E1246 - Natural Language Understanding
Assignment3 : Sequence Tagging

Tapan Bhardwaj(14704)
tapanb@iisc.ac.in

Abstract

In Assignment-3, I have developed an NER system for diseases and treatment using a CRF model that performs sequence tagging on every word of sentences in the dataset with D, T and O. Here I included several features like **POS tagging**, **semantic labelling**, etc. for every training instance which are further explained in the report. I have also performed ablation study and noted down my observations and reported them later in this report. The judgment for each model in the ablation study was on the basis of test accuracy.

1 Implementation

1.1 Model used:

MALLET is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text.

It includes sophisticated tools for document classification: efficient routines for converting text to "features", a wide variety of algorithms (including Nave Bayes, Maximum Entropy, and Decision Trees), and code for evaluating classifier performance using several commonly used metrics.

In addition to classification, MALLET includes tools for **sequence tagging** for applications such as named-entity extraction from text. Algorithms include Hidden Markov Models, Maximum Entropy Markov Models, and Conditional Random Fields. Hence, it is very good suited for solving our problem statement.

1.2 Model features

I have implemented the sequence models with the following attributes(features) for the given datasets.

- One of the feature is the semantic labeling from wordnet. Here, for each token for the given dataset, we append its semantic label from wordnet as a feature.
- POS tagging is another features appended to the data instances. Here, for every instance, we append its respective POS derived from nltk library's POS tag function.
- Capital is another feature added in the model. In this case, for every instance, I appended the word 'CAPITAL' as a feature if the training instance starts with a capital letter.
- One of the feature is included for every training instance if the instance is only made up of digits. In that case, the word 'DIGITS' is appended as a feature.
- Another important feature added to the model is 'stopwords'. For the stopwords identified from nltk library in the training instances, I appended 'stopwords' as a feature that training instance.

1.3 Sequence Tagging

For sequence tagging, I have used the Mallet's SimpleTagger tool as suggested in the problem statement. I have used 80% of the shuffled data(along with different subsets of above-mentioned features) for training and the rest of the 20% for evaluating test accuracy. I performed ablation study and switched on and off different subset of features and reported the results accordingly.

2 Result

For all the comparisons in the data, I have used 80% as training data and 20% as test data.

Features used	Train accuracy
No features	0.8846
Semantic	0.8924
POS	0.8892
Semantic + POS	0.8892
Capital + Digits	0.8928
Stopwords	0.8928
POS + Stopwords	0.8928
Semantic + Stopwords	0.8928
Semantic + POS + Cap + Dig	0.8889
All features	0.8928

Table 1: Accuracy with different feature subsets on training data

Features used	Test accuracy
No features	0.8795
Semantic	0.8869
POS	0.8853
Semantic + POS	0.8853
Capital + Digits	0.8872
Stopwords	0.8872
POS + Stopwords	0.8872
Semantic + Stopwords	0.8872
Semantic + POS + Cap + Dig	0.8861
All features	0.8872

Table 2: Accuracy with different feature subsets on test data

2.1 Experiment Analysis

Table 1 lists the accuracy of the model on the training data for different subsets of combinations of features. This is the ablation study on training data.
Table 2 lists the accuracy of the model on the test data for different subsets of combinations of features. This is the ablation study on test data.

An important observation that can be made here is that, 'stopwords' feature is a great influencer of the accuracy improvement for both the training and the test data. For the different cases studied here, without 'stopwords', almost every time, accuracy is less than that of with 'stopwords'.

From the above graphs it can be inferred that 'stopwords' is a very important feature because, whenever we include 'stopwords', we get very good(the highest mostly) accuracy results. This is also in line with our understanding that, while considering text features, if a words is not a stopword, we give it more importance than if it is a stopword.

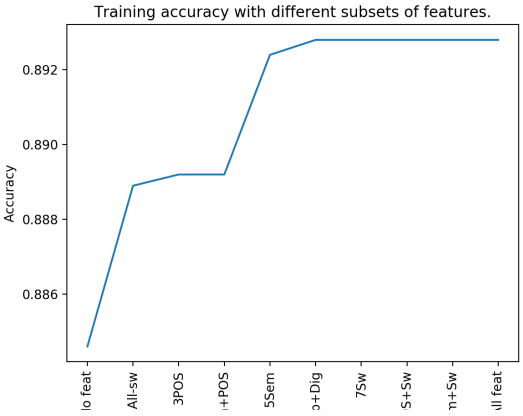

Figure 1: Figure shows the training accuracy with different subset of features.

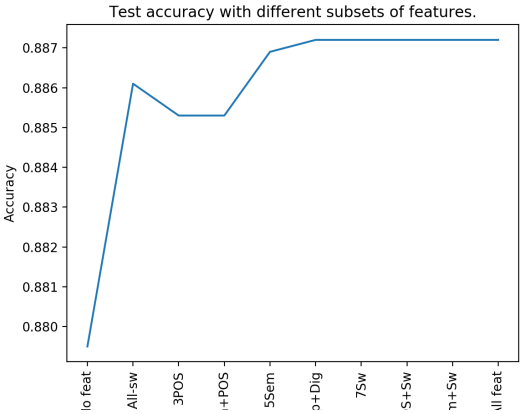

Figure 2: Figure shows the testing accuracy with different subset of features.

Following is the order of the features mentioned in the graphs above:

1. No features
2. All features included except 'stopwords'
3. only POS tagging added
4. Semantic labeling + POS taggin added
5. only Semantic labeling added
6. Capital and Digits feature added
7. only Stopwords added
8. POS tagging + Stopwords added
9. Semantic Labeling + Stopwords added
10. All the features added.

2.2 Accuracy

Best **training** accuracy considering all the features: **89.28%**

Best **test** accuracy considering all the features: **88.72%**

3 Usage

To run the generated text file with the required features, the following command has to be run:
java -cp "/home/username/mallet-2.0.7/class:/home/username/mallet-2.0.7/lib/mallet-deps.jar" cc.mallet.fst.SimpleTagger –train true –model-file model_file –training-proportion 0.8 –test lab –threads 2 ner_feat_new.txt
where, ner_feat_new.txt is the name of the text file with features appended that is generated when the code file is run with input file given in the problem. Also please add the required class path for the mallet function to be executed.

4 Github Link

https://github.com/TapanBhardwaj/NLP-projects