Assignment 2 Report

Thomas Reolon - 221247

This report is only an introduction: more detailed explanations in the <u>notebook</u> (point 0. in the notebook corresponds to part1)

part 1 - Evaluating SpaCy at token & chunk level

I loaded the dataset with a function called get_data, which returns the sentences with their annotations + the sentences in string format.

```
>>> data, corpus = get_data()
>>> data[0]
[('SOCCER', 'NN', 'B-NP', 'O'), ('-', ':', 'O', 'O'), ('JAPAN', 'NNP', 'B-NP', 'B-LOC'), …]
>>> corpus[0]
'SOCCER - JAPAN GET LUCKY WIN , CHINA IN SURPRISE DEFEAT .'
```

Then I used spacy to extract the named entities. Unfortunately spacy's tokenizer creates tokens in a different way wrt. conll2003 dataset. To solve this problem 2 solution were implemented: OPTION1 (elegant) substitutes spacy's tokenizer with a whitespace tokenizer; OPTION2 (slightly better performances) after having processed the tokens with spacy's tokenizer, a function maps spacy's tokens to conll tokens. **Figure 1** shows the results obtained with each method.
Before computing the metrics I had to convert spacy's NER annotations into CoNNL2003 annotations, so I wrote a custom function for this purpose (more info in **Appendix A** of the <u>notebook</u>).

The results of this part of the exercise showed that chunk level performances are a little worse wrt. token level performances.
note: most of these functions are components that can be added to spacy's pipeline and store the computations inside an attribute (of Token or Doc).

part 2 - Grouping of entities

Chunks from Doc.noun_chunks can contain multiple entities, so I wrote a function that checks if an entity is contained in one of these chunks, if it is we add that entity to group[chunk_id], else the entity will be put in a separate 'private' group.

The results (**Figure 2**) showed that most of the entities are not grouped and that groups of 2+ entities often include PERSON.

part 3 - Fixing Segmentation

A possible post processing step would be to adjust the span of the entities. To do so we expand the entities through the compound dependency, if two entities overlap after the expansion, they are merged.

This process greatly reduces the final number of entities by merging them. Unfortunately, this approach seems to lead to worse performances, as we can notice from **Figure 1**. How expansion affects the token_level tags can be seen in **Appendix C** of the notebook.

	p	r	f	s
ORG	0.438	0.300	0.356	1661
PER	0.718	0.579	0.641	1617
LOC	0.771	0.668	0.716	1668
MISC	0.719	0.540	0.617	702
total	0.663	0.519	0.582	5648

Default (OPT1: WhiteSpaceTokenizer)

	p	r	f	s
PER	0.735	0.595	0.658	1617
MISC	0.725	0.547	0.623	702
LOC	0.783	0.713	0.746	1668
ORG	0.459	0.302	0.365	1661
total	0.683	0.538	0.602	5648

OPT2: spacy2conll tokens

	p	r	f	s
ORG	0.422	0.290	0.344	1661
PER	0.713	0.573	0.635	1617
LOC	0.760	0.658	0.705	1668
MISC	0.712	0.534	0.610	702
total	0.653	0.510	0.573	5648

Expanding compound dep

	p	r	f	s
PER	0.456	0.288	0.353	1617
MISC	0.635	0.660	0.647	702
ORG	0.557	0.481	0.516	1661
LOC	0.753	0.779	0.766	1668
total	0.616	0.536	0.573	5648

Transformer (Appendix B)

Figure 1. Four different procedures to compute NER entities and evaluate them on conll2003/test.txt. OPT1 and OPT2 refers to part1; Expanding compound refers to part 3; Transformer refers to Appendix B.

```
entities sharing a group: 423    entities in a 'private' group: 7080
CARDINAL          : 1821
GPE               : 1269
PERSON            : 1043
ORG               : 993
DATE              : 940
NORP              : 302
MONEY             : 144
ORDINAL           : 115
TIME              : 98
CARDINAL-PERSON   : 91
PERCENT           : 86
QUANTITY          : 79
EVENT             : 62
LOC               : 51
NORP-PERSON       : 45
```

Figure 2. top 15 groups of entities for part 2.