from the Semeval-2019 shared task 5 [1], this project develops a binary affect classification system aimed at Multilingual detection of hate speech against immigrants and women in twitter (Basile et al., 2019). The data is text-based [2], and comes from shared task 5 in the form of pre-scraped and pre-labeled tweets, classified with either a 1 or a 0 on three metrics: Hate Speech, Personal Target, and Aggression. Hate Speech (0 - hate speech not present, 1 - hate speech present) refers to whether or not the target tweet disparages a person or a group on the basis of some characteristic such as race, color, ethnicity, gender, sexual orientation,

[1] The shared task CodaLab page can be found here or at this URL: https://competitions.codalab.org/competitions/19935#learn_the_details

[2] The data is not available on the shared task CodaLab page, but it is available on GitHub here or with this URL: https://github.com/cicl2018/HateEvalTeam

nationality, religion, or something else. Personal Target (0 - general group, 1 - specific individual) refers to tweets that do have hate speech, and reflects whether the hateful tweet is targeting a group of people generally, or whether there is a specific person being attacked. Aggression (0 - no aggression, 1 - aggression) refers to tweets that do have hate speech, and reflects whether the tweeter is aggressive or not. For our primary task, our system will only be using the English data to classify English tweets on the three binary metrics listed above. For our adaptation task, we will adjust our system in order to classify tweets in both English and Spanish on the three binary metrics listed above. In both cases, evaluation will be done by comparing our system's classification of each tweet with the pre-labeled classifications. Precision and recall of the classification will be amalgamated into a macro-averaged F1-score. Additionally, we will calculate an Exact Match Ratio (EMR), the metric used by the shared task to rank submissions from most- to least- effective at hate speech detection.

3 System Overview

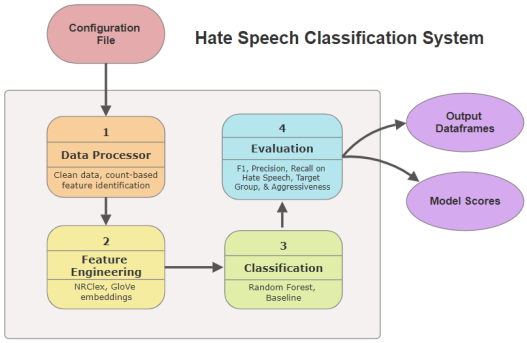

Figure 1: A graphical depiction of our Hate Speech Classification system

Figure 1 shows the flow of our system, from input configuration through the Hate Speech system, outputting both a dataframe of features and classification predictions, and an evaluation of those predictions using precision, recall, accuracy, and F1 scores. In this system, once the input tweets have been processed, each tweet is treated as the input for the Hate Speech Classification System. Features are selected from these tweets to use as the basis for classification, the model is trained, and then the development data is classified using the trained model according to the methods specified within the configuration file. The classification predictions are then used to evaluate the model's performance.

4 Approach

Our approach has four major components: (1) Data Processing; (2) Feature Engineering; (3) Classification; and (4) Evaluation. This document reflects the state of the system as of its current implementation (D2). Each component is detailed below.

4.1 Data Processing

The DataProcessor is used to pull and clean the raw data. The data is read in from .csv files that are either saved locally, or downloaded from GitHub. The raw data comes from two separate .csv files, separated into training and validation data. Each file has five columns: id, text, HS, TR, AG. 'id' contains the unique identification numbers for each tweet; 'text' contains the raw text of the tweet (this includes URLs, hashtags, emojis, references to twitter accounts, slang and misspellings, etc.); HS is a binary (0 or 1) tag that indicates whether (1) or not (0) the tweet is classified as hate speech; TR is a binary tag that indicates whether the tweet is targeted at an individual (1) or a general group (0); AG is a binary tag that indicates whether (1) or not (0) the tweet is aggressive.

The cleaning process separates URLs and hashtags from the text and replaces emojis with English descriptions. It also calculates and adds a feature indicating what percentage of the text is capitalized, before lower-casing the cleaned text. Similarly, the cleaning process generates counts for special punctuation symbols, before removing all punctuation from the cleaned text. Finally, Twitter ID references are stored in a separate feature and replaced with the string 'user' in the cleaned text.

4.2 Feature Engineering

The FeatureEngineering class takes in the output of the data cleaning process and creates an expanded dataframe with additional columns for each new feature that is generated. This class contains two main methods. First is the fit_transform method, which intakes the training data in order to train the feature-generating helper methods, wherever such training is required. The second main method is the transform method, which uses the trained helper methods from an earlier call to fit_transform in order to transform the validation and test datasets to include the complete list of generated features for

each dataset. The outputs of the fit_transform and transform methods will be transformed dataframes that contain additional fields for each of the features mentioned below.

4.2.1 NRC Counts

This feature involves the binary classification of words across eight emotional dimensions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust), then a positive dimension and a negative dimension. Raw counts are normalized across tweets.

4.2.2 GloVe Embeddings

This feature calls on pretrained GloVe style embeddings from GloVe Twitter to create word embedding representations for the cleaned_text. The embedding dimension is determined by the version of glove.twitter that is specified by the embedding_filepath (in the case of D2 they are d=25). Embeddings are generated for each word in the text and then aggregated (using the component-wise mean) to generate a proxy for a sentence-level embedding that is stored in Aggregate_embeddings.

4.3 Classification

The ClassificationModel uses the engineered features to train a model to predict the target class. This is accomplished by first training the model on a training dataset that includes gold-standard labels for the classification task of interest. Once a model is trained the predict method can be used to make predictions with that model over a new dataset that contains all the features (except for the gold-standard targets) used during model training.

There are two classification methods implemented in the current system. The first is a baseline classification that predicts that no tweet contains hatespeech, as this is the most frequently seen label in the training data (HS=0). This is a simple implementation intended to provide a base on which we can compare future classification methods that we implement. The second is a random forest method, using features as split-points to separate and classify the data.

4.4 Evaluation

The Evaluator takes the output of the classification and runs an analysis to evaluate the model's performance on the affect classification task. For each metric (HS, TR, AG), the evaluator calculates the precision, recall, accuracy, and F1 scores achieved by the model. This class is an adaptation of the Evaluation script provided by the SemEval 2019 Task 5 team to ensure the same evaluation techniques were used by all teams participating in the shared task. It was then modified to be a class within our model instead of a separate script.

Evaluation is broken into two separate tasks. Task A analyzes at the system's predictions for the binary Hate Speech classification, while Task B analyzes the system's predictions for the binary classifications of Hate Speech, Personal Target, and Aggression, as well as a Macro averaging of all the categories.

5 Results

Table 1: Initial System (D2) Evaluation Scores

	Performance Metrics			
	F1	Precision	Recall	Accuracy
Baseline Scores				
Task A	0.364	0.787	0.500	0.573
Task B (Macro)	0.415	-	-	-
Task B (HS)	0.364	0.787	0.500	0.573
Task B (TR)	0.439	0.891	0.500	0.781
Task B (AG)	0.443	0.898	0.500	0.796
Random Forest Scores				
Task A	0.299	0.714	0.500	0.427
Task B (Macro)	0.370	-	-	-
Task B (HS)	0.299	0.714	0.500	0.427
Task B (TR)	0.640	0.668	0.745	0.669
Task B (AG)	0.169	0.602	0.500	0.204

Table 1 lists the results of our initial system for both Task A and Task B. As illustrated, our initial system achieved an F1-score of 0.299 in identifying hate speech in English. For the personal target classification, we achieved an F1-score of 0.64. As for the aggressive behavior detection, our initial system has an F1-score of 0.169. Our current system under performs the baseline in all metrics, except for Recall on the Target-vs-Group classification subtask of Task B.

6 Discussion

The model significantly over-predicts that a given tweet can be categorized as hate speech (HS=1) and aggressive (AG=1). We suspect this is due to the balanced-subsample class weighting method, resulting in the model becoming biased towards these specific classifications. Future iterations of this project will benefit from an exploration of various resampling or data augmentation approaches, in order to prevent the model from biasing towards a particular class label. In addition to this issue with the classifier, we have identified some weak-

nesses in our current feature set that we hope to improve upon in future iterations.

For embeddings, GloVe twitter embeddings were chosen due to results from (Basile et al., 2019) showing that the use of GloVe embeddings helped the fifth top performing team obtain their results. However, the higher ranked teams used different embedding types, which we also have access to and will explore in future versions of our system. Namely, these are sentence level embeddings from Google's Universal Sentence Encoder, fastText word embeddings, and BERT word embeddings trained on Twitter data (BERTweet embeddings, in our case). In exploratory analysis of models with these embeddings types as features we expect to find better performance. Additionally, as we explore different classifier models we may see that embeddings having a larger effect on classification choices and therefore bringing our scores above the Baseline. For example, most of the highest ranked teams in the task used the word embedding types named above as inputs to Neural Network architectures like CNNs, LSTMs, and Bi-LSTMS.

The majority of our non-embedding features involve stylistic information about the tweet that may be useful in detecting which tweets are more likely to involve hate speech. These include the percent of the tweet that is capitalized and normalized counts of special symbols $, !, ? and *. When combined with other features that contain more semantic information, we expect these stylistic features may provide an edge in detecting hate speech. However, with the current set of features they don't appear to contain adequate information for the random forest classifier to utilize when performing hate speech classification.

In addition to the aforementioned features we included normalized counts of words that fall into different emotion and valence categories from the NRC lexicon. These counts are believed to contain more semantic and pragmatic information that may be useful in detecting hate speech. Unfortunately, this feature is limited by the lexicon to which the tweet words have been matched.

7 Ethical Considerations

8 Conclusion and Future Work

In future iterations of this project we plan to improve our feature set in several ways. We plan to add embedding features that represent the hashtag and emoji information, which is currently removed from the text during the data cleaning process. We also plan to utilize a slang dictionary in order to generate valence scores for slang terms that appear in the tweet text. We will also improve the NRC word counts feature by training a linear classifier to predict the emotion and valence category values using word embeddings, so that we can extend our counts to include all words in the text, and not just those that are found in the lexicon. Finally, we plan to explore a variety of different classifiers and different configurations for each.

A Appendix: Workload Distribution

- **Ben Cote**:
  - Design of system architecture, configs, requirements
  - Implemented lexicon-based NRClex features
  - Adapted and implemented evaluation code
  - Compiled System Overview, Approach, and Results sections
- **Madhav Kashyap**:
  - Researched shared tasks
  - Researched and set up configuration for future deliverables
  - Set up GitHub repo
- **Lindsay Skinner**:
  - Set up frameworks for the DataProcessor, FeatureEngineering and ClassificationModel classes
  - Implemented the data processing code
  - Implemented the feature-normalization method in FeatureEngineering
  - Implemented the classification model code
  - Contributed to the Discussion and Conclusion and Future Work sections
- **Keaton Strawn**:
  - Researched shared tasks
  - Implemented GloVe, FastText, BERT, and Universal Sentence Encoder embeddings in FeatureEngineering class
  - Contributed to the Discussion and Feature Engineering sections
- **Allan Tsai**

- Researched shared tasks
- Researched resources for slang dictionary
- Project system write-up

B Appendix: Code Repository & Additional Resources

Our team's repository can be found here on GitHub or directly via this URL: https://github.com/madhavmk/affect_classification_ling_573

Additional Resources:

- nltk

References

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.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*, pages 512–515, Montreal.

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Alison Ribeiro and Nádia Silva. 2019. Inf-hateval at semeval-2019 task 5: Convolutional neural networks for hate speech detection against women and immigrants on twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 420–425.

Zeerak Waseem. 2016. Are you a racist or am I seeing things? annotator influence on hate speech detection on Twitter. In *Proceedings of the First Workshop on NLP and Computational Social Science*, pages 138–142, Austin, Texas. Association for Computational Linguistics.

Zeerak Waseem, Thomas Davidson, Dana Warmsley, and Ingmar Weber. 2017. Understanding abuse: A typology of abusive language detection subtasks. In *Proceedings of the First Workshop on Abusive Language Online*, pages 78–84, Vancouver, BC, Canada. Association for Computational Linguistics.