CENG 463 - Introduction to Natural Language Processing
Spring 2017-2018
Programming Assignment 1

Duygu Dogan
e1941962@ceng.metu.edu.tr

***Abstract*—This report aims to analyze implemented Part-of-Speech, Chunk and Name-Entity Relation taggers; on two different mathematical models (Logistic Regression and Conditional Random Fields) and come to a conclusion by a comparison.**

***Index Terms*—sequence tagging, part of speech tagging, chunking, shallow parsing, named entity recognition, maximum entropy models, logistic regression, conditional random fields**

I. Introduction

Here in this report, methodologies and terminologies of some basic concepts of Natural Language Processing will be explained firstly. Then, the process of selecting features and finding better ways to improve the performance of both classifiers and the algorithm will be told. Lastly, evaluation scores the classifiers on two different models will be observed, compared and concluded.

II. Used Technologies

- Python 3.6
- Natural Language Toolkit
- Libraries such as scikit-learn, nltk, matplotlib

III. Evaluation Metrics

Evaluation metrics are some values calculated on how tagging were on point. They are not the ultimate estimations the tagger's general performance independently so they need to be taken into consideration as a whole.

Before moving on, some abbreviations needs to be explained:
- True Positive (TP) means a tag was successfully on point.
- True Negative (TN) means a tag was avoided successfully.
- False Positive (FP) means tagger tagged a token it shouldn't have.
- False Negative (FN) means tagger didn't tag a token it should have.

A. Precision

Precision is the fraction of the examples which are actually positive among all the examples which we predicted positive.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Eg. According to Fig.1.;

$$P = \frac{100}{110} = 0.91$$

B. Recall

Recall is the fraction of the examples which are detected as positive among all the examples which are actually positive.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Eg. According to Fig.1.;

$$R = \frac{100}{105} = 0.95$$

C. F-Score

F-Score is the weighted average of Precision and Recall. F is usually more useful than accuracy in most of the cases since it includes precision and recall.

$$FScore = \frac{2 * (R * P)}{R + P} \tag{3}$$

Eg. According to Fig.1.;

$$F - 1 = \frac{2 * (0.95 * 0.91)}{0.95 + 0.91} = 0.93$$

D. Accuracy

Accuracy is the ratio of correct predictions to the total predictions. Higher accuracy is good but it's not much of a hint to interpret an estimation on the performance. There are cases where accuracy is high while precision and recall is not. This implies a problem with the algorithm.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (4)$$

Eg. According to Fig.1.;

$$A = \frac{100 + 50}{100 + 10 + 5 + 50}$$

E. Confusion Matrix

Confusion Matrix is a table to visualize evaluation metrics TP, TN, FP and FN differentiation. (Fig.1.)

Confusion matrices with heatmap plots for the pos tasks on both CRF and LR models can be seen in Fig.5.

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Fig. 1: Confusion Matrix

IV. Basic Concepts

A. Training, Evaluation and Test Splits in the Data

Splitting data is a traditional and intuitive method to estimate the validation of the model where total number of examples are split and used as train and evaluation sets. (The split may also provide some part for test data). This is done in order to make a better estimation on the classifier's performance on the real problem. The main goal is to predict how accurately the model will perform in practice. There are 90-10, 80-20, 70-30, 60-20-20, 50-25-25 as common partitionings but they don't depend on a rule at all. The thing is, having no randomness while splitting the data is highly probable to lead to overfitting (See Section IV-E for overfitting). (Fig.2.)

B. Cross Validation

Cross Validation is a better approach on model validation estimation when compared to data splitting because it prefers, well, cross validating the performance iteratively taking different parts of the set of examples as development set and the rest as training set. This will avoid the probability of landing on a rather inaccurate estimation stemming from the unevenly distributed data. (Fig.2.)

Related Task: Cross validation was implemented for logistic regression chunk task. Without cross validation, accuracy score was 1.0 since the training and development data were same. This process made the accuracy estimation more accurate.

This task's CRF part is missing since there was an incompatibility between the shape of the training data that suits to the *crf_suite* library and the shape that cross validation library needs.

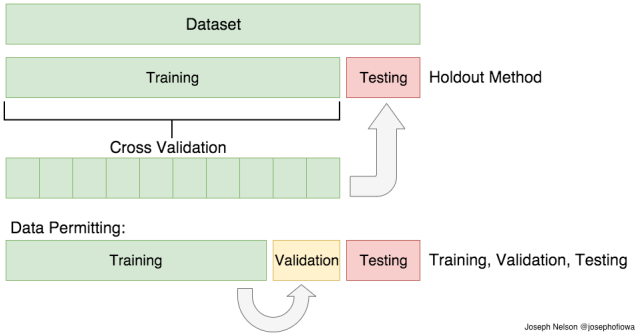


Fig. 2: Data Splitting and Cross Validation

C. Learning Curve

Learning Curve is a plotting concept where the aim is to show how growing amounts of data affects the performance of the model. When the amount of data at it's lower limit, 1, it's perfectly easy to fit a model in. But the problem is, this model will do a miserable job at predicting real world problems because it has almost no experience to infer from. On the other hand, as the data grows, the model's fit will become more and more flawed. But after some point, theoretically, the model will have nothing more to learn so having more data won't affect the performance anymore. Learning curves are used to observe and interpret such points, deciding how much data is enough and make further optimizations on the model. (Fig.3.)

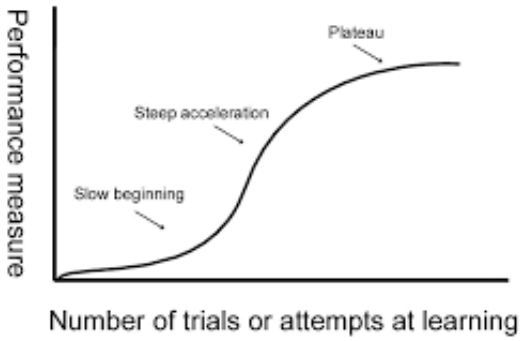


Fig. 3: A Theoretical Learning Curve

D. Parameter Optimization

Parameter Optimization is one of the most crucial parts of a Machine Learning problem since some mischosen parameters may affect performance drastically. There is something to emphasize though, the referred "parameters" here are not some values the model can predict. These are the parameters do decide how the algorithm work, so they are separated

from and external to the model itself. There are lots of different parameter optimization algorithms depending on how the problems vary.

Related Task: In order to improve quality, regularization parameters using randomized search and k-fold cross validation were going to be used. However, a time related technical problem occured and I had to kill the process.

E. Overfitting

Overfitting is the name for the problem which occurs when the classifier is over trained that it converges to the training data super closely. This is nothing more than training the model to only one instance of problem. It rather has to be training to predict different problems rather than training to perfectly solve what is obvious. So, overfitting will cause the model to be disguised as perfect but will probably fail on real data because it is only generalized for a specific problem set and it has nothing more in hands to make inferences when different problems are taken into observation.To avoid this, setting generalized rules rather than trying to hit a perfect score on the training data should be followed as a path.

V. Feature Selection Process

Features are the key elements to train a model to make it able to comprehend probabilistic conditional occurences and transitions. It is a rule of thumb to have 4 groups of features to have this web-like structure to perform well; character, word, tag and position based features. I had the basic ones first, then added or removed the ones which are for good and which didn't affect the performance outcome at all. It's all done by trial and error.

A. Rationale Behind Features

Following features were added to the feature list in the following order:

- *Character Based:* is_all_caps, is_all_lower, has_hyphen, is_numeric, capitals_inside, is_capitalized
  These features were chosen because the characteristics of characters in a word may affect the aimed meaning of the word pretty much. All caps hints an organisation, starting with a capital after the first word of a sentece hints a unique name etc.

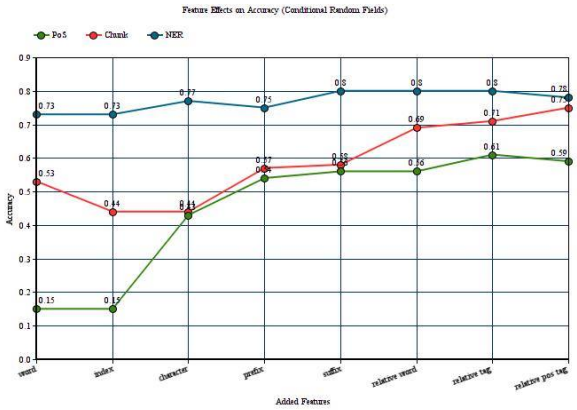
- *Word Based:* word, prefix-1/2/3, suffix-1/2/3
  These features were chosen considering the meaning of the words itself and the prefixes and suffixes affect the duty of the token.

- *Position Based:* index, prev_word, next_word
  These features were chosen to spot the possible relations between the words in distance of 1. And they are also good at giving hints about how the relative location affects the construct of the token. So the features include the previous and the next words with their tags and the index of the word itself.

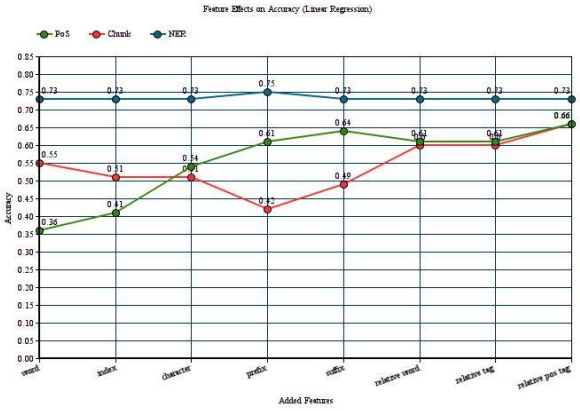
- *Tag Based:* pos_tag, prev_pos_tag, next_pos_tag
  These features are the strongest supply of inferences since they bear first-hand instructions for further predictions since it's "the" aim to apply what it's learnt from the training data. In order to achieve it, *nltk pos tag* function was used for chunk and named entity recognition tasks. Yet, its effect was not as it was expected.

B. How Adding Features Effects Accuracy

Almost every other addition to the feature set was a success and they affected evaluation scores positively. Since there is not one feature that affects all the tasks in a negative way, but there are, of course, some features which affects a couple of tasks negatively; these were handled in the code with conditional statement.



(a) Conditional Random Fields



(b) Logistic Regression

Fig. 4: Effects of Feature Selection

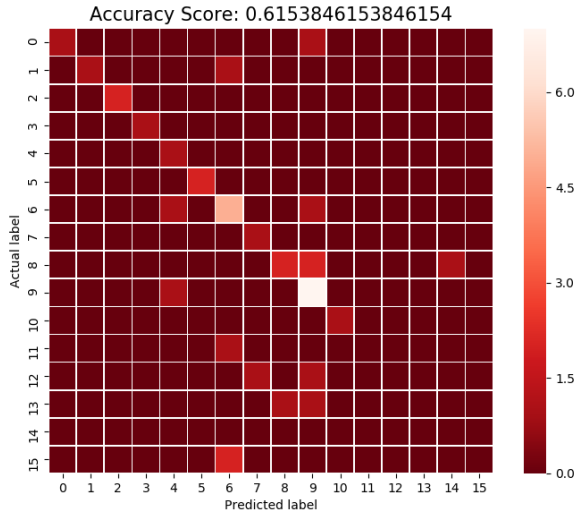
VI. Conclusion

A. Comparison between LR and CRF

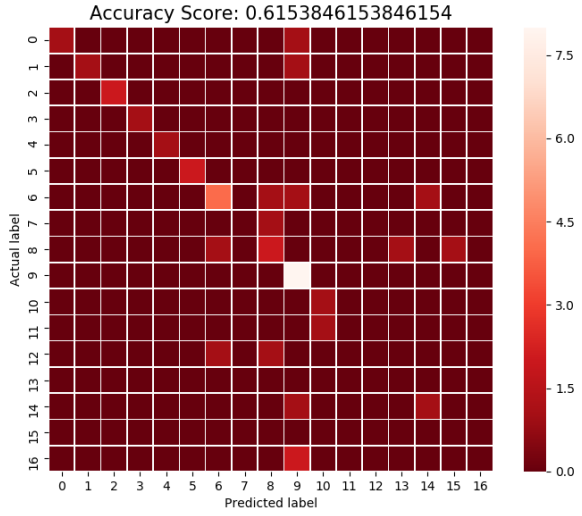
Even if two models look alike by comparing the accuracy scores, we can easily say that CRF performed sligthly better than LR when we take the other evaluation scores into consideration. That was already expected in the very beginning because CRF is only a generalized model of LR.

Additionally, training the CRF model takes more time than the LR model. It depends on the problem to choose CRF over LR because CRF is far more complex and generally, costs of a better model doesn't worth the little performance difference.

B. Confusion Matrices and Heatmap Plots



(a) Conditional Random Fields Pos Task



(b) Logistic Regression Pos Task

Fig. 5: Confusion Matrices with Heatmap Plots

C. Notes

For the NER tasks, O tags were omitted while calculating precision, recall and f-scores. This resulted in getting zero scores on these metrics with mini data because of the high percentage of O-tags which were omitted.