**Assignment 4: Relation Extraction**

**Report**

## Description of our system

In the past three weeks we try many methods, machine-learning system and rule-based system, but unfortunately, we had to give up on them because they did not work as we expected. The system we delivered is machine-learning approach. The model is train with sklearn's SVC model.

Since we have seen that there is a certain similarity in entities between different relations, e.g.: 'live\_in' & 'work\_in' are both use person for OBJ1 and GPE for OBJ2 - we decided to train the model on all of the relations in the train corpus, so it could distinguish between such similarities. In predict time we return only the pairs of entities that was predicted we relation of 'live\_in'.

We start to use the following features. but later, we dropped some of them to get better performance in F1, recall, and prediction.

Features:

* Entity type
* POS
* DEP
* IOB

Our system trains on the corpus with SVC and with the features that mention above. After the model will write two files to later use – feature map file, and model file. Our model can later been load from these files and predict other corpus files without re-train.

## Error analysis

As mention above, we got the many dead-end roads in our process. We deal with a lot of problems and challenges in the way. We will try to describe as many as we can remember from these.

### Spacy entry text problem

We found that many entities from spacy have text that have a small and annoying differences between the annotation file result. As a result, many examples appeared in a slight change as both FN and FP. Examples:

* "Mrs. Higgins" return from spacy as "Higgins"
* "Umbria" return from spacy as "Umbria province"
* "Hakawati Theatre" return from spacy as "the Hakawati Theatre"

This kind of mistakes was decease our recall and our prediction rates, also in train time this caused us to miss about half of the relations in the train.annotation file. We try many methods to reduce the recurrence of these errors. We tried to use both of spacy.ents and spacy.noun\_chunks to maximized entities group for the process sentence. We tried to use a black-list of the word the spacy put inside of the entity (e.g. 'the'). In the end we decided that this type of error it not really mistake of our model, because is answer is still correct for the pair of entities, so we stop compare the text of the entities and start to search for the best containment between the answers and the prediction.

### Spacy tagging mistakes

We found that spacy was tagged wrongly some fields we used in our model. e.g. some entity that their "ent\_type" should be 'GPE' tag with tags like 'PRESON' or 'NORP'. Because we think the 'ent\_type' is the most effective feature for 'live in' relation, these mistakes are critical for our model performance. We tried to use Wikipedia API to double check spacy tagging. It was a trade-off with the time that are model runs, because every query for Wikipedia have a delay time to get back an answer. we failed to achieve a major improvement in the tagging correctness to justify this delay. Because of this - we dropped this approach.

After we tried to use other tools, and also failed to achieve a major improvement with them, we decided to hope our model could handle this mistake alone.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Relation | Dev Recall | Dev Prec | Dev F1 | Test Recall | Test Prec | Test F1 |
| Live in |  |  |  |  |  |  |