Mini-Projects 1 and 2, Sentiment Analysis, SS2016

Mariia Kashpur
University of Stuttgart
mariia.kashpur@gmail.com

1 Mini-Project 2: Unsupervised Learning of a Polarity Lexicon

First, I worked on the second mini-project: *Unsupervised learning of polarity lexicon*, and then I did the task from the first mini-project, i.e. used my own polarity lexicon to classify reviews.

The code for the first part is contained in the "polarity.py" file (see "README.txt" file for command line arguments). Overall, the polarity lexicon creation task was not very difficult, but I expected better results. The accuracy I was able to obtain is only about 65 per cent (0.65590809628 to be exact), and it is even lower if a smaller data file is used.

The approach used was to extract positive and negative lexicons from the provided files, and collect statistics for each of these words occurring within a 5-window frame with words *excellent* and *poor*. The statistics were obtained from a corpus file provided. Later, the total word polarity was calculated using the formula from the handout: SO(X) = log (hits(X NEAR excellent) * hits(poor) / hits(X NEAR poor) * hits(excellent)). 0.01 was added to all counts before calculation.

The results can be seen in the "lexicon.p" (pickle file) or "generated_lexicon.txt" files: if the score is positive, the word was classified as such bearing positive meaning, and vice versa.

Examples of words where the program made a mistake:

- *hating* got a score of around 3.5 (you would think it should be negative), and I have actually found this example from a corpus: "this is another movie where you find yourself actually rooting for the vampire instead of hating them . this is an excellent movie..." The word in question occurs near the word *excellent*, and although in this example the message sentiment is positive, you could think of a different example where the sentiment is negative.

- *best-selling* got a surprisingly high negative score (-5.7), here is an example of where the scoring went in a wrong direction: "... adaptation of the best-selling memoir of growing up poor in ireland"... Obviously, *poor* is just factual information here, without any negative connotation.

2 Mini-Project 1: Document Classification Using a Polarity Lexicon

The code for this part is contained in the "classify.py" file. After I created the polarity lexicon, I saved it into a pickle file ("lexicon.p") and used to classify each document from the "pos" and "neg" folders with movie reviews. The documents can be either positive or negative in its polarity. I computed each document's score using my polarity lexicon according to the formula: score(d) = num of positive terms - num of negative terms. If the score is larger than 0, it's a positive document, if it is smaller than 0, it is negative. For 0, I chose to return positive score. The classification results for all documents can be seen in "classification_results.txt".

I then evaluated each review from both folders and counted overall predictions accuracy, which amounted to 0.7165.

Error analysis of correctly and wrongly classified documents:

- Correctly classified: predicted score for file "pos/cv002_15918.txt" is 5.

  This is a positive review of "You've Got Mail" using many words from polarity lexicon with a positive score: *better*, *popular*, *cute* etc.

- Incorrectly classified: predicted score for file "neg/cv799_19812.txt" is 12.

That's a negative review of "The Blair Witch Project". The positive word *intelligent* is used in the review in the sentence "But i have intelligent friends who like this movie...", so I think there's nothing we could have done to help in this particular case. Same story with *recommend*; the author writes: "for current films , i highly recommend the sixth sense", so again the bag of words problem gets in the way. The word *chance* is used to say the author hopes that there is a good chance that the directors will regret filming this movie. So, I think the problem with this review was not in the polarity lexicon quality but rather in the whole approach of using polarity lexicon to predict sentiment. Here, all the positive words were used to convey the negative emotion.

Overall, it was quite interesting to work on the projects. I found them quite useful, especially the possibility to use the outcome of one project for doing the other.