EDAN20 Assignment 5
Subject–verb–object triples

Matilda Andersson, ma4748an-s

October 2020

1 Introduction

The purpose of this laboratory report is to describe and comment on the subject-verb-object extractor composed in this assignment. The extractor composed used a word function looking at relations between words in a sentence and found the connected entities, the subject and the object, by looking at the word relations, represented as the verb. Using this

$$subject - (verb) - > object$$

relations and dependency parsing we can then build a knowledge base. The corpora used for the assignment were the Universal Dependencies (UD) framework which is a fully annotated dataset with consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages, all presented in a CoNLL-U format.

2 Process description

The following subsections contain a more in depth description of the process for creating the subject-verb-object extractor.

2.1 Examining the annotation

As a first step in this assignment we looked at the two first sentences in the corpus and visualized them using the conllu.js tool linked in the assignment description. The results we got can be seen in figure 1. Subsequently, we applied a dependency parser using the Langforia pipeline and obtained another parsed visualization of the sentences which can be found in figure 2.

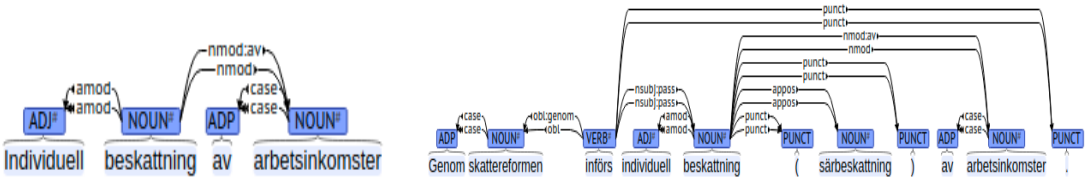

Figure 1: Graphical representations of the sentences using the conllu.js tool.

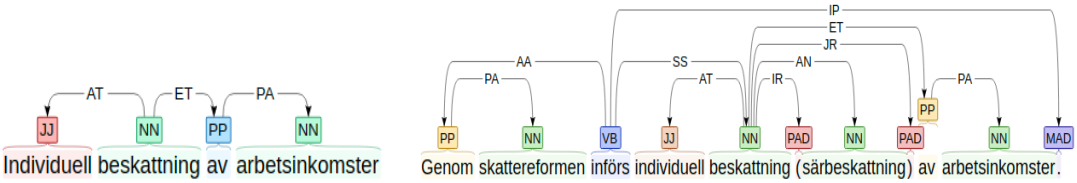

Figure 2: Graphical representations of the sentences using the Langforia dependency parser.

Although using different word tags, the graphs looks over all pretty similar. The conllu.js parser has a more detailed and descriptive arch presentation which depicts all the entity connections bound together by the relation-function. These relations can also be found in the Langforia parsed graph, but only by hoovering over the word tags and on your own following the described relations step by step.

2.2 Data preparation

In order to efficiently extract the word triples we started of by importing the Swedish training corpus and formatting the sentences and words into dictionary representations where the keys were the words indexes in the sentence. The training corpora used used later on in the assignment for the other languages also went through this process and ended up being stores in dictionary representations.

2.3 Subject-verb-object extractions

This part of the assignment focused on writing a function for extracting the word triples. Looping through each sentence we started by locating the words annotated as subjects and when one was found we noted its 'HEAD', or the verb the subject corresponded with. Then, looking through the same sentence again now searching for words labeled as 'objects', we checked the located words against the subjects 'HEAD' and it both words were pointing at the same head (root word), we had found a triple. All discovered triples were stored in a master dictionary with the triples as keys and its frequency count as value. Running the Swedish, English, Russian and French word corpora through this function we could extract the N most frequent tuples.

2.4 Italian, Danish and Japanese extractions

For the optional language analysis I chose to extract the pairs and triples for the Italian, Danish and Japanese text corpora. Doing this resulted in the code and results presented in figure 3 and 4.

```
corpus_dicts = []
ud_paths = ['ud-treebanks-v2.6/UD_Italian-ISDT/it_isdt-ud-train.conllu', 'ud-treebanks-v2.6/UD_Danish-DDT/da_ddt-ud
for path in ud_paths:
    sentences = read_sentences(path)
    formatted_corpus_dict = convert_to_dict(split_rows(sentences, column_names_u))
    corpus_dicts.append(formatted_corpus_dict)

it_corpus_dict, da_corpus_dict, ja_corpus_dict = corpus_dicts[0], corpus_dicts[1], corpus_dicts[2]

it_pair, it_triples = extract_pairs_and_triples(it_corpus_dict, nbest)
da_pair, da_triples = extract_pairs_and_triples(da_corpus_dict, nbest)
ja_pair, ja_triples = extract_pairs_and_triples(ja_corpus_dict, nbest)
```

Figure 3: Code for extracting pair and triples from Italian, Danish and Japanese corpora.

```
[(('nome', 'qual'), 36), (('che', 'hanno'), 33), (('presidente', 'chi'), 32)]

[(('individuo', 'ha', 'diritto'), 22),
 (('sigla', 'significa', 'cosa'), 14),
 (('proprietario', 'ha', 'diritto'), 11)]


 [(('det', 'er'), 40), (('vi', 'har'), 31), (('han', 'siger'), 21)]
 [(('det', 'drejer', 'sig'), 8), (('de', 'taler', 'tyrkisk'), 3), (('han', 'gjorde', 'det'), 3)]


[(('こと', '多い'), 14), (('こと', '可能'), 11), (('性', 'ある'), 9)]
[(('こと', '意味', 'こと'), 3), (('側', '起こし', '訴訟'), 2), (('年度', '目指す', '契約'), 2)]
```

Figure 4: Pairs and triples with highest frequency count in Italian, Danish and Japanese.

2.5 Resolving the entities

As a last task we were to go through a similar process as for the aforementioned triples, only this time extracting the relations involving named entities, that is where both the subject and the object were proper nouns. We used the same approach as for the first type of word extraction but included a step where the subjects' and the objects' position of speech were controlled to be labeled as 'PROPN', proper noun.

3 PRISMATIC

In the second step of the PRISMATIC pipeline, the Frame Extraction step, the word labeling used is very similar to the one used in our assignment for labeling the subject, object and verbs. Furthermore, in the third step of the PRISMATIC pipeline they focus on conducting text analysis by dissecting frames along different dimensions, one of them being frame slicing after the S-V-O structure where S - subject, V - verb, O - object are specified as a cut. These frame-cut structures are identified over all frames (note that we did the same cut, or extraction, over all sentences) and frequency information for each cut is tabulated, a step also being performed in our assignment. Identifying the relations and dependencies between words becomes essential in creating a tool that analyzes specific words typical relation to other words, such as its predicates and arguments.