CMPUT497/501 Intro. to NLP
Assignment 3: Part of Speech (POS) Tagging

Part 1: Part of Speech Tagging.

Part-of-speech tagging is an important task when analyzing natural language data. Often, tagging is an early step before the data can be processed for other tasks. The accuracy of a tagger is important: if your tagger makes a lot of mistakes, the errors propagate throughout the system, and your final results suffer. As such, the choice of a tagger is an important one that must be considered whenever we use parts-of-speech as features.

The Stanford POS Tagger is a popular and accurate Maximum Entropy tagger. Download this tagger, and train two different models: Stanford1 on Domain1Train.txt, and Stanford2 on Domain2Train.txt.

You will now repeat this process using the NLTK Hidden Markov Model tagger (nltk.tag). Download this tagger and build two different models; one trained on DomainTrain1.txt, and one trained on DomainTrain2.txt. The data has been tokenized, and contains the "gold-standard" tag, in the form of one word and tag per line.

Sentences are separated with an extra line between them. The data may require some pre-processing before using your taggers.

You are encouraged to change the parameters of your taggers however you see fit to maximize accuracy, and to report your tuning efforts in your final report. When tuning parameters, please ensure that any backoff is within the same class of taggers (i.e., if you start with a sequential tagger then you can only back off to a sequential tagger) .

The nltk also includes a second tagger called the Brill tagger. The Brill tagger is what is known as a transformational tagger: it starts with a baseline tagger, and then learns rules that change the predicted tags to increase accuracy. Please train the Brill tagger simultaneously with your other tagger and perform error analysis on both the baseline and Brill taggers.

Test your four models on their respective testing sets (i.e., test the taggers trained on Domain1Train.txt on Domain1Test.txt, and the ones trained on Domain2Train.txt on Domain2Test.txt). Record the results.

Now, test your four models on their opposite testing sets (e.g., test the taggers trained on Domain1Train.txt on Domain2Test.txt). Report all results and discuss the relative differences in the accuracies with respect to different train and test sets.

Part 2: Error Analysis.

Different taggers make different mistakes. Take a look at which words are incorrectly tagged by each of the taggers. Are there certain types of errors that one tagger makes more often than another? How do the taggers handle out-of-vocabulary words (this is particularly worth noting when crossing domains).

Part 3: Learner English

Taggers do a pretty good job on Standard English (the state-of-the-art has a word accuracy rate in the high 90's), but a lot of data is not "standard". Speakers learning English as an additional language often have peculiarities to the way they write; data whether they are from learners or online sources, such as Twitter, often contain abbreviations, extra symbols, misspellings, and other features that not only increase the percentage of out-of vocabulary words, but also change the syntax of a sentence.

First, run your models (trained on Domain1Train.txt and Domain2Train.txt) on ELLTest.txt, which is data from English language learners. Again, record the accuracy and investigate what kind of errors are being made.

Now, re-train your taggers on ELLTrain.txt, which is also data from English language learners, and test on ELLTest.txt. How does the accuracy compare to that of the taggers on "Standard" English?

Data:

There are three training sets and three test sets available in the zipped folder Assignment3Data.

We recommend you extract a subset of the training data for development purposes as this will decrease the amount of time needed to refine your approach.

For your final submission, you must train your models on the complete training sets and test on the testing sets.

The three data sets are from different domains in order to test the effectiveness of various taggers when testing on in-domain and out-of-domain data. More details of the data can be found in the README.

The data uses the Penn-Treebank Tag set; This set can be found inside the cover of the Jurafsky & Martin textbook, or online.

If you want to introduce more data into your training set, feel free, but report it as a "Non-standard" run of your task. You must run and report on each of the standard runs in addition to any non-standard runs you wish to run.

What to Submit:

- The output from each of the required runs of a test set.
- Your code and a readme file so that we can run your code on the lab machines.
- A report detailing your findings.
    - This report should be in the same format as your project report. The style files are provided under the project section of eClass.
    - This report should contain one section for each of the assignment parts
    - Please use appropriate sub-headings so that it is easier for the marker to find key information