Ian Laird & Andrew Walker

Natural Language Processing

May 8, 2019

**Final Project Report**

Our program starts out by finding the potential aspects of the reviews. To do this we first create a corpus of the reviews using the MyCorpusReader class created by the first undergraduate group. Next we run the TF-IDF model created earlier in the semester on the review corpus (it is trained on the Brown Corpus). This allows the most important words in the review corpus to be found.

These potential aspects are further filtered down by finding the most common bigrams in the review corpus. For each common bigram (w1,w2) if both w1 and w2 are potential aspects we no longer consider w2 as a potential aspect. The reason for this is that if two aspects appear together frequently, they are probably both part of the same aspect.

All potential aspects that do not appear most commonly as nouns are eliminated (using nltk.pos\_tag()) and any aspects that contain punctuation or numeric digits are removed.

At this point the wu palmer distance between all remaining potential aspects is calculated using the method created by the second undergraduate group. All potential aspects with an average Wu Palmer distance of below the cutoff point are eliminated. For this project we used 0.35 as the cutoff point, which was found using trial and error.

We then used the Stanford Dependency Parser to parse the review corpus, and we only considered aspects that appear as a subject at least once in at least one of the parse trees.

The potential aspects remaining at this point are considered to be the aspects of the reviews. Each sentence that contains an aspect (guaranteed to be at least one for each aspect) is analyzed for its sentiment. Sentiment detection is done using machine learning (sk-learn). Each sentence that contains an aspect can either be positive, negative, or neutral.

Detailed instructions for how to get the project running are in the readMe.md file in the project. The most important part of the readMe is duplicated below.

* Download CoreNLP at <https://stanfordnlp.github.io/CoreNLP/download.html>
* Navigate to where this is unzipped and run the following command (this is used for the dependency parsing)
  + java -mx4g -cp "\*" edu.stanford.nlp.pipeline.StanfordCoreNLPServer -preload tokenize,ssplit,pos,lemma,ner,parse,depparse -status\_port 9000 -port 9000 -timeout 15000
* If it does not already exist in the project create a data folder.