STONY BROOK UNIVERSITY

CSE 538 : Assignment 2 (Part of Speech Tagging)

Tejas Naik (111425071)

1. Viterbi Implementation

I implemented the following steps for Viterbi decoding:

1) Made a numpy matrix of zeros of size LxN where L is the number of tags and N is the number of words.
2) Also made indices matrix initalized with -1s with the same dimensions. This matrix is to keep track of the maximum index for each location. This comes into use when we want to track the best sequence.
3) After utilizing the first score list given to us to initialize the first column, then for each location in the matrix, I found the best possible score for that location by finding the maximum of the sum of the previous column score, the emission from that label to that particular word and the transmission score.
4) Finally, by taking the index of the maximum score after adding the end scores, I traced back the entire label of sequence which is the optimal one.

2. Added Features

2.1 Preprocessing:

a) Brown Clustering:

Brown clustering is a hierarchical agglomerative clustering algorithm which is used to group words which are semantically similar. The output of this algorithm is a binary string consisting of 1's and 0's. This string helps in grouping of the words into clusters. Given a value of k bits, we can generate 2^k clusters.

I tried out different number of clusters. I find that increasing k makes the clusters really sparse, and decreasing k makes non-similar words to be grouped together too. Increase in number of features leads to overfitting as well and affects the accuracy. I have tried to find a tradeoff between these two:

K = 2 (Number of clusters = 4)
Dev evaluation
Token-wise accuracy 85.09933774834437
Token-wise F1 (macro) 84.3503554259091

Token-wise F1 (micro) 85.09933774834437
Sentence-wise accuracy 13.392857142857142

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254
ADJ	0.56	0.51	0.53	99
ADP	0.87	0.87	0.87	151
ADV	0.82	0.69	0.75	129
CONJ		0.98	0.95	0.96	42
DET	0.98	0.92	0.95	130
NOUN		0.77	0.86	0.81	479
NUM	0.78	0.74	0.76	34
PRON		0.97	0.92	0.95	194
PRT	0.89	0.89	0.89	57
VERB		0.83	0.85	0.84	362
X	0.87	0.80	0.83	183
micro avg	0.85	0.85	0.85	2114
macro avg	0.86	0.83	0.84	2114
weighted avg		0.85	0.85	0.85	2114

K = 5 (Number of clusters = 32)

Dev evaluation
Token-wise accuracy 85.43046357615894
Token-wise F1 (macro) 84.39667246885465
Token-wise F1 (micro) 85.43046357615893
Sentence-wise accuracy 12.5

	precision	recall	f1-score	support
.	0.97	0.98	0.97	254
ADJ	0.64	0.53	0.58	99
ADP	0.88	0.86	0.87	151
ADV	0.84	0.67	0.75	129
CONJ		0.97	0.93	0.95	42
DET	0.95	0.93	0.94	130
NOUN		0.76	0.88	0.82	479
NUM	0.77	0.71	0.74	34
PRON		0.97	0.93	0.95	194
PRT	0.88	0.89	0.89	57
VERB		0.86	0.83	0.85	362
X	0.85	0.80	0.83	183

micro avg	0.85	0.85	0.85	2114
macro avg	0.86	0.83	0.84	2114
weighted avg		0.86	0.85	0.85	2114

K = 7 (Number of clusters = 128)

Dev evaluation
Token-wise accuracy 85.00473036896878
Token-wise F1 (macro) 83.97934012906546
Token-wise F1 (micro) 85.00473036896878
Sentence-wise accuracy 11.607142857142858

	precision	recall	f1-score	support
.	0.94	0.98	0.96	254
ADJ	0.61	0.55	0.57	99
ADP	0.84	0.88	0.86	151
ADV	0.85	0.68	0.76	129
CONJ		0.87	0.95	0.91	42
DET	0.94	0.92	0.93	130
NOUN		0.79	0.84	0.81	479
NUM	0.83	0.71	0.76	34
PRON		0.95	0.93	0.94	194
PRT	0.88	0.89	0.89	57
VERB		0.84	0.84	0.84	362
X	0.85	0.85	0.85	183
micro avg	0.85	0.85	0.85	2114
macro avg	0.85	0.83	0.84	2114
weighted avg		0.85	0.85	0.85	2114

I observed that K=5 i.e. 32 clusters works the best because probably K=7 creates sparse clusters and K=2 creates very few clusters to map the similarity between the words. Hence I went with K=5.

But I also observed, that without using Brown Clustering, my token wise accuracy tends to be higher. This might be happening because Brown Clustering is unable to cluster the words according to their POS tags.

Output without clustering (after applying all the features mentioned below):

Dev evaluation
Token-wise accuracy 86.09271523178808
Token-wise F1 (macro) 85.48562328057541
Token-wise F1 (micro) 86.09271523178808

Sentence-wise accuracy 12.5

	precision	recall	f1-score	support
.	0.95	0.99	0.97	254
ADJ	0.69	0.56	0.61	99
ADP	0.85	0.89	0.87	151
ADV	0.90	0.67	0.76	129
CONJ		0.95	0.93	0.94 42
DET	0.96	0.92	0.94	130
NOUN		0.79	0.87	0.83 479
NUM	0.84	0.76	0.80	34
PRON		0.95	0.94	0.95 194
PRT	0.88	0.91	0.90	57
VERB		0.83	0.85	0.84 362
X	0.87	0.83	0.85	183
micro avg	0.86	0.86	0.86	2114
macro avg	0.87	0.84	0.85	2114
weighted avg		0.86	0.86	0.86 2114

2.2 Feature Engineering:

Here, I try to add each feature to my baseline model and compare the token wise accuracies of my model with basic features and my model with each added feature.

2.2.1 Suffix as a feature

a) **Verb Suffixes**

I analyzed the training data for verbs, and made a simple dictionary to maintain a count of last three letter or last two letter suffixes. I then printed the sorted version of the dictionary which helped me to get an idea of which suffixes are most common for verbs. This is the output of the verb dictionary :

The format of the following dictionary is **<verb_suffix, number of occurrences>**

(u'ing', 134) (u'is', 116) (u'ed', 86) (u've', 71) (u'be', 63) (u"'s", 60) (u'll', 45) (u'et', 41) (u"'m", 38) (u'ave', 36) (u'as', 35) (u'do', 32) (u'go', 32) (u're', 31) (u'nt', 30) (u'ill', 25) (u'ay', 24) (u'was', 24) (u'get', 23) (u'es', 22) (u'ow', 20) (u'are', 19) (u'in', 18) (u'an', 17) (u'en', 17) (u'got', 17) and so on.

Hence, I have added a verb suffix feature if it ends in ing/ify/ed/ill as I checked that the rest of the suffixes were causing other non-verb tokens to be classified as token as well.

The accuracy on dev set using CRF model before and after adding this feature is as follows:

Before applying verb suffix feature:

Dev evaluation
Token-wise accuracy 84.29517502365185
Token-wise F1 (macro) 83.21108699638205
Token-wise F1 (micro) 84.29517502365185
Sentence-wise accuracy 11.607142857142858

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254
ADJ	0.64	0.55	0.59	99
ADP	0.86	0.87	0.87	151
ADV	0.83	0.62	0.71	129
CONJ		0.95	0.93	0.94	42
DET	0.96	0.91	0.93	130
NOUN		0.79	0.86	0.82	479
NUM	0.85	0.68	0.75	34
PRON		0.99	0.93	0.96	194
PRT	0.84	0.84	0.84	57
VERB		0.79	0.84	0.82	362
X	0.80	0.78	0.79	183
micro avg	0.84	0.84	0.84	2114
macro avg	0.85	0.82	0.83	2114
weighted avg		0.84	0.84	0.84	2114

After applying verb suffix feature:

Dev evaluation
Token-wise accuracy 85.19394512771996
Token-wise F1 (macro) 83.89856926707404
Token-wise F1 (micro) 85.19394512771996
Sentence-wise accuracy 13.392857142857142

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254
ADJ	0.63	0.49	0.55	99
ADP	0.85	0.88	0.87	151

ADV	0.82	0.60	0.70	129	
CONJ		1.00	0.93	0.96	42
DET	0.95	0.91	0.93	130	
NOUN		0.80	0.87	0.83	479
NUM	0.81	0.74	0.77	34	
PRON		0.97	0.94	0.96	194
PRT	0.87	0.91	0.89	57	
VERB		0.84	0.86	0.85	362
X	0.79	0.80	0.80	183	
micro avg	0.85	0.85	0.85	2114	
macro avg	0.86	0.83	0.84	2114	
weighted avg		0.85	0.85	0.85	2114

INFERENCE: We observe the sentence wise accuracy increases by approx 2%, the verb precision, recall, f1 score increases from around 0.79 to 0.83-0.85.

b) Adjective Suffixes

Similar to the verbs, I also analyzed the training data for adjectives, and made a simple dictionary to maintain a count of last three letter or last two letter suffixes. This is the output of the adjective suffix dictionary:

(u'st', 33) (u'er', 15) (u'ext', 12)
The accuracy on dev set using CRF model before and after adding this feature is as follows:

Before applying adj suffix feature

Dev evaluation
Token-wise accuracy 84.29517502365185
Token-wise F1 (macro) 83.21108699638205
Token-wise F1 (micro) 84.29517502365185
Sentence-wise accuracy 11.607142857142858

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254
ADJ	0.64	0.55	0.59	99
ADP	0.86	0.87	0.87	151
ADV	0.83	0.62	0.71	129
CONJ		0.95	0.93	0.94	42
DET	0.96	0.91	0.93	130

NOUN		0.79	0.86	0.82	479
NUM	0.85	0.68	0.75	34	
PRON		0.99	0.93	0.96	194
PRT	0.84	0.84	0.84	57	
VERB		0.79	0.84	0.82	362
X	0.80	0.78	0.79	183	
micro avg	0.84	0.84	0.84	2114	
macro avg	0.85	0.82	0.83	2114	
weighted avg		0.84	0.84	0.84	2114

After applying adj suffix feature

Dev evaluation
Token-wise accuracy 84.10596026490066
Token-wise F1 (macro) 83.3104900285992
Token-wise F1 (micro) 84.10596026490065
Sentence-wise accuracy 9.821428571428571

	precision	recall	f1-score	support
.	0.95	0.99	0.97	254
ADJ	0.59	0.53	0.56	99
ADP	0.85	0.87	0.86	151
ADV	0.85	0.58	0.69	129
CONJ		1.00	0.90	0.95	42
DET	0.98	0.91	0.94	130
NOUN		0.78	0.85	0.81	479
NUM	0.83	0.71	0.76	34
PRON		0.97	0.94	0.96	194
PRT	0.84	0.95	0.89	57
VERB		0.79	0.84	0.81	362
X	0.80	0.78	0.79	183
micro avg	0.84	0.84	0.84	2114
macro avg	0.85	0.82	0.83	2114
weighted avg		0.84	0.84	0.84	2114

INFERENCE: Since the token accuracy decreases, I observed that this is not that good a feature.

c) Adverb Suffixes

Similar to the analysis I did before, I added an adverb suffix feature, “HAS_ADV_SUFFIX”, which evaluates to True if it contains an suffix in ["ly","ard","en","ow","re","n't","lly"], else to False.

The accuracy on dev set using CRF model before and after adding this feature is as follows:

Before applying adv suffix feature:

Dev evaluation
Token-wise accuracy 84.29517502365185
Token-wise F1 (macro) 83.21108699638205
Token-wise F1 (micro) 84.29517502365185
Sentence-wise accuracy 11.607142857142858

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254
ADJ	0.64	0.55	0.59	99
ADP	0.86	0.87	0.87	151
ADV	0.83	0.62	0.71	129
CONJ		0.95	0.93	0.94	42
DET	0.96	0.91	0.93	130
NOUN		0.79	0.86	0.82	479
NUM	0.85	0.68	0.75	34
PRON		0.99	0.93	0.96	194
PRT	0.84	0.84	0.84	57
VERB		0.79	0.84	0.82	362
X	0.80	0.78	0.79	183
micro avg	0.84	0.84	0.84	2114
macro avg	0.85	0.82	0.83	2114
weighted avg		0.84	0.84	0.84	2114

After adding adv suffix feature:

Dev evaluation
Token-wise accuracy 85.71428571428571
Token-wise F1 (macro) 85.17768066926673
Token-wise F1 (micro) 85.71428571428571
Sentence-wise accuracy 13.392857142857142

	precision	recall	f1-score	support
.	0.94	0.98	0.96	254
ADJ	0.67	0.52	0.58	99
ADP	0.88	0.89	0.88	151

ADV	0.83	0.67	0.74	129	
CONJ		1.00	0.95	0.98	42
DET	0.98	0.92	0.95	130	
NOUN		0.79	0.86	0.83	479
NUM	0.83	0.74	0.78	34	
PRON		0.97	0.94	0.96	194
PRT	0.90	0.93	0.91	57	
VERB		0.84	0.86	0.85	362
X	0.79	0.81	0.80	183	
micro avg	0.86	0.86	0.86	2114	
macro avg	0.87	0.84	0.85	2114	
weighted avg		0.86	0.86	0.86	2114

INFERENCE: We observe the sentence wise accuracy increases by approx 2%, the adverb precision, recall, f1 score increases from around 0.62 to 0.67.

2.2.2 Hashtag as a feature

Since the data under consideration is Twitter data, presumably, there will be many hashtags and direct addresses, which are classified into the X label class. Hence, I checked if the first letter is either a '#' or a '@', and if it is, the feature "X_class" evaluates to true, else false.

The results on the dev set using CRF model before adding this feature are:

Dev evaluation
Token-wise accuracy 84.29517502365185
Token-wise F1 (macro) 83.21108699638205
Token-wise F1 (micro) 84.29517502365185
Sentence-wise accuracy 11.607142857142858

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254
ADJ	0.64	0.55	0.59	99
ADP	0.86	0.87	0.87	151
ADV	0.83	0.62	0.71	129
CONJ		0.95	0.93	0.94	42
DET	0.96	0.91	0.93	130
NOUN		0.79	0.86	0.82	479
NUM	0.85	0.68	0.75	34
PRON		0.99	0.93	0.96	194
PRT	0.84	0.84	0.84	57
VERB		0.79	0.84	0.82	362

X	0.80	0.78	0.79	183
micro avg	0.84	0.84	0.84	2114
macro avg	0.85	0.82	0.83	2114
weighted avg		0.84	0.84	0.84	2114

The results on the dev set using CRF model after adding this feature are:

Dev evaluation
Token-wise accuracy 84.72090823084201
Token-wise F1 (macro) 84.24401500566904
Token-wise F1 (micro) 84.72090823084201
Sentence-wise accuracy 12.5

	precision	recall	f1-score	support
.	0.95	0.99	0.97	254
ADJ	0.59	0.52	0.55	99
ADP	0.89	0.87	0.88	151
ADV	0.82	0.60	0.69	129
CONJ		1.00	0.95	0.98	42
DET	0.98	0.92	0.94	130
NOUN		0.79	0.84	0.82	479
NUM	0.86	0.74	0.79	34
PRON		0.96	0.94	0.95	194
PRT	0.85	0.93	0.89	57
VERB		0.79	0.85	0.82	362
X	0.84	0.83	0.84	183
micro avg	0.85	0.85	0.85	2114
macro avg	0.86	0.83	0.84	2114
weighted avg		0.85	0.85	0.85	2114

INFERENCE: The recall, precision increase from 0.79 to 0.83-0.84.

3. **Comparison of my features with basic features**

Features Added :

1. Suffix Features: Verbs
2. Suffix Features: Adverbs

3. Suffix Features: Adjectives:
4. Prefix Features: Noun
5. Prefix Features: HashTag or Direct address

(I have not added Brown Clustering as it is having a negative impact on the accuracy).

Adding these features give a result of (using CRF model) :

Dev evaluation
Token-wise accuracy 86.09271523178808
Token-wise F1 (macro) 85.48562328057541
Token-wise F1 (micro) 86.09271523178808
Sentence-wise accuracy 12.5

	precision	recall	f1-score	support
.	0.95	0.99	0.97	254
ADJ	0.69	0.56	0.61	99
ADP	0.85	0.89	0.87	151
ADV	0.90	0.67	0.76	129
CONJ		0.95	0.93	0.94	42
DET	0.96	0.92	0.94	130
NOUN		0.79	0.87	0.83	479
NUM	0.84	0.76	0.80	34
PRON		0.95	0.94	0.95	194
PRT	0.88	0.91	0.90	57
VERB		0.83	0.85	0.84	362
X	0.87	0.83	0.85	183
micro avg	0.86	0.86	0.86	2114
macro avg	0.87	0.84	0.85	2114
weighted avg		0.86	0.86	0.86	2114

Without adding these features, with only the basic features, I get an output of (using CRF model) :

Dev evaluation
Token-wise accuracy 84.29517502365185
Token-wise F1 (macro) 83.21108699638205
Token-wise F1 (micro) 84.29517502365185
Sentence-wise accuracy 11.607142857142858

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254

ADJ	0.64	0.55	0.59	99
ADP	0.86	0.87	0.87	151
ADV	0.83	0.62	0.71	129
CONJ		0.95	0.93	0.94	42
DET	0.96	0.91	0.93	130
NOUN		0.79	0.86	0.82	479
NUM	0.85	0.68	0.75	34
PRON		0.99	0.93	0.96	194
PRT	0.84	0.84	0.84	57
VERB		0.79	0.84	0.82	362
X	0.80	0.78	0.79	183
micro avg	0.84	0.84	0.84	2114
macro avg	0.85	0.82	0.83	2114
weighted avg		0.84	0.84	0.84	2114

Hence I observe that the adding my features helped to boost my token wise accuracy from 84.29 to 86.09 and the sentence wise accuracy from 11.6 to 12.5.

4. Comparison of MEMM and CRFs

Basic Feature model on LR(MEMM):

Dev evaluation
Token-wise accuracy 84.38978240302744
Token-wise F1 (macro) 83.33422799705717
Token-wise F1 (micro) 84.38978240302745
Sentence-wise accuracy 8.928571428571429

	precision	recall	f1-score	support
.	0.94	0.98	0.96	254
ADJ	0.73	0.36	0.49	99
ADP	0.92	0.88	0.90	151
ADV	0.94	0.59	0.72	129
CONJ		1.00	0.93	0.96	42
DET	0.99	0.92	0.95	130
NOUN		0.73	0.90	0.80	479
NUM	0.85	0.68	0.75	34
PRON		0.99	0.92	0.96	194
PRT	0.89	0.88	0.88	57
VERB		0.80	0.85	0.82	362

X	0.81	0.77	0.79	183
micro avg	0.84	0.84	0.84	2114
macro avg	0.88	0.80	0.83	2114
weighted avg		0.85	0.84	0.84	2114

Basic Feature model on CRF:

Dev evaluation
Token-wise accuracy 84.29517502365185
Token-wise F1 (macro) 83.21108699638205
Token-wise F1 (micro) 84.29517502365185
Sentence-wise accuracy 11.607142857142858

	precision	recall	f1-score	support
.	0.95	0.98	0.97	254
ADJ	0.64	0.55	0.59	99
ADP	0.86	0.87	0.87	151
ADV	0.83	0.62	0.71	129
CONJ		0.95	0.93	0.94	42
DET	0.96	0.91	0.93	130
NOUN		0.79	0.86	0.82	479
NUM	0.85	0.68	0.75	34
PRON		0.99	0.93	0.96	194
PRT	0.84	0.84	0.84	57
VERB		0.79	0.84	0.82	362
X	0.80	0.78	0.79	183
micro avg	0.84	0.84	0.84	2114
macro avg	0.85	0.82	0.83	2114
weighted avg		0.84	0.84	0.84	2114

Enhanced Feature model on LR(MEMM):

Dev evaluation
Token-wise accuracy 85.71428571428571
Token-wise F1 (macro) 85.17929223770858
Token-wise F1 (micro) 85.71428571428571
Sentence-wise accuracy 12.5

	precision	recall	f1-score	support

.	0.94	0.99	0.97	254
ADJ	0.77	0.43	0.55	99
ADP	0.93	0.88	0.90	151
ADV	0.90	0.67	0.77	129
CONJ	1.00	0.93	0.96	42
DET	0.99	0.92	0.95	130
NOUN	0.73	0.90	0.81	479
NUM	0.88	0.68	0.77	34
PRON	0.99	0.93	0.96	194
PRT	0.89	0.89	0.89	57
VERB	0.83	0.84	0.84	362
X	0.89	0.81	0.85	183
micro avg	0.86	0.86	0.86	2114
macro avg	0.90	0.82	0.85	2114
weighted avg	0.86	0.86	0.85	2114

Enhanced Feature model on CRF:

Dev evaluation
Token-wise accuracy 86.09271523178808
Token-wise F1 (macro) 85.48562328057541
Token-wise F1 (micro) 86.09271523178808
Sentence-wise accuracy 12.5

	precision	recall	f1-score	support
.	0.95	0.99	0.97	254
ADJ	0.69	0.56	0.61	99
ADP	0.85	0.89	0.87	151
ADV	0.90	0.67	0.76	129
CONJ	0.95	0.93	0.94	42
DET	0.96	0.92	0.94	130
NOUN	0.79	0.87	0.83	479
NUM	0.84	0.76	0.80	34
PRON	0.95	0.94	0.95	194
PRT	0.88	0.91	0.90	57
VERB	0.83	0.85	0.84	362
X	0.87	0.83	0.85	183
micro avg	0.86	0.86	0.86	2114
macro avg	0.87	0.84	0.85	2114
weighted avg	0.86	0.86	0.86	2114

Thus, CRF performs much better than LR using enhanced features.