Part of Speech Tagging

1. Description of viterbi implementation

Viterbi decoding is used to find best sequence from a possible list of sequences. We have been given emission score, transition score, start score and end score. I am maintaining 2 dimensional array 't' to maintain score of best sequence till current now. For example, t[i,] contains score of sequence ending with i-1. This structure need to implement following equation. Since we are going to operate on log values, we don't need multiply these scores, rather we will perform addition.

$$T(i, y) = \psi_x(y, i, x) + \max_{y'} \psi_t(y', y) + T(i-1, y')$$

first row in 't' matrix will be summation of start score and first row of emission score. Now we need to update subsequent rows of 't'. Each element of row can be calculated using previous row of 't' and current transitional score. We will add emission score of current element in it. This how we will update 't' element by element. We can get final score using last row of 't' and end scores.

To find out path giving this best score, we need to keep track of indexes giving the best scores. This is done by pathExtractPtrs variable. At each step, we will store index which is giving the best score so far. At the end, we will follow these indexes to find 'y'.

2. Description of the added features

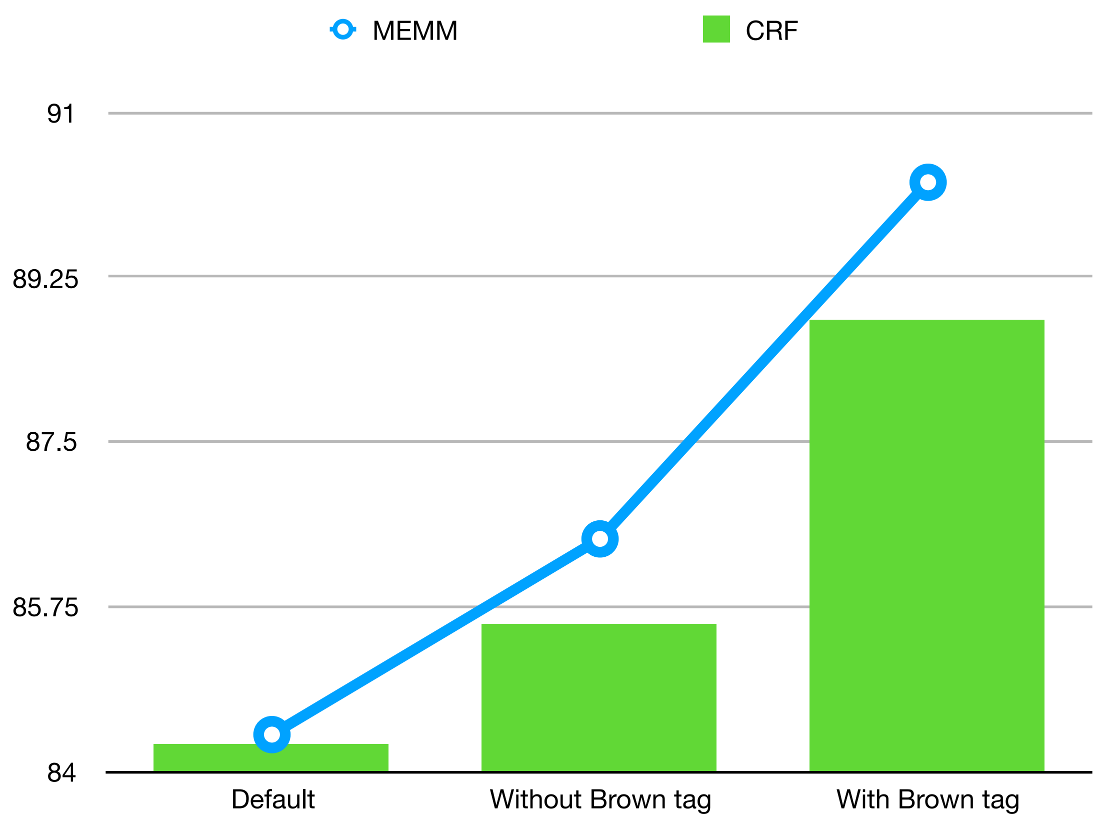
Feature	Motivation & Description	Example	Accuracy change
IS_FIRST_WORD	Generally sentences starts with some specific types of words. So if current word is first word of the sentence then it is quite possible that it's noun, verb or pronoun.	**He** is going to ace NLP exam. **Are** you going to the party? **USA** is a great country.	MEEM: 0.01% CRF:- 0.29%
IS_LAST_WORD	Just like first word of sentence, last word can also have some specific type of tag.	I am going to **library**. This is **dustbin**. [both ends with noun]	

Feature	Motivation & Description	Example	Accuracy change
IS_FIRST_CHAR_UPPER	First char of many words are capitalized. There are mainly two reasons for this, either it's first char of string or it's noun.	I am planning to **Google**. I studied at **SBU**. Both of above are noun.	MEEM: 0.25% (84.64%) CRF:0.05% (84.34%)
IS_CNT_1, IS_CNT_2, IS_CNT_3…..	How frequently word occurs in the sentence can be useful feature to type of that word.	Dog chase cat. Cat chase rat. Rat chase insect. Chase:3 Dog:2	MEEM:-0.04% (84.30%) CRF:-0.15% (84.15%)
IS_POST_……	suffix of word can be helpful a lot to identify type of tag. Postfixes like 's', 'es', 'ly', 'ing', 'ed' etc can be helpful a lot	I was **running** at that time. **Surprisingly**, he **studied** for 12 hours.	MEEM: 2.2% (86.61%) CRF: 1.13% (85.43%)
IS_PRE_……	Just like suffix, prefix can also be helpful.	It was an **unsuccessful** attempt. **What** is timing of the event and **where** is it?	MEEM: 0.47% (84.86%) CRF: 0.34% (83.96%)
IS_CLUSTERBIT0, IS_CLUSTERBIT1, IS_CLUSTERBIT1 …..	Brown clustering divides words into separate clusters. Based on cluster we can classify words in separate part of speech.		Didn't include this in the final code since it is taking too much time.
IS_T_LBL …	We can use already known labels as features. Is we create a separate feature for frequent words then it can improve accuracy substantially.	He(pronoun) is(v) driving(v).	MEEM: 4.1% (89.45%) CRF:3.83% (88.13%)
IS_STOPWORD	Stopwords like A,an,the, i, me ect are of specific types. Generally they are not adj, adverb, verb etc.	**I am** stopword.	MEEM: 0.0% (84.34%) CRF: 0.42% (84.72%)

3. Comparison of the my features against the basic features

I have added features mentioned above. I have excluded brown clustering since it's taking really long time.

I have used frequent words brown label as a feature. so, I will mention accuracy with and without that feature.



As we can see there is around 2% accuracy increment on dev dataset with out brown dataset tag and around 5% increment with the help of brown tag.

Here is an example of sentence which was not tagged properly just using basic features. Here words like 'dumb', 'things' were classified incorrectly with default features but with new features they were classified correctly.

```
Sentence:
[u'RT', u'@rockerfuckerak', u':', u'What', u"'s", u'done', u'is',
u'done', u',', u'Just', u'leave', u'it', u'alone', u',', u'and',
u'do', u"n't", u'regret', u'.', u'Sometimes', u'somethings', u'turn',
```

```
u'in-to', u'dumb', u'things', u'.', u'And', u'that', u"'s", u'when',
u'...']

True Labels
['X', 'X', '.', 'PRON', 'VERB', 'VERB', 'VERB', 'VERB', '.', 'ADV',
'VERB', 'PRON', 'ADV', '.', 'CONJ', 'VERB', 'ADV', 'VERB', '.', 'ADV',
'NOUN', 'VERB', 'ADP', 'ADJ', 'NOUN', '.', 'CONJ', 'DET', 'VERB',
'ADV', '.']

default feature result
['X', 'X', '.', 'PRON', 'VERB', 'NOUN', 'VERB', 'VERB', '.',
'ADV','VERB', 'PRON', 'VERB', '.', 'CONJ', 'VERB', 'ADV', 'VERB',
'.','VERB', 'NOUN', 'NOUN', 'X', 'VERB', 'NOUN', '.', 'CONJ',
'DET','VERB', 'ADV', '.']

With My features
['X', 'X', '.', 'PRON', 'VERB', 'NOUN', 'VERB', 'VERB', '.',
'ADV','VERB', 'PRON', 'ADV', '.', 'CONJ', 'VERB', 'ADV', 'VERB',
'.','NOUN', 'NOUN', 'NOUN', 'ADP', 'ADJ', 'NOUN', '.', 'CONJ',
'DET','VERB', 'ADV', '.']
```

4. Comparison of MEMM and CRFs

MEMMs compute the probability of the next state given the current state and the observation and CRF computes all state transitions globally, in a single model. So, CRF helps to overcome label bias problem.

Here is performance comparison between these models on dev dataset.

	Without Brown tag	With Brown tag
MEMM	86.47	90.26
CRF	85.57	88.79

CRF is expected to perform better than MEMM since it's performance is not affected by cable bias problem. But in this implementation, MEMM is having higher accuracy than CRF. There several possible reason for such behavior.

One reason of low accuracy for CRF is type of data we are processing. We are testing our results on twitter dataset and in twitter, all tweets are independent of each other and if we interpret data globally then context from other tweets might not be same to current one. So, it will not help in improving accuracy. It seems after adding so many features, it over-complicates the model and it becomes difficult to learn parameters.

Here is much closer look of performance for both models.

Model	Perfromance
MEMM (using frequent brown tags)	Token-wise accuracy 90.35004730368968 Token-wise F1 (macro) 89.3468288623874 Token-wise F1 (micro) 90.35004730368968 Sentence-wise accuracy 25.892857142857146
CRF (using frequent brown tags)	Token-wise accuracy 88.0321665089877 Token-wise F1 (macro) 87.01935836521488 Token-wise F1 (micro) 88.0321665089877 Sentence-wise accuracy 18.75
MEMM (without frequent brown tags)	Token-wise accuracy 86.47114474929045 Token-wise F1 (macro) 85.13684428658034 Token-wise F1 (micro) 86.47114474929045 Sentence-wise accuracy 16.071428571428573
MEMM (without frequent brown tags)	Token-wise accuracy 85.57237464522233 Token-wise F1 (macro) 84.47217261150757 Token-wise F1 (micro) 85.57237464522233 Sentence-wise accuracy 11.607142857142858

Let's take a look of sentence for which CRF is better than MEMM.

Here, the tag emboldened are wrongly predicted. We can notice that 5 of MEMM are predicted incorrectly and 2 of CRF are predicted incorrectly. If sentence has contextual connection with previous words then CRF is performing better than MEMM.

```
Sentence:
[u'@sarahk47', u'Okay', u'The', u'doc', u"'s", u'saying', u'Keep',
u'it', u'tamped', u'down', u'a', u'coupla', u'days', u'then',
u'bring', u'her', u'out', u'early', u'?']


True Labels
 ['X', 'X', 'DET', 'NOUN', 'VERB', 'VERB', 'VERB', 'PRON', 'VERB',
'PRT', 'DET', 'NOUN', 'NOUN', 'ADP', 'VERB', 'PRON', 'ADP', 'ADV',
'.']

CRF Predicted labels
```

```
array(['X', 'X', 'DET', 'NOUN', 'VERB', 'VERB', 'VERB', 'PRON',
'VERB','ADP', 'DET', 'NOUN', 'NOUN', 'ADP', 'VERB', 'PRON', 'PRT',
'ADV','.']


MEMM Predicted labels
['X', 'VERB', 'DET', 'NOUN', 'PRT', 'VERB', 'VERB', 'PRON', 'VERB',
       'ADP', 'DET', 'NOUN', 'NOUN', 'ADV', 'VERB', 'PRON', 'PRT',
'ADV','.']
```

Let's take a look of sentence for which MEMM is performed better than CRF.

```
Sentence:
[u'Dollar', u'steady', u'versus', u'yen', u'but', u'market', u'wary',
u'of', u'intervention', u'|', u'Money', u'...:', u'The', u'dollar',
u'held', u'near', u'its', u'highest', u'in', u'a', u'month', u'...',
u'http://bit.ly/a0F3dO']

True Labels
['NOUN', 'ADV', 'CONJ', 'NOUN', 'CONJ', 'NOUN', 'ADJ', 'ADP', 'NOUN',
'.', 'NOUN', '.', 'DET', 'NOUN', 'VERB', 'ADP', 'PRON', 'ADJ', 'ADP',
'DET', 'NOUN', '.', 'X']

CRF Predicted labels
['NOUN', 'NOUN', 'VERB', 'PRT', 'CONJ', 'VERB', 'ADV', 'ADP','NOUN',
'.', 'NOUN', '.', 'DET', 'NOUN', 'NOUN', 'NOUN','PRON', 'ADJ', 'ADP',
'DET', 'NOUN', '.', 'X']


MEMM Predicted labels
['NOUN', 'NOUN', 'VERB', 'NOUN', 'CONJ', 'VERB', 'NOUN', 'ADP','NOUN',
'.', 'NOUN', '.', 'DET', 'NOUN', 'VERB', 'NOUN', 'PRON', 'ADJ', 'ADP',
'DET', 'NOUN', '.', 'X']
```

5. References

- Frequent words labels from NLTK brown tag
- Brown clustering data code from https://github.com/percyliang/brown-cluster
- Viterbi algorithms explanation http://www.davidsbatista.net/blog/2017/11/11/HHM_and_Naive_Bayes/, https://arnoutdevos.github.io/Viterbi-Pseudocode/