UDACITY

PROJECT SPECIFICATION

Part of Speech Tagging

General Requirements

CRITERIA	MEETS SPECIFICATIONS
Submission includes all files required for grading	• Includes `HMM Tagger.ipynb` displaying output for all executed cells • Includes `HMM Tagger.html`, which is an HTML copy of the notebook showing the output from executing all cells
Submitted files are complete and do not include any disallowed changes	Submitted notebook has made no changes to test case assertions

Baseline Tagger Implementation

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Student correctly implements the `pair_counts()` function	Emission count test case assertions all pass. • The emission counts dictionary has 12 keys, one for each of the tags in the universal tagset • "time" is the most common word tagged as a NOUN
Correct baseline MFC tagger implementation	Baseline MFC tagger passes all test case assertions and produces the expected accuracy using the universal tagset. • >95.5% accuracy on the training sentences • 93% accuracy the test sentences

Calculating Tag Counts

CRITERIA	MEETS SPECIFICATIONS
Correct `unigram_counts()` implementation	All unigram test case assertions pass
Correct `bigram_counts()` implementation	All bigram test case assertions pass

Correct CRITERIA `start_counts()` and `end_counts()` implementation	All start and end count test case assertions pass MEETS SPECIFICATIONS

Basic HMM Tagger Implementation

CRITERIA	MEETS SPECIFICATIONS
Correct HMM network construction	All model topology test case assertions pass
Correct basic HMM tagger implementation	Basic HMM tagger passes all assertion test cases and produces the expected accuracy using the universal tagset. • >97% accuracy on the training sentences • >95.5% accuracy the test sentences

Suggestions to Make Your Project Stand Out!

Students may run their taggers on more complex datasets (for example, the `nltk.corpus.brown` or `nltk.corpus.treebank` datasets).

Students may also try more advanced HMMs:

- Using pseudocounts or interpolated smoothing to handle missing data

- Retrain the hidden markov model using Baum-Welch re-estimation (available via the `.fit()` method in Pomegranate)