Lab 4: POS Tagging

Author: Henry Thompson
Author: Bharat Ram Ambati
Date: 2014-10-01

POS Tagset

- Based on your intution guess the most and least frequent tags in a data.

  In English data, nouns (NN, NNS, NNP etc.), verbs (VB, VBD, VBP etc.), prepositions (IN) might be more frequent. Tags like UH (interjection), LS (list marker) might be less common.

- What is the difference between the tags DT and PDT?

  DT is the POS tag for determiners and PDT is the tag for pre-determiners. As the name says, pre-determiners occur before determiners. For example, in "both the books", "the" is determiner and "both" is pre-determiner.

- Can you distinguish singular and plural nouns using this tagset? If so, how?

  We can distinguish singular and plural nouns in this tagset. NN (noun singular) and NNP (proper noun singular) are the tags for singular nouns and NNS (noun plural) and NNPS (proper noun plural) are the tags for plural nouns.

- How many different tags are available for main verbs and what are they?

  There are six different tags for main verbs. VB (base form), VBD (past tense), VBG (gerund/present participle), VBN (past participle), VBP (sing. present, non-3d), VBZ (3rd person sing. present).

Distribution of sentence lengths

- What are the minimum and maximum sentence lengths that you see? What kind of distribution is this?

  Minimum sentence length is 1 and maximum sentence length is 249. This is a *normal* distribution.

Distribution of tags

- How many distinct tags are present in the data ?

  There are 45 distinct tags in this data.

- List the tags in the descreasing order of frequency.

```
sorted(tag_dist.items(), key=lambda x:  x[1], reverse=True)
```

- What are the 5 most frequent and least frequent tags in this data. How does this compare with your intuition in the previous section ?

NN, IN, NNP, DT, NNS labels for common noun, preposition, proper noun, determiner, and plural common noun respectively are the 5 most frequent tags. 5 least frequent tags are SYM (Symbols), UH (interjection), FW (foreign words), LS (list marker) and WP$ (possessive wh-pronoun).

- Using plot_histogram, plot a histogram of the tag distribution with tags on the x-axis and their counts on the y-axis, ordered by descending frequency

```
plot_histogram(sorted(tag_dist.items(), key=lambda x:  x[1], reverse=True))
```

- What kind of distribution do you see in the plot?

  A *Zipf's law* distribution

Distribution of tags and words

- How many entries are there in your big CFD? Given what each entry corresponds to, what does this number tell us about the corpus?

  There are 11968 entries.

- What is the number of tags and the most frequent tag for the words `the`, `in`, `book`, `eat`.

  word_tag_dist["the"], word_tag_dist["in"], word_tag_dist["book"], word_tag_dist["eat"] will give these results. For example, for `book`, `NN` and `VB` is the list of all possible tags and the most frequent tag is `NN`.

- Which word out of the whole corpus has the greatest number of distinct tags? What parts of speech do they represent ?

  Words "set", have the greatest number of distinct tags, which is 5. POS tag list for the word "set" is ""VBN", "VBD", VB", "VBP", and "NN" which represent different verb forms and common noun. Words "back" and "hit" also have 5 distinct tags.

Unigram Tagger

- Why is this called a Unigram tagger? How does it differ from an HMM tagger?

  This is called unigram tagger as it uses only the current word as input. HMM taggers are *bigram* taggers: they use the previous word/tag pair as context.

- Run this simple tagger and tag the sentences a) "book a flight to London" and b) "I bought a new book". Look at the pos tags. Are there any errors in the pos tags ? If so, what could be the reason for them ?

  In the first sentence "book" is a verb and in the second sentence "book" is a noun. But the unigram tagger assigned noun tag in both the cases as noun is the most frequent tag. This is because, unigram tagger treats words in isolation. But in pos tagging context places an important role.

Going Further

1. Run the simple tagger developed on the sentence "I bought two new books." What error do you see, and what improvement can you think of that can handle it?

   Splitting the sentence based on space treats "books." as a single token. As this is an unseen word, tagger assigns the default tag. Instead of splitting the sentence, running a tokenizer `word_tokenize(sentence)` treats "books" and "." as two lexical items and hence the tagger is able assign correct tags.

2. Do you think the Penn tagset will work well for social media text such as twitter data which contains non-standard English text?

   Not really. There has been some recent work on developing a new tagset for twitter data.
   See : https://www.aclweb.org/anthology/P/P11/P11-2008.pdf

3. NLTK has libraries to train different taggers. Using these libraries, build unigram and Hidden Markov Model taggers and evaluate them. First, split the data into two parts(90%, 10%). Consider first 90% of the data as training data and the remaining 10% of the data as testing data. Build taggers using the training data. Then run the taggers on the test data and evaluate the performance of the tagger

   ```
   unigram_tagger = nltk.UnigramTagger(train_sents, backoff=nltk.DefaultTagger('NN'))
   hmm_tagger = HiddenMarkovModelTagger.train(train_sents)
   unigram_tagger.evaluate(test_sents)
   hmm_tagger.evaluate(test_sents)
   ```