CE807 Text Analytics
Assignment 2: Information Extraction
Task 4

Nicolás Reyes Huerta - ID: 1700927

April 2018

1 Introduction

The following report will discuss how a NER system has been trained and evaluated using the "wp2" corpus, an automatically annotated corpus created by [4], and WikiGold, a gold standard annotated data set, (both based on Wikipedia articles). In particular, it will explain how features were selected, the classifier chosen and the results obtained. All this work was coded using python.

2 Feature Extraction

As any other machine learning process, feature design is the most critical aspect in a classification task, since it determines the labeling accuracy. The feature extraction carried out is inferred from the literature reviewed from task 1. The general idea behind feature extraction, as [1], [3], [5] and [9] describe, is to capture how an entity relates in a sentence. In this sense, the method used by them follows both lexical and syntactic patterns, such as sequence of words between an entity, if the first letter is in capital, if there is a digit in it, the POS-tag label among others.

Given the computational limitation of the machine in which the NER system was trained, it was decided to select a small set of features to create a simple baseline, as it is called in the literature, based on a window side of +1/-1 word for the ith word in a sentence. In other words, the information of the previous and the following word, for the ith word in a sentence is also considered relevant to train the NER system. This approach allows the model to learn and be able to recognize the entities that are at most three words long. By introducing the features describe below, it is aimed to:

- The word itself : Allows the classifier to find relationships between a sequence of words by knowing the word.
- POS-tag : Allows the classifier to find possible relationships between a sequence of words by finding patterns between different parts of speech tags.
- Digit : in a sequence of words there might be entities that have number as part o, such as Matchbox 20.
- Uppercase : If the word is in uppercase generally, it makes reference to an entity in English language.
- Title : In English language, the first letter of a named entity is always an uppercase word, such as Bill Gates, Apple, Chile, The Beatles.

3 Model Selection

To train the NER system, several classifiers were considered based on the literature review, such as logistic regression ([5] and [4]), Conditional Random Fields or CRF ([3] and [5]) and BBN's IdentiFinder ([2]). In particular, CRF was the chosen classifier, since its probabilistic framework make it ideal for labeling and segmenting sequential data. Additionally, CRFs have great flexibility in terms of how features can be included and finally, the documentation to train and evaluate the model was richer than for the other ones.

4 Training the Model

The objective of any training process is to allow an algorithm to generalize the information which is present in the training set, in order to make predictions when it faces unseen data. One of the problems that can arise during this phase is overfiting. Overfiting arise when the model, is not capable of generalizing and instead, learns specific elements which are present only in the training set and not necessarily reflects the real picture of the real world. In this sense, after being trained and faced with unseen data, the model performance is poor.

Regularization is a technique which prevents the model to overfit. By restricting its freedom, regularization introduces a penalty, on the different parameters of the model, for complexity during the training phase. Therefore, the model will be less likely to learn the particularities of the training set and will improve its ability to generalize.

There are several regularization techniques that can be used to penalize a complex model. Elastic-net is one of them. This technique was preferable among others for incorporating two kinds of penalties equal to the sum of the absolute value of the coefficients and the sum of the squared value of the coefficients. In this sense, the former will limit the number of features that can be used while the latter will force coefficient to be small, helping generalization.

It is worth pointing out that the architecture for training and evaluating the CRF classifier was based on a sklearn-crfsuite documentation example [7]. Therefore, some of the configurations that are set there were also adopted for simplicity. For comparison then, two model are presented. The first model (called "non- optimize") was trained employing all the training set using an arbitrary regularization parameters (RP) [1]. The second model (called "partially optimize") randomly generates the RP, using an exponential function and then, a 10 fold cross-validation was carried out 10 iteration each. This generates ten random models where the best, in terms of accuracy, was used to test the model obtained. Nevertheless, due to the lack of computational power, the training phase on the second one was carried out employing the first 10,000 examples of the training set. Additionally, the random generation of the regularization parameters follows an exponential distribution, as it appears in [7]. It is worth to mention that the[1] and [5] used a Gaussian distribution, but for simplicity the exponential distribution was adopted.

Finally, both models were trained using the L-BFGS method using gradient descent, since it is the default configuration for crfsuite in python.

5 Results and Discussions

The accuracy for both models described in the previous section are summarized in table 1.

Trained Model	F-Score (%)
Non Optimize	66.8
Partially Optimize	58.5

Table 1: Mean accuracy for different trained models

The results obtained are ambiguous. On the one hand, the non optimized model did not consider cross-validation so, the results obtained might be overestimated. Nevertheless, it gives us a good estimate of how capable the model could be. On the other hand, the partially optimized model did not consider all the data during the training set so, it is highly possible that the trained model is underfitted.

Even though the results obtained are ambiguous, it is believed that the partially optimize model does not reflect the true potential of the classifier. In this sense, it is believed that the true potential should be around the non optimize model and it is what will be taken as a final result for further discussions.

6 Improvements

The final results obtained in the previous section shows the potential that the model proposed has. Although the final result is far from the state of the art achieved by [8], it is not far from previous states of the art, such as [4] or [8]. Moreover, it is remarkable that the model achieved so similar results with a simple approach. Nevertheless, being so far form the state of the art, some improvement can be introduced in order to reduce this gap.

[1] It was decided to give the same penalty weight to both parameters for simplification, since it was not possible to optimize them. In this sense, the values presented in [7] were the same during the training process. Nevertheless, by reducing both values it was seen that the accuracy improved, since it is less costly to overfit.

As it was mention in section 2, not all the features that have been found in the literature review were possible to introduce. In this sense, the rest of the features found and proposed in the literature reviewed can be introduced particularly, by increasing the window size, which could allow to extend the ability to extract bigger relations between words and entity relations. Also, lemmatization and stemming process can be introduced to reduce the number of relations between word with the same root among other features. Although more and more features can be introduced into the baseline, it is believed that these approaches will just boost the performance slightly. Therefore, a most robust technique might be needed in order to keep boosting the performance of the model. As [10] proposes, for example, a word embedding technique, such as Glove, Word2Vec or Canonical Correlation Analysis can be introduce to train the NER System. In this sense, word embedding helps to group words with similar meanings and reducing the number of relations that the model could find. Moreover, it makes the model less susceptible to data scarcity.

References

[1] Kazama Jun'ichi, Torisawa Kentaro. *Exploiting Wikipeadia as External Knowledge for Named Entity Recognition.*

[2] Richman Alexander, Schone Patrick. *Mining Wiki Resources for Multilingual Named Entinty Recognition.*

[3] Kim Sungchul, Toutanova Kristina and Yu Hwanjo. *Multilingual Named Entety Recognition using Parallel Data and Metadata from Wikipedia.*

[4] Nothman Joel, Ringland Nicky, Radfoard Will, Murphy Tara and Curran James R. *Learning Multilingual Named Entity Recognition from Wikipedia.*

[5] Mintz Mike, Bills Steven, Snow Rion and Jurafsky Dan. *Distant Supervision for Relation Extraction without Labeled Data.*

[6] Miran Seok, Hye-Jeong Song, Chan-Young Park, Jong-Dae Kim and Yu-seop Kim *Named Entity Recognition using Word Embedding as a Feature.*

[7] *https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.htmlfeatures*

[8] Agerri Rodrigo and Rigau German. *Robust Multilingual Named Entity Recognition with Shallow Semi-Supervised Features.*

[9] Abbas Ghaddar and Philippe Langlais. *WiNER: A Wikipedia Annotated Corpus for Named Entity Recognition.*

[10] Seok Mirian, Song Hye-Jeong, Park Chan-Young, Kim Jong-Dae and Kim Yu-seop. *Named Entity Recognition using Word Embeding as a Feature.*

Appendices

A Precision, Recall and F.Score by Entity for the Non Optimize Model

Entity	Precision(%)	Recall(%)	F-Score (%)
I-PER	77.1	82.4	79.7
I-ORG	69.0	53.5	60.3
I-LOC	65.1	80.4	71.9
I-MISC	51.5	59.8	55.4
Average	66.4	68.3	66.8

B Precision, Recall and F.Score by Entity for the Non Optimize Model

Entity	Precision(%)	Recall(%)	F-score (%)
I-PER	67.0	81.2	73.4
I-ORG	62.9	42.7	50.9
I-LOC	55.9	65.1	60.1
I-MISC	45.0	56.0	49.9
Average	58.5	60.4	58.5