FN Project - CSC583

Adam King

May 9, 2017

1 Classification

To begin, a high-level look at the task at hand. We are given a long list of *headlines* from news articles and a long list of *text* from news articles. Our task here is to predict the type of relation a headline and article body have: whether the two are **unrelated**, the body of the text **disagrees** with the headline, the body of the text **agrees** with the headline or if the body of the text simple **discusses** the topic of the headline, but takes no major position.

To accomplish this, my classifier had **four** steps, which I will describe quickly here and in more detail later.

1. Pre-processing
   The first step was to take the two `.csv` files, load them into the classifier and processing the text (either stemming, lemmatizing or both)

2. Classify Related vs. Unrelated
   This involved using a cosine similarities `tf-idf` scores for documents/headlines to determine if they were **unrelated** or one of the other labels

3. Classify Disagree vs. Discuss/Agree
   This step involved training a Support Vector Machine (SVM) to distinguish between **disagree** labels and **discuss/agree** labels

4. Classify Discuss vs. Agree The final step involved another SVM, this time to split **discuss** and **agree**

1.1 Pre-processing

To build the classifier, I used `python3` and various packages in `nltk` and `sklearn`. For preprocessing, I first loaded in the text from the articles and

all headlines, to build a set of vocabulary terms. I folded case for all terms, but tried versions of the classifier where I 1) left the terms unmodified, 2) stemmed all terms and 3) lemmatized all terms. As an important note, it is a bad idea to use data from testing in training. However, when building the vocabulary, I did **not** store any information about a terms frequency, only if it existed or not. This was a result of the technical detail of implementation; in a term-incidence matrix, you need to know what terms you're *going* to see before you can count.

1.2 Related vs. unrelated

The next step of classification was to distinguish **related** from **unrelated** documents. In this case, **related** documents were any of the **discuss, disagree** or **agree** labels. To do classification here, I used `sklearn`'s implementation of a `tf-idf` vectorization and cosine similarity. To do so, I converted each headline and document into lists of terms and then converted those list of terms into `numpy` arrays where each value in the array indicated the frequency of a specific term. Using each array as a point in N-dimensional space, I calculated the cosine similarity between each headline and corresponding document. After surveying the range of values for the **related** and **unrelated** documents, I found the a cosine similarity of `.1` was a very good indicator of relatedness.

So, knowing this, whenever the model saw a document-headline pair that it did not have gold data for, it would classify it as **unrelated** if the cosine similarity was less than `.1` and as **related** if greater than `.1`.

1.3 Disagree vs. discuss/agree

The next step of the classification was an SVM to disambiguate between **disagree** labels and **discuss** and **agree** labels. To do so, I scored each headline-document pair with a variety of real-valued features and then projected these features into N-dimensional space. To classify, the SVM finds the line that best separates all labels of one type from another. In this case, each feature was hand-written and something *I* thought would be an indicator of the proper label.

To build the features, I first went into the headline and did part-of-speech tagging. I did this to get the verbs of the headline (as they indicate the *actions* and the nouns of the headline (as they indicate the actors or patients). Any document that either agrees or disagrees with the relevant headline is likely to include a sentence that either describes the *action* or *participants* of the action **and** words that are either contradictory (like negation) and confirmatory (like

proved). Many of my features involved looking at the proximity of terms from the headline and words that signaled some sort of polarity.

The features were:

- The number of times negation appears near a term from the headline
- the number of times a "disagreeing" word appears near a term from the headline
- The number of times an "agreeing' word appears near a term from the headline
- The number of times a term indication "discussion" appears near a term from the headline
- The number of times negation appears near a term from the headline
- The shared n-grams in headline and body (2, 3 and 4)
- The count of the various types of words: agreement, disagreement, discussion, negation

Once each document/headline pair had its value for each feature, I trained the SVM to distinguish **disagree** labels from **agree/discuss** labels.

1.4 Discuss vs. agree

This section of the classifier was very similar to the previous. In this case, the features were the same, but it was trained **only** on distinguishing between **agree** and **disagree** labels.

2 Measuring Performance

2.1 Results

Below are the results of the classifier using the FN Challenge's scoring function. All training was done on `train_stances_csc483583.csv` and all testing was done on `test_stances_csc483583.csv`.

Lemmatization/stemming performed identically on the task and did slightly better overall than without any form of tokenization.

- No stemming/lemmatization

```
-------------------------------------------------------------
|           |   agree   | disagree  |  discuss  | unrelated |
-------------------------------------------------------------
|   agree   |    99     |     0     |   1888    |    109    |
-------------------------------------------------------------
| disagree  |    14     |     8     |    428    |    37     |
-------------------------------------------------------------
|  discuss  |    101    |     0     |   3804    |    205    |
-------------------------------------------------------------
| unrelated |     7     |     0     |    120    |   10135   |
-------------------------------------------------------------
Score: 7052.5 out of 9258.5 (76.1732462061889%)
```

- With stemming

```
-------------------------------------------------------------
|           |   agree   | disagree  |  discuss  | unrelated |
-------------------------------------------------------------
|   agree   |    124    |     0     |   1880    |    92     |
-------------------------------------------------------------
| disagree  |    22     |     7     |    427    |    31     |
-------------------------------------------------------------
|  discuss  |    131    |     0     |   3841    |    138    |
-------------------------------------------------------------
| unrelated |     1     |     0     |    148    |   10113   |
-------------------------------------------------------------
Score: 7115.25 out of 9258.5 (76.85100178214614%)
```

- With lemmatization

```
-------------------------------------------------------------
|           |   agree   | disagree  |  discuss  | unrelated |
-------------------------------------------------------------
|   agree   |    124    |     0     |   1880    |    92     |
-------------------------------------------------------------
| disagree  |    22     |     7     |    427    |    31     |
-------------------------------------------------------------
|  discuss  |    131    |     0     |   3841    |    138    |
```

```
-------------------------------------------------------------
| unrelated |     1     |     0     |    148    |   10113   |
-------------------------------------------------------------
Score: 7115.25 out of 9258.5 (76.85100178214614%)
```

In this case, the scoring is done via a modified F-score. In this case, simple accuracy is *not* a good measure in that some labels (**discuss** and **unrelated**) are much more prevalent than the other two. F-score shows how good the precision and recall is on *each* class and so is a better measure for this task than accuracy.

3 Error Analysis

For this task, it seemed that lemmitization or stemming improved the score, but not by much. Regardless of how the tokens are processed, though, the classifier is very good at determining **related** and **unrelated** labels. That being said, my system does very poorly on classifying **disagree** and **agree**. In this case, the system assigned the label **discuss** when it should not have frequently.

Looking through some of the misclassified documents, I saw that it was common for terms to be mentioned in the headline, but their *synonyms* to be mentioned in the document. As it stands now, my classifier does not know the difference between *well* and *good*. If *good* appears in the headline but *well* and not *good* appears in the document, the classifier would not see the connection. Similarly, I also saw a lot of antonyms showing up in disagreement documents. That is, if *good* appears in the headline, *bad* often appears in the document. A better classifier would also be able to handle antonyms to capture what I missed.

I also looked through the documents and saw some terms that seemed to indicate agreement or disagreement, that I did not include in my list of polarity terms.

One final error that I found in my system was not handling the *distance* between relevant terms. After reading a paper posted on Piazza, I decided to try another feature that is based off of the distance between documents in the body that also appear in the headline. That is, if a document **discusses** but does not **agree** or **disagree**, it is more likely that words that it shares with the headline will be more spread out in the document.

I decided to add a couple more features to each document-headline pair and re-run the classifier to see if anything improved.

4 Improving Classification

In the *more advanced* version of the classifier, I added the following features.

- Added synonyms to the list of relevant items. To do this, I used `WordNet` to find all synonyms for verbs and nouns in the headline.
- Added features to look to antonyms in the document body.
- I added a feature that looks at the average and maximum distance between relevant terms in the body.

Unfortunately, these new features did not have the desired effect. In fact, the score went down.