CS671: Introduction to NLP
Assignment #2: Tagging, Word and document vectors, sentiment analysis

Due on: 7-10-2016, 23.59 20-9-2016
MM: 750

1. Use the Brown corpus (available on the ftp site) to learn a tagging model and then test it by using 80% of the corpus for training and 20% for testing. The corpus has been tagged using the Brown tag set. A manual on the Brown corpus is available at:http://clu.uni.no/icame/brown/bcm.html. Compare the performance for the following approaches to tagging (ensure that the same learning and test set is used for each approach).
   (a) Use the simplest possible algorithm. Give the most frequent tag in the learning corpus to words in the test corpus.
   (b) Use the generative tagging model that was discussed in class and the Viterbi decoder to find the tags of the test corpus.
   (c) Train an LSTM using the learning corpus and then predict the tag sequence on the test corpus.
   (d) Use the NLTK (available at: http://www.nltk.org/) tagger.
   (e) Use the OpenNLP (available at: https://opennlp.apache.org/index.html) tagger.

   [50,100,100,50,50=350]

2. Use the Brown corpus to create embedded vector representations for words in the corpus. Google has already trained word vectors for a very large vocabulary using a billion word corpus. Compute the cosine similarity between the vector for each word in the Brown corpus and the corresponding vector in the Google set. Google has a vector space of 300 dimension so the Brown corpus word to vector embedding should also be a dimension 300 vector space. Distribute the cosine similarity values into 10 equal sized buckets based on the range of values available and plot a histogram. This will give us an idea of how different the vectors are for a corpus of 1 Million versus 1 Billion. You can use the skip-gram technique to create the word vectors. Multiple open implementations are available on the web (e.g. gensim).

   [150]

3. The IMDB large movie review data set is available at the ftp site. More details are at: http://ai.stanford.edu/ amaas/data/sentiment/. You have to create document vectors for each review, then train a classifier on the training set and test it on the test set - both are available in the IMDB data set. You can use any classifier of your choice using any ML library. Do this for the following document representations and compare results.
   (a) Binary Bag of Words (bBoWs) - simple presence/absence.
   (b) BoWs using tf-idf.
   (c) Word2Vec vectors averaged for all words in the document.
   (d) Bag of Vectors (BoVs) - represent each word by its word2vec vector.
   (e) Weighted Bag of Vectors - weight each vector in the Bag of Vectors by the corresponding tf-idf value.

   [50x5=250]