Detecting Racism in Text : Progress Report

Fiete Botschen, Sean Noran

November 12, 2015

1 Motivation

Racism is a world-wide ubiquitous problem that violates essential human rights. In recent years, racist comments have become much more frequently expressed as a result of the anonymity established in the web. In particular, many Germans have expressed hatred and xenophobia toward the recent influx of refugees. Online news sources such as *Spiegel Online*[1] have completely disabled viewer comments for many articles as a result of the profusion of offensive comments and their inability to moderate posts effectively. In response to pressure by organizations in Germany, Facebook has pledged to combat hate speech on its German-language network[2]. Many argue, however, that Facebook remains a platform which disseminates racist thought[3].

2 Previous Work

Surprisingly, we have found minimal research on this topic. Greevy and Smeaton[4] used a Support Vector Machine with a polynomial kernel to classify documents as racist over both bag-of-words features and word bigrams separately. They achieve 92.78% precision and 90% recall on 200 documents after training their model on 800 documents.

[1] http://www.spiegel.de/

[2] Zeller, Frank. "Facebook Pledges to Combat Racism on German Platform." Yahoo! News. Yahoo!, n.d. Web. 23 Oct. 2015.

[3] Rauch, Shannon, and Kimberly Schanz. "Advancing Racism with Facebook: Frequency and Purpose of Facebook Use and the Acceptance of Prejudiced and Egalitarian Messages." Computers in Human Behavior 29.3 (2013): 610-15. Print.

[4] Greevy, Edel, and Alan Smeaton. "Classifying Racist Texts Using a Support Vector Machine." SIGIR 4 (2004): 468-69. Print.

Kwok and Wang[5] detect racism specifically against blacks in Tweets with 76% accuracy using a Naive Bayes model over bag-of-words features.

Rauch and Schanz[6] use a multi-level Naive Bayes and rule-based approach to detect offensive language, a more general task than detecting solely racism.

Warner and Hirschberg[7] use a Support Vector Machine to detect hate speech over labeled data provided by Yahoo! and the American Jewish Congress. They used unigram and bigram features, as well as part-of-speech tags over windows and log-likelihood ratios between counts of positive and negative instances. The highest F1 score they achieve is 0.63.

3 Data Collection

To our knowledge there are no existing public datasets that we could leverage to train our model. We have requested the labeled data of racist tweets from Kwok and Wang; however, their data only contains instances of racism against blacks, while we are interested in more general cases of racism. We have also requested the datasets from Warner and Hirschberg. We may additionally construct our own dataset of tweets and label the data either manually or using a crowd-sourcing framework such as Amazon Mechanical Turk.

4 Software Dependencies

We will be working primarily in Python. We will use scikit-learn to explore various Machine Learning techniques, i.e. Support Vector Machines, Random Forests. If time permits, we would also like to explore deep learning techniques such as Recurrent Neural Networks using Theano/Lasagne.

5 Baseline

For our baseline performance, text will be classified by instances from a pre-defined collection of words that are relevant to racism. That is, if a text contains

[5]Kwok, Irene, and Yuzhou Wang. "Locate the Hate: Detecting Tweets against Blacks." AAAI Conference on Artificial Intelligence 27 (2013).

[6]Rauch, Shannon, and Kimberly Schanz. "Advancing Racism with Facebook: Frequency and Purpose of Facebook Use and the Acceptance of Prejudiced and Egalitarian Messages." Computers in Human Behavior 29.3 (2013): 610-15. Print.

[7]Warner, William, and Julia Hirschberg. "Detecting Hate Speech on TheWorldWideWeb." LSM Proceedings of the Second Workshop on Language in Social Media 12 (2012): 19-26

any words that are relevant to racism, then it will be classified as racism. The obvious flaw in this is that racism is defined not by the individual words but by the meaning conveyed in the context surrounding those words.

6 Preliminary Experiments

Our first step is to get the data we are interested in. If we do not receive the data we requested, then we may have to label our own dataset manually. We may also create a small dataset regardless, while we wait for a response. We will then extract relevant features over text such as bag-of-words descriptors or bigram features, as described in previous work. Then we will train and evaluate a classifier using these features and see how it performs in comparison to our baseline approach. We expect to have these preliminary results in our progress report submission.