CSE538 Natural Language Processing Fall 18

Prof. Niranjan Balasubramanian
Stony Brook University

Assignment 2
19^{th} October, 2018

Name - Rushikesh Nalla
ID - 111971011

1 Viterbi Implementation -

Approach -

- Viterbi is a general dynamic programming algorithm. Implementation involves dividing the whole problem into various sub problems.
- Solving such sub-problems and storing the results which can be used in further steps. This reduces the number of computations as we can make use of previously stored values (computed sub-problems).

Implementation -

- I have taken the transpose of the emission scores so that each row corresponds to a tag and each column to a word. (For easy implementation as done in class).
- Used two matrices of $L \times N$ dimension where L is number of tags and N is number of words for maintaining the scores and the tags.
- Initialized the first column of the matrix with sum of emission and starting scores.
- For each entry in the second column (word) we take the max of (scores of prev column + transition probability from previous to current tag) and add to it corresponding emission score.

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

  where T(i, y) is the score of the best sequence from 1 to i such that $y_i = y$,
  $\psi_x(y, i, x)$ is the emission probability of word x given tag y,
  $\psi_t(y', y)$ is the transition probability from y' to y,
  $T(i - 1, y')$ is the score of the best sequence from 1 to i-1 such that $y_{i-1} = y'$.
- Using this procedure all the columns are computed and results are stored in scores and tags matrices.
- We add the end scores to the final column and find the final tag position.
- We take the help of tags matrix so that we can backtrack and find the best scoring tag sequence.

2 Feature Engineering

Features added -

Below is a table containing the additional feature's name, description and some examples.

Feature name	Feature Description	Example word
IS_URL	To check if the word is a url	amazon.com, http://care.org
IS_HASHTAG	To check if the word starts with #	#nlp
IS_MENTION	To check if the word starts with @	@rushikesh
IS_PRONOUN	To check if the first letter of the word is uppercase	He, her
IS_PLURAL	To check if the last letter of the word is 's'	days
IS_PUNCTUATION	To check if the word is a punctuation	, ! ;
IS_ADVERB	To check if the word ends with "ly"	happily
IS_VERB	To check if the word ends with "ed" or "ing"	committed, committing
IS_ADJECTIVE	To check if the word starts with "un" or ends with "st"	uneasy latest
IS_HYPHEN	To check if the word contains '-'	data-set
PREFIX	Adding the prefix of the word as a feature	word = understand prefix = "un"
SUFFIX	Adding the suffix of the word as a feature	word = buyer suffix = "er"
STEMMED	Adding the stem of the word as a feature	word = studies stem = studi
LEMMATIZED	Adding the lemma of the word as a feature	word = studies lemma = study
CLUSTER	Adding the cluster number of the word as a feature	word = love cluster = 111111011100

Features not added but explored -

- Named Entity Recognition using nltk and Stanford NER tagger. This improved the accuracy slightly as well as number of features.
- Used a slang word dictionary containing mapping between the slang and actual word. This didn't improve the accuracy much.

- Adding previous 2 words and next 2 words features instead of just 1. The improvement in accuracy is negligible as compared to increase in number of features.
- Adding some more conditions for identifying conjunctions, prepositions etc.

3 Performance of new features over basic features -

Feature Sets -

I have divided the additional features into various sets and here is the description of what each set represents and captures -

- **Basic -** These are the features that are already given and they capture any word's basic properties.
- **Set 1 -** As we are using tweets, there might be URL's, hastags(#) and at signs(@).
- **Set 2 -** Used some of the common ways (by looking at it's suffix) in which we can identify a word's part of speech.
- **Set 3 -** Prefixes and Suffixes of the word are very helpful in identifying a word's part of speech.
- **Set 4 -** Doing stemming and lemmatization to remove the inflectional endings and use the base word as a feature. This helps in grouping of similar words.
- **Set 5 -** Assigning the same cluser number to words that are semantically related and appear in similar contexts.

Set Name	**Constituent Features**
Basic Features	SENT_BEGIN, SENT_END, IS_ALNUM, IS_NUMERIC, IS_UPPER, IS_LOWER, IS_DIGIT, PREV_, NEXT_
Set 1	IS_URL, IS_HASHTAG, IS_MENTION
Set 2	IS_PUNCTUATION, IS_ADVERB, IS_VERB, IS_ADJECTIVE, IS_HYPHEN, IS_PLURAL
Set 3	PREFIX, SUFFIX
Set 4	STEMMED, LEMMATIZED
Set 5	CLUSTER

Results -

Here is the table containing the features that I have used and their corresponding accuracy. We can see that as we keep adding new features, the model begins to perform better. The difference between the models is seen on doing stemming, lemmatization and brown clustering which is discussed in next part.

Feature Set	LR Accuracy	CRF Accuracy
Basic Features	F1 - 84.389 F1 macro - 83.334 F1 micro - 84.389 Sentence - 8.928	F1 - 84.295 F1 macro - 83.211 F1 micro - 84.295 Sentence - 11.607
Basic Features + Set 1	F1 - 84.720 F1 macro - 83.430 F1 micro - 84.720 Sentence - 11.607	F1 - 84.626 F1 macro - 83.583 F1 micro - 84.626 Sentence - 11.607
Basic Features + Set 1 + Set 2	F1 - 85.808 F1 macro - 84.667 F1 micro - 85.808 Sentence - 16.071	F1 - 85.572 F1 macro - 84.019 F1 micro - 85.572 Sentence - 12.5
Basic Features + Set 1 + Set 2 + Set 3	F1 - 86.802 F1 macro - 85.412 F1 micro - 86.802 Sentence - 17.857	F1 - 86.518 F1 macro - 85.637 F1 micro - 86.518 Sentence - 16.071
Basic Features + Set 1 + Set 2 + Set 3 + Set 4	F1 - 87.322 F1 macro - 85.867 F1 micro - 87.322 Sentence - 16.071	F1 - 86.565 F1 macro - 84.729 F1 micro - 86.565 Sentence - 15.178
Basic Features + Set 1 + Set 2 + Set 3 + Set 5	F1 - 86.423 F1 macro - 85.446 F1 micro - 86.423 Sentence - 15.178	F1 - 86.707 F1 macro - 85.515 F1 micro - 86.707 Sentence - 14.285
Basic Features + Set 1 + Set 2 + Set 3 + Set 4 + Set 5	F1 - 87.464 F1 macro - 86.123 F1 micro - 87.464 Sentence - 16.071	F1 - 86.329 F1 macro - 84.811 F1 micro - 86.329 Sentence - 13.392

4 Comparision of MEMM and CRF -

Features used (Case 1) -

- **Features used -** Basic Features + Set 1 + Set 2 + Set 3 + Set 4 + Set 5
- The total number of features were 37000. Stemming and Lemmatization add a lot of features (13000 to be precise).
- Though LR gives a higher overall accuracy, if we look at the individual parts of speech (f1 scores), CR does better in most of them. Refer to figures 1 and 2.
- Such a large increase in number of features for a small increment in accuracy is not efficient.
- So, in the next case we evaluate the models without stemming and lemmatization.

```
D:\PycharmProjects\NLP Assignment 2 new>perl conlleval.pl -r -d \t < ./predictions/twitter_dev.lr.pred
processed 2114 tokens with 2114 phrases; found: 2114 phrases; correct: 1849.
accuracy:  87.46%; precision:  87.46%; recall:  87.46%; FB1:  87.46
                .: precision:  96.20%; recall:  99.61%; FB1:  97.87  263
              ADJ: precision:  73.02%; recall:  46.46%; FB1:  56.79  63
              ADP: precision:  92.00%; recall:  91.39%; FB1:  91.69  150
              ADV: precision:  89.32%; recall:  71.32%; FB1:  79.31  103
             CONJ: precision: 100.00%; recall:  92.86%; FB1:  96.30  39
              DET: precision:  99.18%; recall:  93.08%; FB1:  96.03  122
             NOUN: precision:  78.21%; recall:  91.44%; FB1:  84.31  560
              NUM: precision:  80.65%; recall:  73.53%; FB1:  76.92  31
             PRON: precision:  99.46%; recall:  94.33%; FB1:  96.83  184
              PRT: precision:  89.09%; recall:  85.96%; FB1:  87.50  55
             VERB: precision:  85.18%; recall:  87.29%; FB1:  86.22  371
                X: precision:  86.13%; recall:  81.42%; FB1:  83.71  173
```

Figure 1: Predictions for LR

```
D:\PycharmProjects\NLP Assignment 2 new>perl conlleval.pl -r -d \t < ./predictions/twitter_dev.crf.pred
processed 2114 tokens with 2114 phrases; found: 2114 phrases; correct: 1825.
accuracy:  86.33%; precision:  86.33%; recall:  86.33%; FB1:  86.33
                .: precision:  98.04%; recall:  98.43%; FB1:  98.23  255
              ADJ: precision:  66.67%; recall:  54.55%; FB1:  60.00  81
              ADP: precision:  87.58%; recall:  88.74%; FB1:  88.16  153
              ADV: precision:  87.62%; recall:  71.32%; FB1:  78.63  105
             CONJ: precision:  92.86%; recall:  92.86%; FB1:  92.86  42
              DET: precision:  95.16%; recall:  90.77%; FB1:  92.91  124
             NOUN: precision:  81.69%; recall:  86.64%; FB1:  84.09  508
              NUM: precision:  69.44%; recall:  73.53%; FB1:  71.43  36
             PRON: precision:  95.26%; recall:  93.30%; FB1:  94.27  190
              PRT: precision:  88.14%; recall:  91.23%; FB1:  89.66  59
             VERB: precision:  82.29%; recall:  87.29%; FB1:  84.72  384
                X: precision:  84.18%; recall:  81.42%; FB1:  82.78  177
```

Figure 2: Predictions for CRF

Features used (Case 2) -

- **Features used -** Basic Features + Set 1 + Set 2 + Set 3 + Set 5
- The total number of features were 24000. Lot less than earlier case.
- CRF gives a higher overall accuracy than LR. It also does better when we look at individual parts of speech (f1 scores). Refer to figures 3 and 4.
- **So here are the best set of features according to me keeping in mind the computation and efficiency - Basic Features + Set 1 + Set 2 + Set 3 + Set 5.**

```
D:\PycharmProjects\NLP Assignment 2 new>perl conlleval.pl -r -d \t < ./predictions/twitter_dev.lr.pred
processed 2114 tokens with 2114 phrases; found: 2114 phrases; correct: 1827.
accuracy:  86.42%; precision:  86.42%; recall:  86.42%; FB1:  86.42
                .: precision:  95.83%; recall:  99.61%; FB1:  97.68  264
              ADJ: precision:  73.77%; recall:  45.45%; FB1:  56.25  61
              ADP: precision:  91.78%; recall:  88.74%; FB1:  90.24  146
              ADV: precision:  90.29%; recall:  72.09%; FB1:  80.17  103
             CONJ: precision: 100.00%; recall:  90.48%; FB1:  95.00  38
              DET: precision:  99.18%; recall:  93.08%; FB1:  96.03  122
             NOUN: precision:  76.01%; recall:  89.98%; FB1:  82.41  567
              NUM: precision:  80.65%; recall:  73.53%; FB1:  76.92  31
             PRON: precision:  99.44%; recall:  92.27%; FB1:  95.72  180
              PRT: precision:  89.09%; recall:  85.96%; FB1:  87.50  55
             VERB: precision:  83.02%; recall:  86.46%; FB1:  84.71  377
                X: precision:  85.88%; recall:  79.78%; FB1:  82.72  170
```

Figure 3: Predictions for LR

```
D:\PycharmProjects\NLP Assignment 2 new>perl conlleval.pl -r -d \t < ./predictions/twitter_dev.crf.pred
processed 2114 tokens with 2114 phrases; found: 2114 phrases; correct: 1833.
accuracy:  86.71%; precision:  86.71%; recall:  86.71%; FB1:  86.71
                .: precision:  96.90%; recall:  98.43%; FB1:  97.66  258
              ADJ: precision:  65.17%; recall:  58.59%; FB1:  61.70  89
              ADP: precision:  88.16%; recall:  88.74%; FB1:  88.45  152
              ADV: precision:  86.67%; recall:  70.54%; FB1:  77.78  105
             CONJ: precision:  92.86%; recall:  92.86%; FB1:  92.86  42
              DET: precision:  98.32%; recall:  90.00%; FB1:  93.98  119
             NOUN: precision:  81.42%; recall:  88.73%; FB1:  84.92  522
              NUM: precision:  78.79%; recall:  76.47%; FB1:  77.61  33
             PRON: precision:  97.80%; recall:  91.75%; FB1:  94.68  182
              PRT: precision:  89.29%; recall:  87.72%; FB1:  88.50  56
             VERB: precision:  83.38%; recall:  87.29%; FB1:  85.29  379
                X: precision:  84.18%; recall:  81.42%; FB1:  82.78  177
```

Figure 4: Predictions for CRF

Sentences which highlight my features over the basic ones -

- Sentences containing urls, hashtags and at signs - @Rushikesh email id is rushikesh.nalla@stonybrook.com #Student #Stony Brook
- Sentences containing capital first letter (pronoun), last letter s (plural), punctuation, ends with "ly" (adverb), ends with "ing" (verb), ends with "st" (adjective) - Rushikesh has many books and committed to studying. He happily plays latest sport games.

Sentences where CRF is much better than MEMM -

- Sentences having tags that can be predicted based on context (previous tag) is where CRF performs much better than MEMM.
- One of the sentences in the dev set was - The dollar held near its highest in a month.
- MEMM model labeled the word "near" as NOUN whereas it is adposition.
- CRF model labeled the word "near" correctly.
- This sentence clearly shows that CRF was able to use the context "The dollar held" whereas LR was not able to.

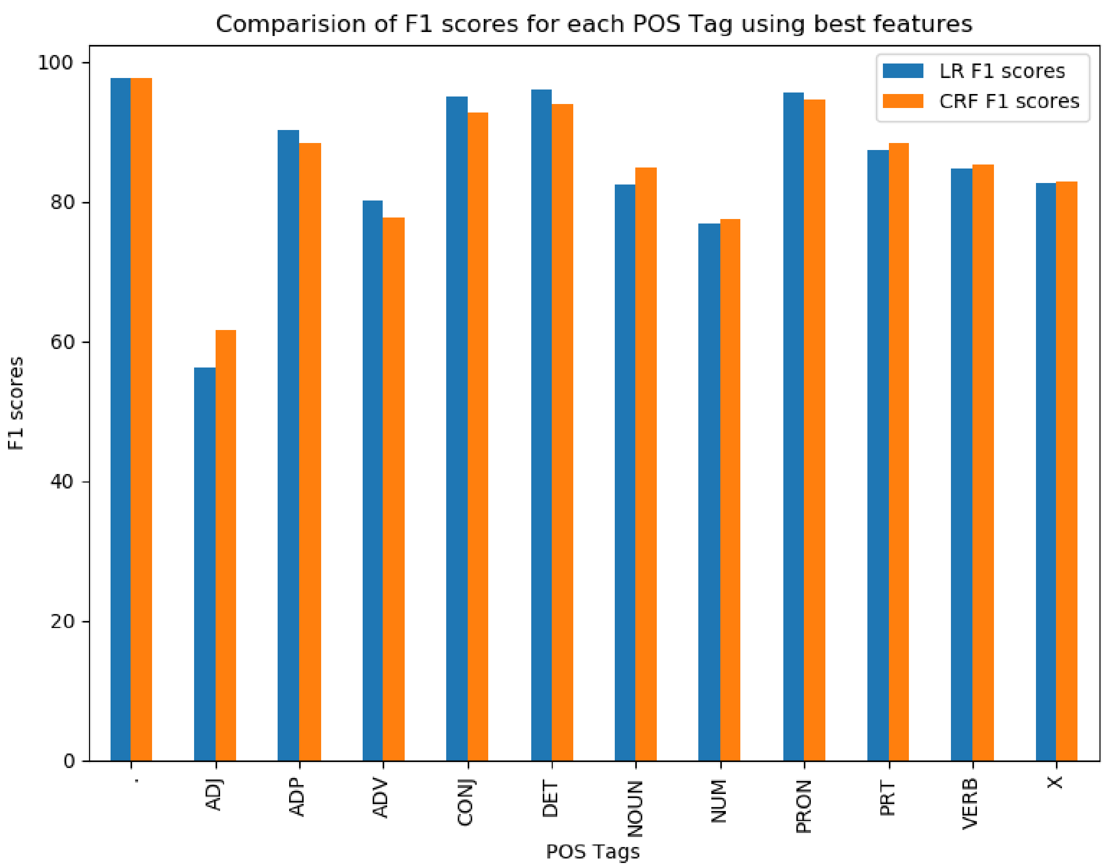

Figure 5: Comparison of F1 scores