UDACITY

Return to "Natural Language Processing Nanodegree" in the classroom

Part of Speech Tagging

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations for passing HMM Tagger project 🎉

Stay Udacious!

General Requirements

- 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

Both notebook and HTML are included with submission.

Submitted notebook has made no changes to test case assertions

No changes are made in test case assertions.

Baseline Tagger Implementation

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**

Well Done! Emission counts dictionary has 12 keys and "time" is the most common word tagged as NOUN.

Suggestion: you could use `nested defaultdict` to avoid explicit initialization on the second key-value pair.

```
nested_dict = defaultdict(lambda: defaultdict(int))
for tag, word in zip(sequences_A, sequences_B):
     nested_dict[tag][word] += 1
return nested_dict
```

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**

MFC tagger accuracy looks good. 👍

Calculating Tag Counts

All unigram test case assertions pass

Your tag unigrams looks good.

You could try this implementation using `itertools.chain` as well:

```
def unigram_counts(sequences):
  return Counter(chain(*sequences))

tag_unigrams = unigram_counts(data.training_set.Y)
```

All bigram test case assertions pass

Your tag bigrams looks good.

You could try this implementation using `itertools.chain` as well:

```
 def bigram_counts(sequences):
  counts = Counter()
  counts.update(chain(*(zip(s[:-1], s[1:]) for s in sequences)))
  return counts

tag_bigrams = bigram_counts(data.training_set.Y)
```

All start and end count test case assertions pass

Well done! Starting and ending counts are correctly calculated and test case assertions are passing.

Basic HMM Tagger Implementation

All model topology test case assertions pass

All Model topology test case assertion are passing.

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**

Great! Accuracy on both training and testing data sets are above threshold.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review