Hate Speech

Anna Astori, Annie Thorburn, José Molina, Jake Freyer
Brandeis University
Email: {akosauro, lthorburn, josemolina, freyer}@brandeis.edu

1. The goal of the task

The task of the present project was to annotate hate speech in Twitter posts, with the goal of training a machine learning algorithm to detect it independently, which can have potential applications in automated forum moderation and filtering applications for social media.

A key component to this task is identifying what constitutes as hate speech and what does not, however the line between the two is often not clear-cut, which required a rather thorough annotation guide in order to insure good inter-annotator agreement.

There is a number of situations that pose a difficulty to identify a tweet as hateful or not: distinguishing between use of a slur by someone outside the group it refers to (probably hate speech) and by someone within the group (probably a reclamation rather than hate speech); identifying statements that seem innocuous out of context but discriminatory in a particular context (or coming from a specific source); identifying statements that may sound discriminatory out of context but appear in a context where it is clear that prejudiced implications are not intended; drawing a boundary through fuzzy areas that toe the line between hatefulness and legitimate discourse on a sensitive topic—including areas like comedy and political speech where reasonable people disagree on what constitutes an offensive statement.

Not everything that could be considered offensive is hate speech; there must be a component of hostility toward a particular demographic.

2. Overview of the Annotation Guidelines

Our DTD specified three extent tags: Group, Sentiment, and Sterotype; and one link tag: Ref. The Group tag contained various attributes such as race, nationality, religion, gender, genderID (gender identity), and sexual_orientation, each with a drop down list of some of the common possibilities, as well as a final category, other, which could then be written in. It also contained a yes or no attribute for hate, a yes or no attribute for reverse, and attribute slur, which could be hate, reclamation, sarcasm, or outside_ref. The Sentiment tag had a type attribute which could be judgement, incitement, warning, reaction, or other. The Stereotype tag could be mental, physical, behavioral, or other. The Ref link tag was to be used to link a Sentiment or Stereotype back to the Group it refers to or to link a Group to a previous mention of the same Group.

The guidelines went into detail, with examples for many, of each of the options of each of the attributes for the tags. They described what was considered to qualify as each of those options and what was not, including what was considered hate speech and what was not, which was definitely as best and as clearly as possible, knowing that this is a very subjective matter.

3. Collecting Data

The data for this annotation project was collected from a Twitter feed. We created an account, @hatespeechMLA, with which we followed as many persistently hateful tweeters as we could find—mostly white supremacists, but with a generous helping of homophobes and anti-Islamic nationalists too. (It took only a few days of expanding our list to follow before Twitter recommended that we follow Donald Trump. We did not take its advice..)

To download the data, we used the Twython Python wrapper for the Twitter API. We employed three approaches to collecting tweets:

1. scraping our account's timeline
2. scraping the timelines of some particular users that we were following (good for distinguishing hateful and nonhateful tweets written with similar attitude)
3. collecting live tweets based on key terms (helpful for getting examples of reclaimed slurs, and certain kinds of hate speech—classism, for example—that is often interspersed with harmless commentary)

Overall, the set distributed to the peer group contained 400 tweets.

4. Difficulties with Collecting Data

All of the weeks had some annotations that contained minor errors including spans errors and/or illegal character errors that were fixed with Python scripts or manually.

Week 1:

On the first week, we miscalculated the time it would take the annotators to read through and familiarize themselves with the guidelines and DTD, as a result we only received an average of 33 of the 100 tweets annotated per person.

Week 2:

The second week, MAE had lag issues caused by the high number of tags per file. The one file of 100 tweets needed to be broken up to avoid this issue, but it resulted in one annotator not splitting his up at all, one splitting up the file of 100 tweets into 8 files of variable sizes, and one splitting it up into 10 files of 10 tweets each.

These were, unsurprisingly, completely inadjudicable, and any attempt at automating a way to consolidate them in a uniform fashion resulted in filed that were not readable by MAE. The solution was to manually convert the 1 file and 8 file annotations to the 10 files of 10 tweets each format manually by opening them in a text editor and the unannotated file of tweets in MAE and copying over tag by tag what the annotator had done into the new files. This was done over several days until the files were adjudicable.

Weeks 3 and 4:

These files were all done in the 10 file with 10 tweets each format and simply had minor errors and a few MAE-readability errors that were fixed with scripts and/or manually.

5. Postprocessing

After editing the files into a format that was adjudicable. They still required some postprocessing for the IAA numbers to be most meaningful. Several of the annotations did not follow some of the specifications of the guidelines such as including punctuation in the spans when a whole sentence is tagged as a `Sentiment` or `Stereotype`. These were fixed with a Python script.

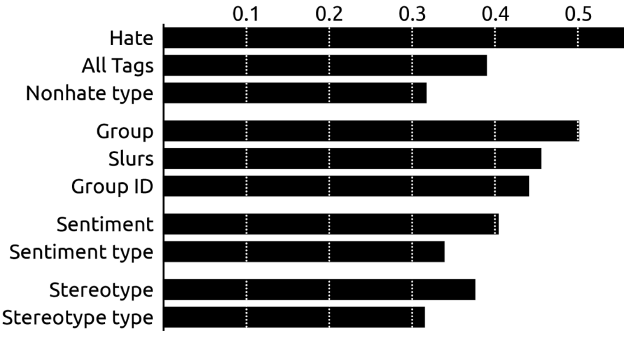
One very particularly prevalent issue was that, although the guidelines specified to write link tags `from <Sentiment/Stereotype> to <Group>` tags, some of the annotators inconsistently reversed them as `from <Group> to <Sentiment/Stereotype>`.

These did not match up with other annotators who had included the same text for the two arguments but in the correct order and would result in disagreement, when they in fact are saying the same thing. These were fixed with a Python script using BeautifulSoup to swap the `toID` and `fromID` and also the `toText` and `fromText` attributes of the link tag if the `fromID` attribute was a `Group` tag.

6. Inter-Annotator Agreement

Because we had three annotators, we calculated agreement using Fleiss's kappa. In order to make agreement calculations of tag spans meaningful on sparse data—many words were untagged, so the base agreement was unreasonably high—we excluded from the calculation any words to which no annotator assigned any tag. For convenience of calculation, and also to minimize the effect of unclear guidelines on the extent of spans, we considered a word to be fully covered by any tag that extends across at least one character of it.

The overall agreement on classifying tweets as hateful (defined as having either a group tag with a `hate="yes"` attribute or a sentiment or stereotype with no `nonhate` attribute) or not was 0.56, which puts us at the high end of "moderate" agreement. Given that our annotation task is inherently ambiguous and subjective, and deeply interpretation-based, we consider this to be a very good figure.



At right is a chart of Fleiss's kappa scores for various tags and attributes. Predictably, each specific tag showed lower a agreement rate than the general hate-nonhate judgement: our annotators developed a good

instinct for what constituted hate speech, but the components of that judgement were often more difficult.

There was substantial agreement on tagging groups (0.50); annotators generally tagged target groups correctly, but were uneven on tagging groups that were not targets. (The guideline included this direction but did not emphasize it.) The agreements on identifying the group and classifying the reference as a slur were just barely lower than the general group agreement—which is to say that, assuming agreement on the presence of a group tag, the agreement on those attributes was almost perfect.

Sentiments and stereotypes saw somewhat lower agreement (0.40 and 0.38, respectively). There was notable inconsistency on the extents of these tags, and the distinction between a sentiment and a stereotype was often blurred even during adjudication. As with the group tag, the agreement was only somewhat lower on the type of stereotype or sentiment than on the tag in general.

7. Machine Learning Experiment

For our machine learning experiment, we used a naïve Bayesian classifier, implimented using NLTK's *classifier* package. Because we had a relatively small sample size, we were not able to use separate training, devtest, and test sets; instead, we used a random split test set. Each time the classifier was run, 80% of the data was randomly assigned to the training set, and 20% was assigned to the test set.

The categories for the classifier were "hate" and "nonhate"; tweets were labeled with the appropriate category based on how annotators had tagged them. If a tweet was tagged with a "group" tag in which the "hate" attribute was "yes," or a "stereotype" or "sentiment" tag that did not have the "nonhate" attribute, it was labeled "hate"; otherwise, it was labeled "nonhate.

Our baseline classifer used as features each individual word in the corpus. The average accuracy of this classifier was 0.5686, based on 5 trials.

The first classifier that we tested used as features the entire tagged text in each "group" tag in the training data (usually a single word, but sometimes a string of several words), and each individual word, bigram, or trigram in the tagged text of each "sentiment" and "stereotype" tag in the training data. This classifier had an average accuracy of 0.5943, based on 5 trials.

In order to assess whether this classifier was overfitting, we also tried variations on it that excluded some of the features that the first version had used. Excluding "group" tags in which the "slur" attribute was "none" yielded an average accuracy of 0.6031; excluding single-word features from "stereotype" and "sentiment" tags yielded an average accuracy of 0.6086; exluding trigram features yielded an average accuracy of 0.62; excluding bigram features yielded an average accuracy of .5657 (based on 5 trials in each case).

We next tested a classifier that used as features each individual word, bigram, or trigram in the tagged text of each "group," "sentiment," and "stereotype" tag in the training data. The average accuracy of this classifier was 0.7, based on 5 trials -- the highest accuracy of any of the classifiers we tested. The "most informative features" for this classifier were puzzling, however; while the other classifiers typically had words pertaining to demographic groups among their most informative features, the most informative features for this classifier, in every trial, were exclusively single-character words. For example, in one trial, the top five most informative features were "contains(4) = true" (was 6.4 times as likely to appear in nonhateful tweets as in hateful tweets), "contains(2) = true" (2.1 times as likely in nonhate as in hate, "contains(u) = false" (2.1 times as likely in nonhate as in hate), "contains(6) = true" (2.1 times as likely in nonhate as in hate), and "contains(&) = true" (2.0 times as likely in hate as in nonhate).

In order to assess how much information these counterintuitive features were contributing, we tested a variation on this classifier that excludes single-character words from the set of word features. The most informative features in this version of the classifier included bigrams and trigrams in addition to single words, and as in most other versions, they included terms pertaining to demographic groups (mainly race), though they did not include any slurs. The average accuracy of this version of the classifier was 0.64, based on 5 trials.

8. Conclusion

In this project, we undertook an ambitious effort to quantify a concept that persistently defies precise definition. We wrote a guideline that specified as clearly as possible how to identify hate speech, and how to separate out its components. Given the clear difficulty of the task, we are proud of the "moderate"

agreement that we achieved. Though we have not yet determined how best to feed our gold standard into a machine learning algorithm, we have achieved accuracy substantially better than chance, and we have reason to believe that with further work it could substantially improve. We are hopeful that the model we have created may be refined in the future and put to use for the study and filtering of hate speech, so that future generations can inherit a less hateful online community.

9. References

Bird, S., Klein, E., & Loper, E. Natural Language Processing with Python. O'Reilly Media.

Stubbs, A. & Pustejovsky, J. Natural Language Annotation for Machine Learning. O'Reilly Media.