Natural Language Processing (CS 6120/4120)

Assignment 3

Distributional Semantics: Brown Clustering, and word2vec

Instructor: Professor Lu Wang

Due Date: 6pm, November 20^{th}, 2017

For the programming question, you can use Java, Python or C/C++. You also need to include a README file with detailed instructions on how to run the code and commands to facilitate understanding for TAs. Failure to provide a README will result in points deduction. No need to print out the code while submitting the hard copy, just turn it in on Blackboard.

CORPUS:
For Question 1, use Brown Corpus from - `https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/packages/corpora/brown.zip`

CLEANING:
Example: "The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd Friday/nr an/at investigation/nn of/in Atlanta's/np recent/jj primary/nn election/nn produced/vbd "/" no/at evidence/nn "/" that/cs any/dti irregularities/nns took/vbd place/nn ./."

will be: START the fulton county grand jury said friday an UNK of UNK UNK primary election UNK no evidence that any UNK took place STOP

1 Brown Clustering [65 points]

Use the Brown corpus given above.

1. The corpus is already tokenized with whitespace. Remove the part-of-speech tags appended to each token with a slash. Lower case all words. Replace all tokens with a count of 10 or less with the conventional out-of-vocabulary symbol UNK. Sort the vocabulary first by decreasing frequency and then alphabetically, to break ties. Submit this ranked vocabulary list, with counts. [10 points]

2. For purposes of the bigram model, treat each sentence on its own line as independent of the others, and assume that each sentence begins and ends with invisible START and END symbols. These symbols should not be included in the clustering, however. Turn in your code that takes the vocabulary you just produced and the corpus as input. [See cleaning example] [15 points]

3. Write a program to implement Brown clustering on this dataset. Let the initial number of clusters $K = 200$. You should therefore give each of the 200 most frequent tokens its own cluster and proceed as described above. [20 points]

   The output will consist of the same vocabulary with a string of 0's and 1's for each, indicating its path from the root of the cluster merge tree to the leaves. Since the clusters start out sorted by decreasing frequency, we stipulate that when merging two clusters, the earlier cluster on the list gets the code 0 and the later cluster gets the code 1.

   You can refer to the playlist: `https://www.youtube.com/watch?v=xGfQMrYoIx4&list=PLO9y7hOkmmSEAqCc0wrNBrsoJMTmIN98M`

4. Design a metric to measure similarity and list top 10 similar words for the following words: [20 points: 15 points for similarity, 5 points for output]

   (a) the

   (b) army

   (c) received

   (d) famous

- Along with your code submission, include a file containing your brown clustering results as *browncluster.txt* and top words results as *results-brown.txt*.

2 word2vec [25 points]

Using word2vec, derive top 10 similar words for following words.

1. the

2. army

3. received

4. famous

- You are going to use Google's pre-trained word vectors trained on a part of Google News dataset comprising of about 100 billion words.
- More information on word2vec can be found here, `https://code.google.com/archive/p/word2vec/`.
- The vectors archive is available here: **GoogleNews-vectors-negative300.bin.gz** - `https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit`.
- You can use gensim in python for implementation, or any tool you would like to use. Following links give more details on implementation. [20 points]
  - `https://radimrehurek.com/gensim/models/word2vec.html`
  - `https://rare-technologies.com/word2vec-tutorial/`
- Along with your code submission, include a file containing your results as *results-word2vec.txt*. [5 points]

3 Discussion [10 points]

Discuss the differences between Brown Clustering and word2vec. Also, elaborate on why the retrieved words are different.