Ling 506: Affective Computing

- Some Past Projects -

1. **Predicting the Emotion Intensity of Tweets**

**Task:** The WASSA-2017 shared task on emotion intensity [[Mohammad](https://www.aclweb.org/anthology/W17-5205.pdf)

[and Bravo-Marquez, 2017](https://www.aclweb.org/anthology/W17-5205.pdf)]

**Dataset:** The WASSA-2017 shared task on emotion intensity annotated datasets for four emotions and baseline prediction results. There are four emotion categories that we are interested in detecting their intensities: anger, sadness, joy, and fear. The training and testing data contain tweets labeled with one of the four emotion category and the intensity of the emotion category, as values between 0 and 1.

**Approach:**

1. Build 1-2 baseline models

2. Do error analysis (i.e., where the baseline models fail)

3. Do detailed analysis of popular features in the community and determine additional features that improve over the baseline(s) results;

4. Explore with different machine learning algorithms (e.g., compare various neural network models in my experiment, using GloVe for word embedding feature vectors.)

5. Analyze and discuss the results;

1. **Sarcasm Detection Using LSTMs With Word-Level Features**

**Task**: Sarcasm detection in tweets

**Data:** 1.3 million sarcastic tweets called the Self-Annotated Reddit Corpus ([Khodak, Saunshi and Vodrahalli, 2017](https://www.aclweb.org/anthology/L18-1102.pdf)).

**Approach:**

Context and word order are very important factors to take into account when looking at a sarcastic statement. One possible approach is to build a model that incorporates semantic meaning and context by using word embeddings, and passes them into a Long Short Term Memory (LSTM) neural network to take word order into account. Word embeddings can be augmented by additionally encoding the presence of specific features (ellipses, exclamation marks, and all capital letters) which yields us an additional .0099 accuracy. The model was also tested on the [Riloff et al. (2013)](https://www.aclweb.org/anthology/D13-1066.pdf) dataset.

1. **A Sexism Detection System**

**Task:** Detecting explicit sexist hate speech toward women.

**Data**: There is no baseline, data, or corpora specific for the creation of such a

system. Researchers have not even agreed on a sole definition of hate speech. Therefore, my analysis is merely a step toward building better, more inclusive hate speech detection systems.

One option is to use the Waseem and Hovy’s publicly available corpus of more than 16K tweets (https://github.com/zeerakw/hatespeech) that underlie their paper “Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter.” From this corpus, one could create a new dataset with tweetIds labeled as either “sexism” or “not sexism.” Useful lexical resources are the Hatebase (http://hatebase.org), Wikipedia’s lists of misogynistic slurs, sexuality and gender-related slurs, slang terms for women, Sacraparental’s “122 Subtly Sexist

Words About Women” as well as AFINN and NRC Emotion and Sentiment Lexicons.

**Approach:**

- experimented with several features in supervised machine learning (like

Multinomial Naïve Bayes, Logistic Regression, and Support Vector Machines) using the packages such as scikit-learn, pandas, and the NLTK. I also created a

python script that allows the user to provide their own strings and have them classified as “sexism” or “not sexism.”

4) **Affective Features For Automated Token-Level Metaphor Detection**

**Task:** State-of-the-art approaches to the token-level metaphor detection task focus on employing a basket of word-level features for classifying usage of tokens in running text as literal or metaphorical.

**Data**: the [VUAmsterdam metaphor corpus](https://pdfs.semanticscholar.org/c33f/e3ea3ee2b8241c972b74661f52dcab7bbbf3.pdf) which consists of a subset of articles from the BNC Baby corpus tagged for metaphorical content and sampled into Academic, News, Spoken, and Fiction categories. Articles and tags are encoded in a hierarchical XML format where every sentence and token is tagged with multiple types of metaphors as well as useful information such as part-of-speech. I also compared the results with those from the [Wee et al. model](A%20Report%20on%20the%202018%20VUA%20Metaphor%20Detection%20Shared%20Task).

**Approach:**

In this paper we propose a supervised machine learning system for token-level metaphor detection that incorporates affective features in addition to unigram features commonly used for this task. We show that the introduction of features capturing emotional categories of words produces substantial performance improvements compared to baselines that mainly use features from the word-level granularity.