CS 445 Natural Language Processing Project 4: Named Entity Recognition

Project Goal and Dataset Explanation

In this project the goal was to develop an effective NER tool for Turkish using Machine Learning approaches. We used CRF implementation of Python sklearn which is called called crfsuite. We tried different features and calculated the precision, recall and F1-Measure.

We were provided with a Turkish dataset which was labeled for PERSON, LOCATION and ORGANIZATION entities. We split the dataset into 5 folds, by iterating over the lines one by one and adding the first one into first fold, the second one into second fold and so on.

In addition to the original data, we were also provided with the morphological analysis of the words in the original data. Each word was in a line and the first element of the line referred to the sentence that the word belonged to, second element was the word itself and the third element was the morphological analysis of the word in the form "müzik+Noun+A3sg+Pnon+Nom". This data was used for constructing the features.

Implementation Approach and Preprocessing

A list of list of dictionaries is given to the model as X_train, the lists inside the biggest list hold sentences and the dictionaries inside the sentence list holds feature dictionaries of each word belonging to that list. Here is an element of this structure printed:

```
[{'INF': 'Venedik+Noun+Prop+A3sg+Pnon+Nom', 'Stem': 'Venedik', 'POS':
'Noun', 'PROP': 1, 'NCS': 'Nom', 'SS': 1, 'OCS': 1, 'LOCLEXICON': 0},
{'INF': 'film+Noun+A3sg+Pnon+Nom', 'Stem': 'film', 'POS': 'Noun', 'PROP':
0, 'NCS': 'Nom', 'SS': 0, 'OCS': 1, 'LOCLEXICON': 0}…]
```

The keys in the dictionary refers to the features that are used, I talk about this in more detail in upcoming parts of the report.

A list of lists was given to the model as y_train. The lists inside the list represent individual sentences, however the words of the sentences are represented with their labels, which are 'B-ORGANIZATION' if the word belongs the organizations and is the first word of that organization name, 'I-ORGANIZATION' if it's not the first word but belongs to organization. Same rules are carried out for 'B-LOCATION', 'I-LOCATION' and 'B-PERSON', 'I-PERSON', if the word is not labeled its value is 'O'. Here is an element printed:

```
['O', 'O', 'O', 'B-ORGANIZATION', 'I-ORGANIZATION', 'I-ORGANIZATION', 'O']
```

After get obtaining X_train and y_train I divided them into 5 folds and fit the model to these 5 folds and calculated the average result as my accuracy.

Features Used and Their Effect

I implemented the following features:

1. **Root (Stem)**
2. **Part-of-Speech (POS)**
3. **Proper Noun (PROP)**
4. **Noun Case (NCS)**
5. **Orthographic Case (OCS)**
6. **Start of the Sentence (SS)**
7. **Location Lexicon (LOCLEX)**

Turkish is a productive language, meaning, many words can be created from a single word by adding derivations etc. This may cause a data sparsity issue when training the models as there could be many variations of one word in data. We use features related to a word's morphology and its root to prevent this.

➔ The accuracy when all the features are used is `0.855904694473485`, this is our benchmark for our comparisons of results.

1. **Root (Stem)**

To implement this feature I extracted the first word up to the '+' in the morphological analysis inside NE.ma.txt. Due to the structure of the analysis the first word always gave the root. Then in the features dictionary, I directly gave the value of root this string I extracted.

Comments on Performance: The accuracy was `0.49654680849196653` when Root feature was not used. The accuracy decreased significantly. We can easily say that using this prevents data sparsity thus improves our accuracy. This makes sense since looking at roots eliminates the data sparsity issue earlier by decreasing the amount of variations that we can get from one word to appear in the data.

2. Part-of-Speech (POS)

This was implemented by getting the second element of morphological analysis, in our case the second element always gave POS, which is Noun, Verb, Adj etc.

Comments on Performance: The accuracy was `0.8535298507679873` when Part-of-Speech Feature is not used. There is a decrease, but this decrease is not as significant as in the case with root feature.

3. Proper Noun (PROP)

If the proper noun existed in morphology analysis, I gave its value 1, if it didn't, I gave 0.

Comments on Performance: The accuracy was `0.8511046977065204` when Proper Noun Feature is not used. There was a decrease, so we can say that using this feature improved the model but this decrease isn't very significant.

4. Noun Case (NCS)

This was the last element of the morphological analysis. I equalized this feature to 0 for non-nominal tokens and the corresponding nominal for the nominals: Nominative(NOM), Accusative/Objective(ACC), Dative (DAT), Ablative(ABL), Locative(LOC), Genitive(GEN), Instrumental(INS), Equative(EQU).

Comments on Performance: The accuracy was `0.7827925192712681` when Noun Case Feature was not used. This can also be counted as a significant decrease in accuracy, we can say that NCS feature is important.

5. Orthographic Case (OCS)

This was a lexical feature. I checked if the first letter of the word was uppercase or not for this feature. If it was uppercase, I gave 1 to its value, if it wasn't, I gave 0.

Comments on Performance: The accuracy was `0.8154412200682473` when Orthographic Case feature was not used. There was a decrease and we can say that this feature improved the model.

6. Start of the Sentence (SS)

This was a lexical feature like OCS and to get this I basically checked if the word was the first element of a sentence list. If it was the value was 1, if it wasn't, it was 0.

Comments on Performance: The accuracy was `0.8512165807717025` when Start of the Sentence feature is not used. There was a decrease but this decrease is not as significant as most of the other cases.

7. LOCATION LEXICON (LOCLEX)

The last feature was an additional feature that was a gazetteer-based feature, I added. I used my 'location.csv' file from my Project 01 and checked if the word existed in that lexicon, this was an indicator feature. If the word existed the value for the feature was 1, if it didn't exist, it was 0.

Comments on Performance: The accuracy was `0.8524276222629513` when Lexicon feature was not used. There was a decrease, but by no means was this decrease significant. We can say that adding this feature didn't cause a significant change.

Conclusion and Additional Comments

→ The CRF model gives a good accuracy of %85.5 while doing Named Entity Recognition on Turkish data. Features could be implemented to the model easily, which is a good thing especially for Turkish data, since Turkish language is a very productive language so the use of morphological analysis of words in training reduces data sparsity.

→To talk about the 'B-' and 'I-' tags of labels we can say that 'B-' labels are more accurately identified than 'I-' labels. We can come to this conclusion by the below screenshot. This maybe because there are more entities with a single word (with label 'B-') in the training data. We can also see that 'I-LOCATION' performs worst, hence why I wanted to implement a feature specifically for location, which was the 'LOCLEX' feature, this didn't have much of an impact, although it improved accuracy by a little.

	precision	recall	f1-score	support
B-LOCATION	0.934	0.917	0.925	480
I-LOCATION	0.500	0.286	0.364	28
B-ORGANIZATION	0.930	0.720	0.812	446
I-ORGANIZATION	0.768	0.684	0.724	228
B-PERSON	0.908	0.888	0.898	890
I-PERSON	0.853	0.883	0.868	283
micro avg	0.894	0.834	0.863	2355
macro avg	0.816	0.730	0.765	2355
weighted avg	0.893	0.834	0.860	2355

→All in all, we can say that the most effective feature is Root Feature, then Noun Case and then Orthographic case. Least effective ones were, Part of Speech and Location Lexicon.
→ During the implementation process, while I was trying to fit the train data into the model I came across an error like this:

```
ValueError: The numbers of items and labels differ: |x| = 41, |y| = 40
```

This error meant that some of the elements of my feature training set and label training set weren't matching. So, I checked the ones that don't match inside a for loop and found out that there were several differences in my way of tokenizing sentences and NE.ma.txt's way of tokenizing them. I decided to drop these sentences that caused a problem, the accuracy was still moderately high after I did this so we can say that it didn't affect accuracy that much.

→ Also, as I was doing the project, I didn't implement all of the features at first try, I only used 3 features which were Root, Proper Noun, All Inflectional Features. When using only these 3 features my accuracy was: `0.6685380708581905`, then it improved to `0.855904694473485` so there was a significant change and we can easily say that implementing morphological features while training Turkish Dataset for NER, improves our accuracy significantly.

→ I wasn't able to implement All Inflectional Features (INF), as it caused some problems in the accuracy and I didn't have enough time to understand this, unfortunately. However, my approach to getting the INF feature is commented out and is visible in my '.ipynb' file.

REFERENCES

[1] Gökhan Tür, Dilek Z. Hakkani-Tür, and Kemal Oflazer. (2003). A statistical information extraction system for Turkish. In Natural Language Engineering, pages 181–210.
[2] Reyyan Yeniterzi. (2011). Exploiting morphology in Turkish named entity recognition system. In Proceedings of the ACL 2011 Student Session, Portland, OR, USA.
[3] Gökhan Akın, Gülşen Eryiğit. (2012) Initial explorations on using CRFs for Turkish Named Entity Recognition, COLING.
[4] Tutorial¶. (n.d.). Retrieved January 24, 2021, from https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html