Natural language procession - COMS 4705
Homework 3-Programming

November 15, 2013

Name: Gaurav Ahuja
UNI:ga2371

This report is divided into three sections
    1. Running the Code
    2. Observations and Comments

****
1. Running the Code
****

This section explains how to run the python scripts.

a. Creating IBM Model 1 't' parameters.
   This will create a file ibmm1.dat which will store the 't' parameter after 5 iterations of EM algorithm.
   Takes about 220 seconds to run.

   python IBM-M1.py corpus.de.gz corpus.en.gz

   Output: ibmm1.dat

b. Creating IBM Model 2 't' and 'q' parameters
   This will create files ibmm2t.dat and ibmm2q.dat which will store 't' and 'q' parameters after 5 iterations of EM algorithm.
   Assumes ibmm1.dat to be present in the directory.
   Takes about 300 seconds to run.

   python IBM-M2.py corpus.de.gz corpus.en.gz

   Output: ibmm2t.dat and ibmm2q.dat

c. Question 4
   This will read IBM Model 1 parameters from ibmm1.dat and write the top 10 german words for each english word in devwords.txt to out-q4-t_model1.txt. This script will also compute and write the arg max alignments for the first 20 sentences to out-q4-alignment_model1.txt.
   Assumes ibmm1.dat, devwords.txt, corpus.de.gz and corpus.en.gz to be present in the directory

   python q4.py

   Output: out-q4-alignment_model1.txt and out-q4-t_model1.txt.

d. Question 5
   This will read IBM Model 2 parameters from ibmm2t.dat and ibmm2q.dat and compute and write the arg max alignments for the first 20 sentences to out-q5-alignment_model2.txt.
   Assumes ibmm2t.dat, ibmm2q.dat, corpus.de.gz and corpus.en.gz to be present in the directory

   python q5.py

   Output: out-q5-alignment_model2.txt.

e. Question 6
   This will read IBM Model 2 parameters from ibmm2t.dat and ibmm2q.dat and compute and write arg max E arg max alignments for all german sentences in original.de. The English sentences are read from scrambled.en and arg max English sentences are stored in unscrambled.en.
   Assumes original.de, scrambled.en, ibmm2t.dat and ibmm2q.dat to be present.

   python q6.py

   Output: unscrambled.en

****
2. Observations and Comments
****

An accuracy of 91% was achieved in unscrambling the sentences from scambled.txt

Some statistics on the corpus:

1. Number of parallel translations: 19900
2. Number of unique German words: 26757
3. Number of unique English words: 14078
4. Nuber of unique German, English word pairs: 2,639,038
5. Average German Sentence length = 20.7 words
6. Average English Sentence length = 21.7 words

Comments of alignments form Model 1 and 2:

From the alignment outputs from IBM model 1 and 2 the following can be observed:
1. Alignments for German words like 'der', 'den', 'das', etc has improved in IBM model 2
2. Distortion seems to be less in alignments produced by IBM model 2

Comments on efficiency:

Since 't' parameters depend on only two variables the German and English word, 't' has been implemented as a two level dictionary.
Since t(f|e), it is only logical to have the english word at the first level. Hence t[e] will be a dictionary of all possible German words that e could be translated to. t[e][f] = t(f|e). The two level dictionary will efficient to find top German words for given Enlish words.

Since 'q' parameters depend on four variables it is cumbersome to implement it as a four level dictionary.

For Questions 6 the Time complexity to unscramble the input containing K German and English senteces is O (K^2*L*M), where L and M is the average English and German sentence length.