1 Used approach

For the purposes of training the NER system, the Learning Based Approach is utilized, using a supervised method. In particular, a Conditional Random Fields (CRF) based model has been used, which in this case is the CRFpp (CRF ++) machine learning framework since for the purposes of implementing the whole system I should use distant learning / supervision methods.

2 NER system Training Analysis

The system consists of two parts, the training of the Named Entity Recognition (NER) system and its testing.

To initiate the training, initially wp2 (aij-wikiner-en-wp2) automatically annotated corpus has been downloaded, which is also almost in IOB format (“traincorpus”). In order to change and bring the training data and its features in accordance with the CRF ++ machine learning framework to the required format, it was pre-processed by using a Perl coded script (preprocessing_traincorpus.pl), using also the terminal of the operating Linux system (Ubuntu) for the purposes of its execution and its output is the preprocessed_traincorpus txt file.

As it was mentioned above, when it comes to IOB format, every token of a sentence in a data set is being labeled with a chunk label, one of the three following exceptional chunking tags, which more specifically are I (Inside), O (Outside), or B (Beginning). In the event that a token denotes the beginning of a chunk, it is labeled as "B". The following tokens in that particular chunk are labeled as "I". Every single other token in the chunk is labeled as "O". Moreover, the labels "I" and "B" are followed by the type of the chunk, as for instance I-NP and B-NP [5].

Moreover, before the beginning of the training phase, CRF++-0.58 toolkit has been downloaded and installed by following a particular procedure. The terminal of the operating Linux system (Ubuntu)has been utilised for the purposes of the aforementioned procedure and to execute the appropriate commands, as it can be seen in the following 4 figures.

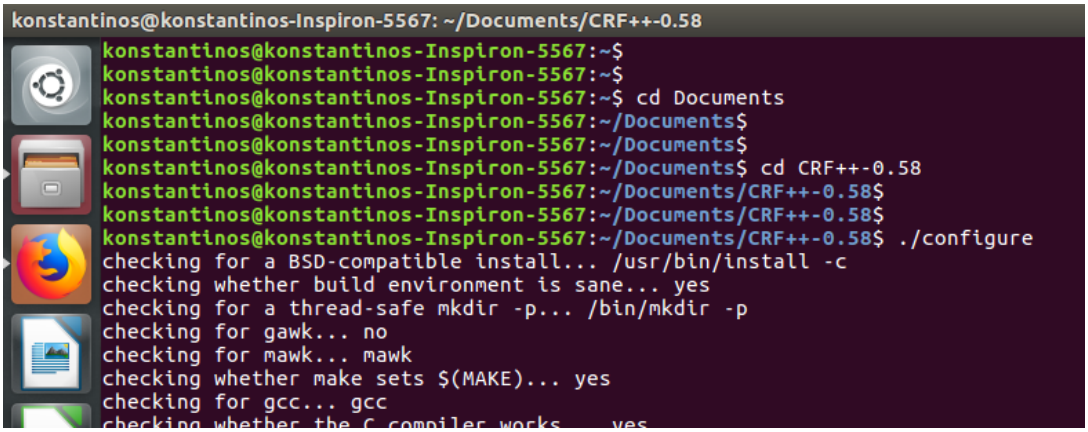

Figure 1: Installation of CRF++-0.58 toolkit

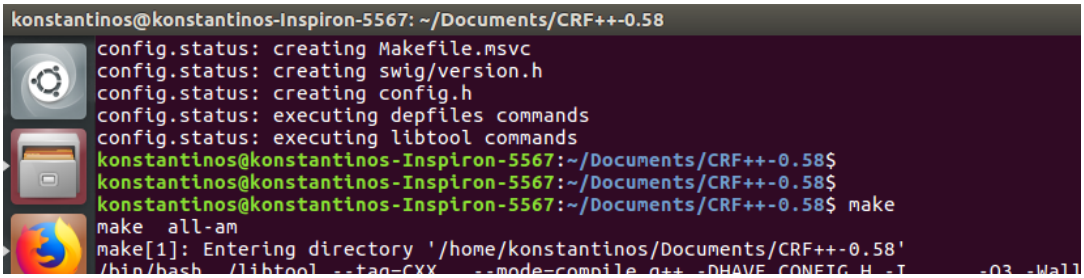

Figure 2: Installation of CRF++-0.58 toolkit

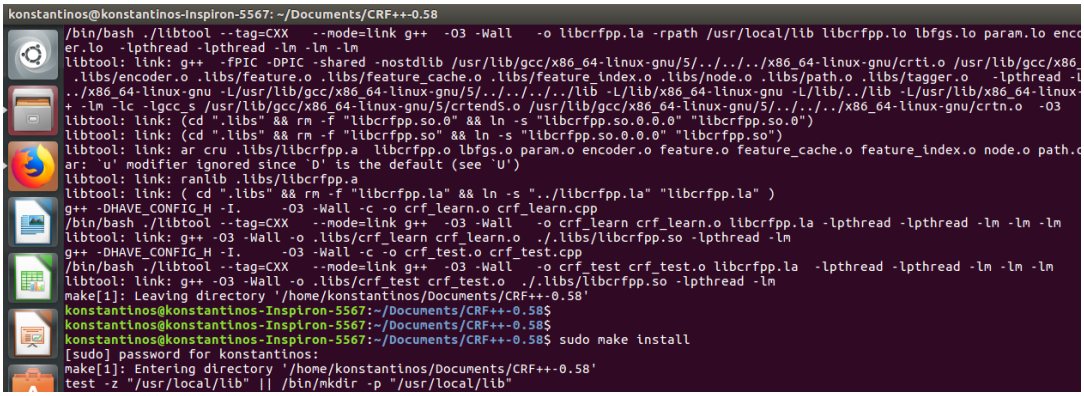

Figure 3: Installation of CRF++-0.58 toolkit

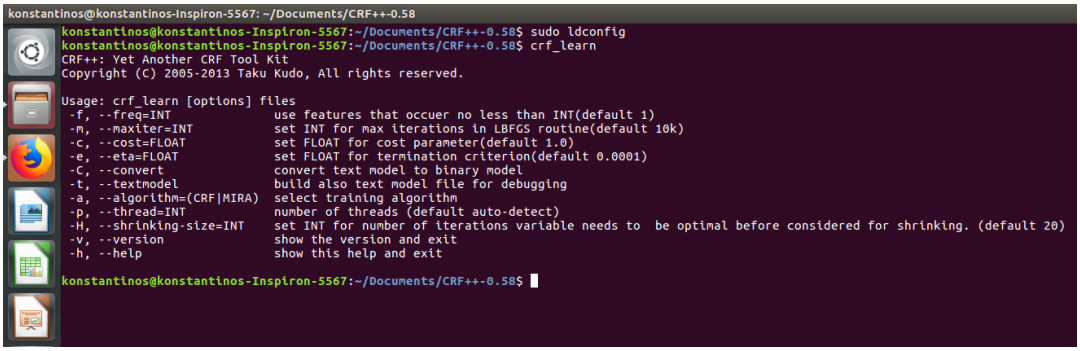

Figure 4: Completed CRF++-0.58 toolkit installation

Afterwards, the preprocessed_traincorpus data set was used together with the CRF ++ template file, which is the “template_pos_bigrams” file, to initiate the training of the NER system by typing in the following command as it can be seen in the figure 1 below, using the terminal of the operating Linux system (Ubuntu) to execute the command.

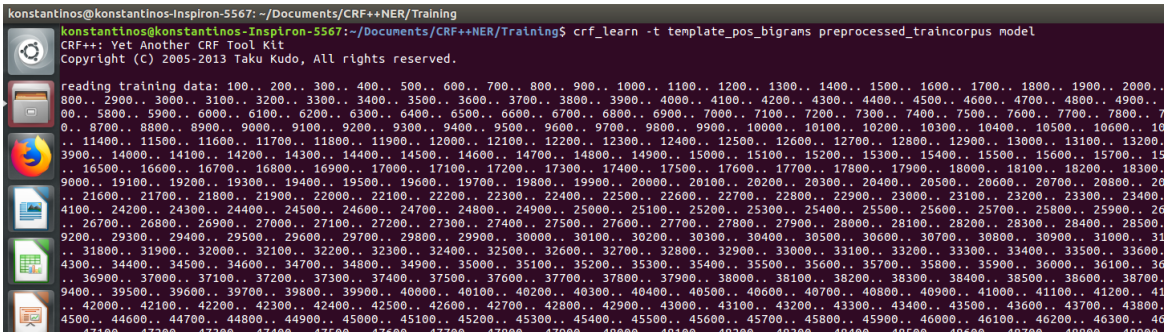

Figure 1: CRF++ training phase

After typing in this command, the CRF++ training model is being extracted.

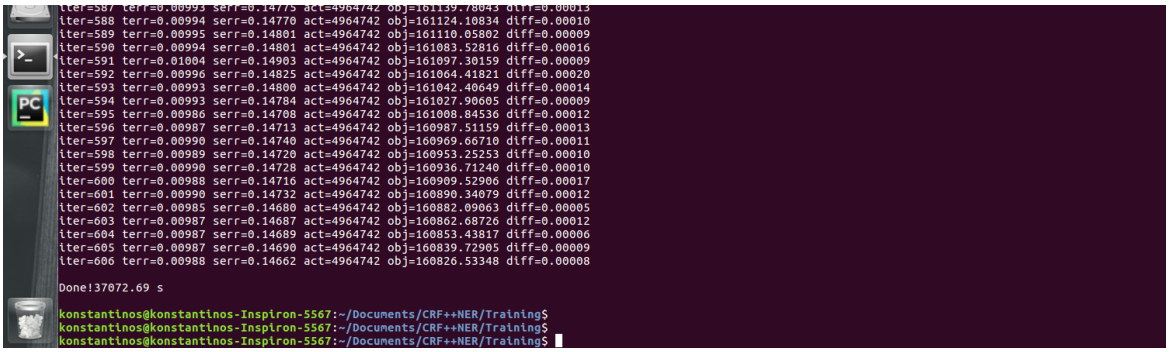

Figure 2: CRF++ training phase finished (606 iterations and approximately 10 hours to finish)

What is more, there is an option of reading the extracted model by typing in the terminal, the command ls model* to display in the terminal the extracted model files and then the command vim model.txt in order to open and read the extracted model after the training phase of the Named Entity Recognition system. Nevertheless, it is exactly the same as locating manually and displaying the file from the path that the extracted model has been stored, in the system’s folder. Also, the extracted model’s txt file has to be stored in the testing folder of the whole system as well, for the purposes of the testing phase, which will be analysed below.

With regards to the template file, it consists of several lines where each one of them signifies one particular template. Every template of the whole template, will have a determined token from the input information (data), which is accomplished by using the special macro %x[row,col], where "row" is for appointing the relative position from the present token which is being focused, as well as "col" indicates the outright column position. However, each line that is not empty, is a training data point in the training data for the CRF ++ train model. It is important to emphasize that only those features that are in the template file and has been stated exactly, will be utilized for

training our model. Also, bigram features are being defined by this template file. More specifically, a blend of the present yield token and past yield token (bigram) is consequently produced by using this template.

As it can be seen in the figure above, "iter" is the number of processed iterations, "terr" is the error rate concerning labels (number of error tags divided by the mumber of all tags), "serr" is sentence error rate (number of error sentences divided by the number of all sentences). Additionally, the "obj" stands for current object value. At the point when this esteem unites to a settled point, the iteration is being stopped by CRF++. Lastly, "diff" is the relative difference from the past object value.

3 Testing Analysis

Subsequently, in order to start the testing phase of the whole NER system, the gold standard data set wikigold.conll.txt has been collected and aggregated in order to evaluate the whole system. In order to change and bring the training data and its features in accordance with the CRF++ machine learning framework to the required format, it was pre-processed by utilizing a Python coded script (Preprocess_testing_corpus.py), using also the terminal of the operating Linux system (Ubuntu) for the purposes of its execution, as its output is the “preprocessed_wikigold.conll” txt file.

Then, the “preprocessed_wikigold.conll” data set was utilized along with an empty txt file (crf_wikigold_conll_testoutput.txt) and the extracted training model from the training phase, in order to test the whole system and create an output “crf_wikigold_conll_testoutput.txt”, which contains the predicted NER features of the “preprocessed_wikigold.conll” data set, as it can be seen in the figure 3 below. The terminal of the operating Linux system (Ubuntu) was utilized to execute the command.

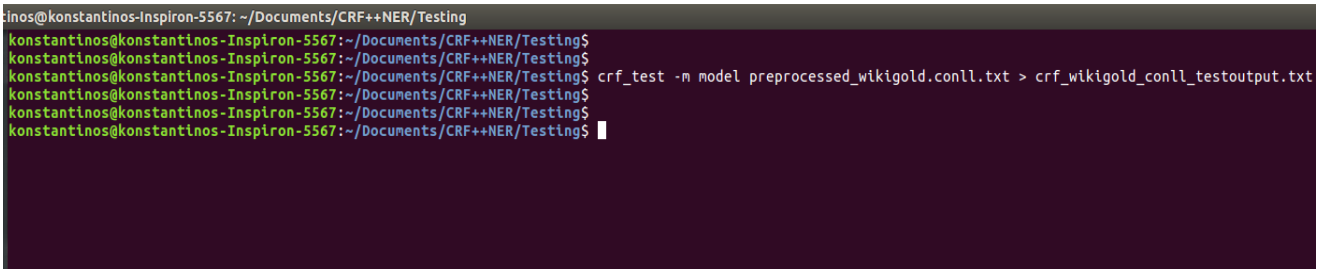

Figure 3: CRF++ testing using the wikigold.conll corpus

4 Evaluation

For the purposes of the evaluation of the extracted model and, our whole Named Entity Recognition (NER) system, a python coded script (Crf_TestEvaluation_Accuracy.py) has been utilized with the crf_wikigold_conll_testoutput.txt file as an input. In this script the main task is to check if the last two words of each row are the same (i.e. compares the actual label and the predicted label). As it can be seen in the figure 4 below, the accuracy score of the system performed is approximately 85.48%.

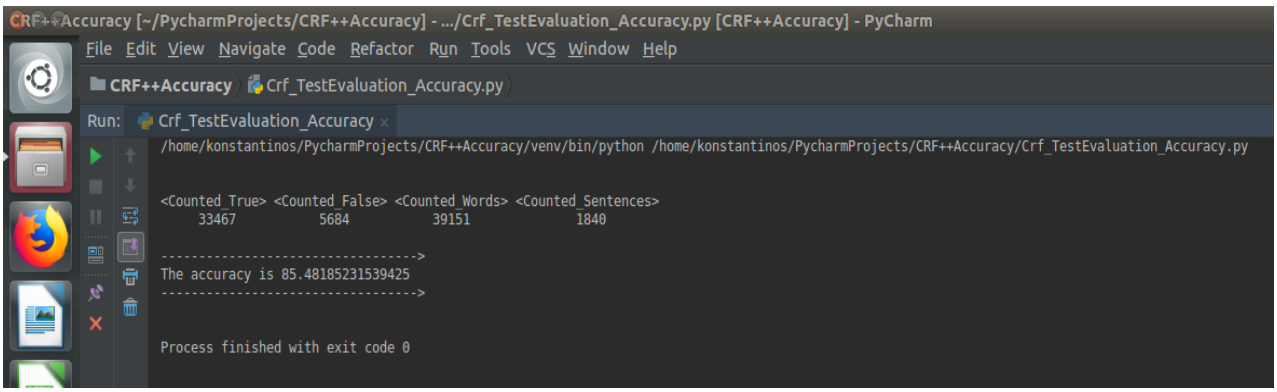

Figure 4: Evaluation of the NER implemented system

5 Comparison with the state of the art

By comparing with state-of-the-art models, we can observe that the performance of the implemented system is quite good and comparable. More specifically, researchers have used Distant Supervision (DS) from Wikipedia (in Nguyen and Moschitti, 2011), a supervised modeled approach for Sentence-level Relation Extraction (SLRE), where they were based on kernel methods (KM) and Support Vector Machines (SVMs) where the best in class models were in light of. In that particular research, the tests demonstrated that the particular approach is vigorous to Web records, as well as high accuracy has accomplished and more specifically, 74.29% for f1 score measure utilising 52 YAGO relations in [1]. As it can be seen, our accuracy is higher than that and thus our approach and model is robust and better than the aforementioned one.

However, our NER system's accuracy is lower than the robust accuracy of 95% achieved after that system's evaluation, by the researchers who utilized the knowledge base and the YAGO ontology (version 2008-w40-2) for 99 relations in [1]. It can be

observed that on average and compared to state-of-the-art models / approaches, the performance of the implemented model is considered quite good.

References

[1] Truc-Vien T. Nguyen and Alessandro Moschitti: Joint Distant and Direct Supervision for Relation Extraction. In Proceedings of the 5th International Joint Conference on Natural Language Processing. Chiang Mai, Thailand, November 8 – 13, 2011 pages 732–740.

[2] Miao Fan, Qiang Zhou and Thomas Fang Zheng: Distant Supervision for Entity Linking. In Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation. Shanghai, China, October 30 - November 1, 2015, pages 732–740.

[3] Mike Mintz, Steven Bills, Rion Snow, Dan Jurafsky: Distant Supervision for Entity Linking. In Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP. Suntec, Singapore, 2-7 August 2009, pages 1003–1011.

[4] Mihai Surdeanuy, Julie Tibshiraniy, Ramesh Nallapati, Christopher D. Manning. “Multi-instance Multi-label Learning for Relation Extraction”.

[5] NLTK, " Extracting Information from Text” [Online]. Available at: http://www.nltk.org/book/ch07.html/ Last accessed: 22 April 2018.