Natural Language Processing, Homework 1

Domenico Alfano & Marco Treglia

1. Introduction

The report describes the results obtained during the development of the first Homework. The main goal of this project has been to evaluate the capabilities and functionalities of Supervised Morphological Segmentation in a Low-Resource Learning Setting using Conditional Random Fields.

2. System overview

The Model has been implemented in Python 3.5. We create a class called Morph where it takes as input the folder of the train data, the folder of the test data and the δ parameter. After loading the data, this model follows two steps: the first step concerns the features extraction from the words, the other one concerns the labels provided by the morphemes.

2.1. Features

In the first step, the words has been extracted with the function *extract_word*, then we create a list of sequence of all the words divided by character with the function *all_words_sequence*. Successively, we create the features from the sequences of characters of the words with the two static method *right_fetures_set* and *left_fetures_set* called by the function *character_fetures* which make a dictionary form the character passed by the function *word_feature_extractor*. This step provides a list of a list of a dictionary called *training_feature* as requested by *crf_suite*.

2.2. Label

In the second step, the morphemes has been extracted with the function *label* which creates the list of morphemes for each word as input for the function *add_tag_to_morph*, that return the label for the correspective word.

3. Result

The results obtained by our trained model, in terms of precision, recall and F1 test, by varying δ in range $[1, 12]$, are shown in the Figure 1.

As we can see, the best F1 score is given with $\delta = 4$. Table 1 shows in detail the scores for $\delta = 4$.
Successively we try different split of the training data, merging the train and dev file and predicting the test file. The results obtained by our trained model, in terms of precision, recall and F1 test, by varying δ in range $[1, 12]$, are shown in the Figure 2.

As we can see, the best F1 score is given with $\delta = 4$ Table 2 shows in detail the scores for $\delta = 4$.

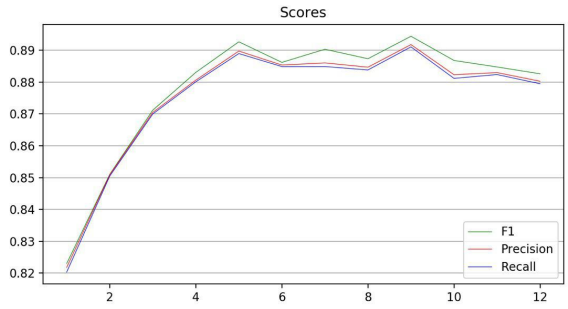

Figure 1. Scores given by training train file, predicting test file

B	M	E	S	AVG
0.929174	0.865973	0.915683	0.934307	0.894402
0.833585	0.945224	0.821483	0.927536	0.891782
0.878788	0.903865	0.866029	0.930909	0.891063
661	1497	661	138	2957

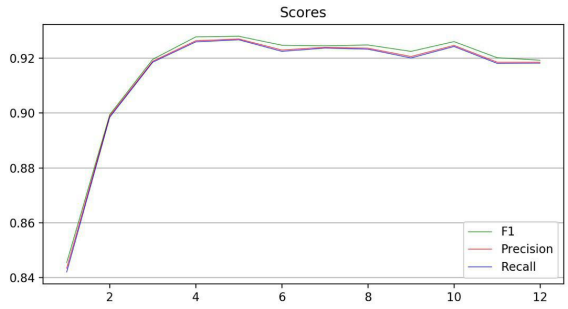

Figure 2. Scores given by training train and dev file, predicting test file

B	M	E	S	AVG
0.948387	0.911616	0.937299	0.961832	0.92792
0.889561	0.964596	0.881997	0.913043	0.926953
0.918033	0.937358	0.908807	0.936803	0.92663
661	1497	661	138	2957

4. Task 2

For this Task we choose the Italian language, creating three text file. The Train file consists in 100 Italian words with the corrispective morphemes, the Test file and the Dev file consists in 25 words also with the corrispective morphemes.
After the creation of these files, we trained our model, as for Task 1, in terms of precision, recall and F1 test, by varying δ in range $[1, 12]$ as we can see in Figure 3.

As we can see, the best F1 score is given with $\delta = 8$.
Table 3 shows in detail the scores for $\delta = 8$.

Successively we try different split of the training data, merging the train and dev file and predicting the test file. The results obtained by our trained model, in terms of precision, recall and F1 test, by varying δ in range $[1, 12]$, are shown in the Figure 4.

As we can see, the best F1 score is given with $\delta = 8$.
Table 3 shows in detail the scores for $\delta = 8$.

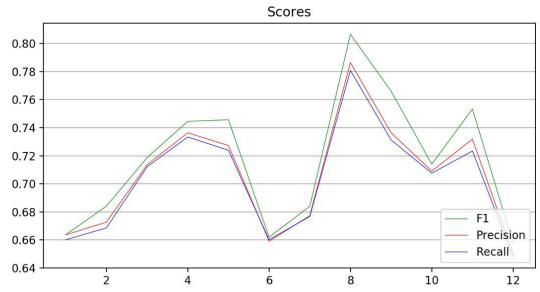

Figure 3. Scores given by training train file, predicting test file

B	M	E	S	AVG
0.842105	0.711864	0.868421	0.923077	0.806535
0.615385	0.954545	0.634615	0.857143	0.786364
0.711111	0.815534	0.733333	0.888889	0.780759
52	88	52	28	220

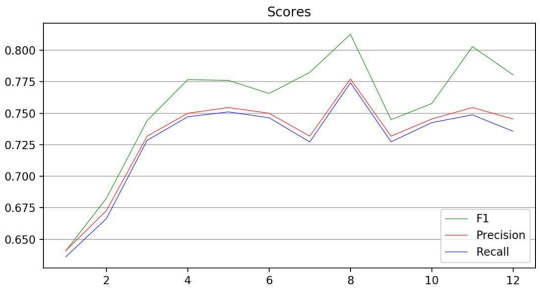

Figure 4. Scores given by training train and dev file, predicting test file

B	M	E	S	AVG
0.888889	0.688525	0.888889	0.92	0.812703
0.615385	0.954545	0.615385	0.821429	0.777273
0.727273	0.8	0.727273	0.867925	0.774265
52	88	52	28	220