INLP course 2016-2017 Autumn Term. Laboratory assignments

The laboratory assignments in this course consist of three exercises to be delivered at the end of October, November, and December. The exercises should be implemented using Python. The norms for the program and instructions for the material to be delivered are placed in an attached document. The laboratory cases should be performed preferably by groups of two students (groups of one or three student are allowed too).

1 Zipf's Law Exercise

From WP:

Zipf's law /ˈzɪf/, an empirical law formulated using mathematical statistics, refers to the fact that many types of data studied in the physical and social sciences can be approximated with a Zipfian distribution, one of a family of related discrete power law probability distributions.

Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table.

1. Use the text file corpus/en.txt and corpus/es.txt
2. Write a program to read the corpus. Tokenize it using whatever tokenizer from NLTK or write your own tokenizer.
3. Write a program to check Zipf's first law (f = K/r) on this real corpus: Count word frecuencies, sort them by rank, and plot the curve.
4. Compute the proportionality constant (K) between rank and frequency for each word. Compute its average and deviation . Discuss the results . Are they consistent with Zipf's Law ?
5. Perhaps you have found problems with the tokenization (Word case, punctuation marks, numbers, etc. Try to fix them and repeat the ítems 3 and 4.
6. Now move to the char level. Repeat the ítems 3 and 4 using now as units not words but chars (letters and punctuation marks).
7. If your program is in python you can use access functions to the text files in auxiliar.py. For plotting there are several python libraries, one of them is matplotlib.

2 Entropy - Language Models

Use the following corpora included in directory corpus/:

- en.txt   A fragment of EFE corpus in English
- taggedBrown.txt   A fragment of Brown corpus in English Pos-tagged

and a set of python functions in auxiliar.py:

- getWordsFromFile(inF):
-     "get a list of words from a text file"
- getTaggedWordsFromFile(inF):
-     "get a list of pairs <word,POS> from a text file"
- getTagsFromTaggedWords(l):
-     "from a list of tagged words build a list of tags"
- countNgrams(l,inic,end=0):
-     From a list l (of words or pos), an inic position and an end position
-     a tuple(U,B,T) of dics corresponding to unigrams, bigrams and trigrams
-     is built

to answer to the following questions:

1. write a python function for computing the order 0 (unigram) model of en.txt

$$H = -\sum_x p(x)\, log\ p(x)$$

2. write a python function for computing the order 1 (bigram) model of en.txt

$$H = -\sum_x p(x) \sum_y p(y|x)\, log\ p(y|x)$$

3. write a python function for computing the order 2 (trigram) model of en.txt

$$H = -\sum_x p(x) \sum_y p(y|x) \sum_z p(z|xy)\, log\ p(z|xy)$$

4. Use now the taggedBrown.txt corpus. Compute the perplexity of the trigram language model for three different sizes of the corpus (the full corpus, half of it and a quarter of it).

5. Smooth the trigram language model going from the trigrams <x,y,z>, to <x',y,z> and to <x',y',z>, where x' is the POS of x and y' is the POS of y. Compute the perplexity as in the previous case. Build the following table and discuss the results.

perplexities	Full corpus	½ corpus	¼ corpus
<x,y,z>			

<x′,y,z>			
<x′,y′,z>			

3 Probabilistic parsing

1. Download NLTK, install the fragment of PTB-II treebank available in NLTK. Reserve 10% of the sentences for testing and use the remaining 90% for learning.
2. Write a program to read the corpus. Provide auxiliary functions for facilitating the management of the trees. I suggest to use the tree class provided by NLTK.
3. Build a treebank grammar for performing chunking of nominal phrases (NP).
4. Transform the treebank grammar into a probabilistic one using counts in the corpus.
5. Test your chunker against the test corpus

Barcelona, 15 September 2016