CSE 5525 **hw2** **Connor Wood**

1. Word's Most Frequent Tag Baseline (2 points)

python baseline.py data/twitter_train_universal.txt data/twitter_test_universal.txt
python accuracy.py mft_baseline.out data/twitter_test_universal.txt
>> ACCURACY=**0.800494350282**

2. Viterbi Algorithm (6 points) & 3. Structured Perception (4 points)

Trained and tested on twitter data: **0.853460451977**

4. Cross-Domain Experiments (2 points)

Train on Penn Treebank data and test on Twitter: **0.738877118644**

Train on NPS Chat data and test on Twitter: **0.81479519774**

Train on all the data and test on Twitter: **0.876765536723**

What can you say about the performance of part-of-speech taggers when they are applied on text outside their training domain? **The accuracy is not as good as when applied to their training domain but is still okay. This makes sense because the words used in these different sources would be different and each have unique words. Combining all of them gives the best results since it knows all of the unique words now.**

5. Named Entity Recognition (4 points)

processed 11570 tokens with 356 phrases; found: 311 phrases; correct: 148.

accuracy: 95.74%; precision: 47.59%; recall: 41.57%; FB1: 44.38

ENTITY: precision: **47.59%**; recall: **41.57%**; FB1: **44.38** 311

Adding NER data to features.py, can be find in data/nerdata/all.txt

name data from: https://github.com/enorvelle/NameDatabases/tree/master/NamesDatabases

company data from https://en.wikipedia.org/wiki/List_of_companies_of_the_United_States

city and states from https://en.wikipedia.org/wiki/List_of_United_States_cities_by_population

processed 11570 tokens with 356 phrases; found: 307 phrases; correct: 148.

accuracy: 95.77%; precision: 48.21%; recall: 41.57%; FB1: 44.65

ENTITY: precision: **48.21%**; recall: **41.57%**; FB1: **44.65** 307