Unitex for NER - exercises

Cvetana Krstev, JeRTeH & University of Belgrade

Denis Maurel, Université de Tours, Lifat

Starting

Start Unitex

Choose English as the working language

Choose text (Menu Text →Open): **ENG18900_Doyle_noHeader_xml.txt**

- It is a text from English ELTeC sub-collection (Arthur Conan Doyle's novel „The sign of four“); header was removed for easier manipullation (for exercises)
- XML tgs are there, but the suffix .xml was replaced by .txt

Processing the text

Answer „yes“ to the question „Do you want to preprecess the text“

Uncheck the boxes „Applay graph in a MERGE mode“ and „Applay graph in a REPLACE mode“

Press „GO“ button

Applying dictionaries

Open Text→Apply Lexical Resources

Check if three dictionaries are set by default: **dela-en-public, CasENAmbiguites-, CasEnIstexDico**;

If not:

- select dictionaries that are not (Ctrl+click)
- Press „Set default“

- Then press „Apply“

If yes:

- everything is OK, exit

Apply lexical mask <N+FirstName>

Open Text→Locate Pattern

Type in „Regular expression“ field: <N+FirstName>

In „Search limitation“ choose „Index all occurrences in text“

Press „Search“ button

You should obtain 391 matchs

Press „OK“, then press „Build concordance“

What can be seen in concordances?

There are positive matches: Abdullah, Abel, Jonathan, etc.

There are also false matches: London, India, An, Even, etc.

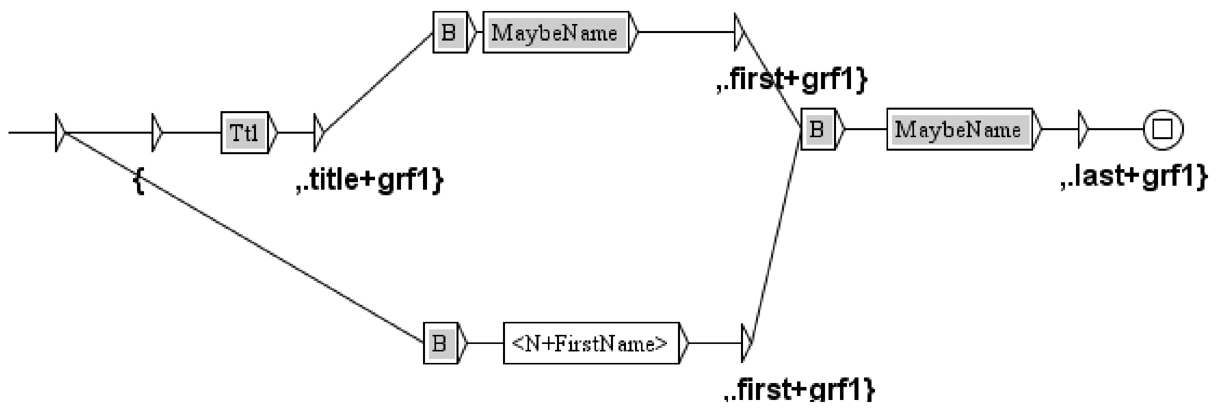
More precise querries are neccesary

We will make a graph that uses the marker „FirstName“ and the context

Open new graph with FSGraph→Open

Make this simple graph

Upper path: Context (some title) followed by a word in upper-case word (most likely a first name) and another upper-case word (most likely a surname)



Lower path: There is no context, but there is aword tagged as a first name (according to dictionaries) followed by an upper-case word (most likely a surname)

The use of subgraphs

In order to invoke a subgraph you have to type in a box a colon and then the name of a subgraph

:B

:Ttl

:MaybeName

This boxes are red until you save your graph in the same folder in which this sub-graphs are stored

This you will do when you finish with your graph, then the boxes will turn gray.

The output of a graph

You want to make a lexical tag out of the recognized sequences.

The begining of a tag is an open curl bracket: <E>/{

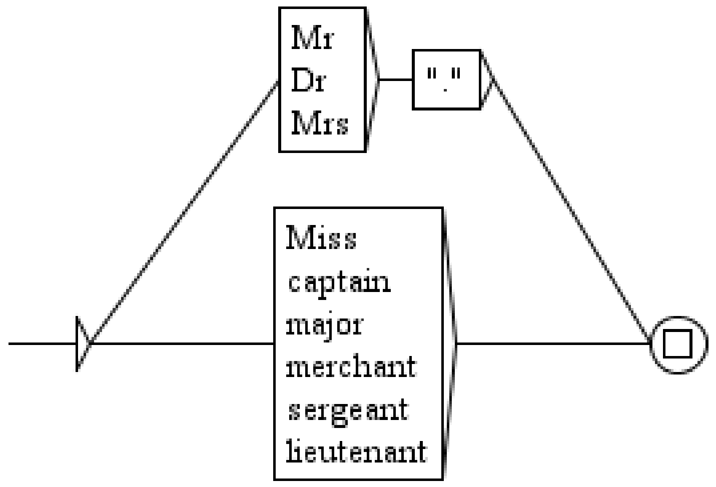
then comes the recognized sequence

the end of a lexical tag is its type, and the closing curl bracket: <E>/,.first+grf1}

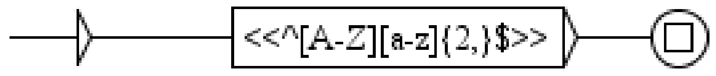
Beside a type we can add the name of the graph, it is good for the contol, to know which graph has tagged what

What the sub-graphs do?

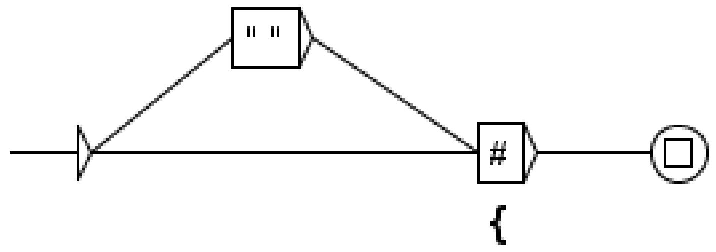
Alt+click to open a subgraph



title.grf



MaybeName.grf



B.grf

Have you produced the graph?

Save it with some name (xxx.grf)

FSGraph→Save as...

Now you can use it:

- Text → Locate Patterb
- Use this graph for search
- Produce concordances with Grammar outputs not taken into consideration (by default), you will obtain **96 lines** (some are OK, some are not)
- Produce now cocordances in the Merge mode, you obtain again 96 lines of concordances, but recognized sequences have the form of lexical tags:
- {Abdullah,.first+grf1} {Khan,.last+grf1}

What else?

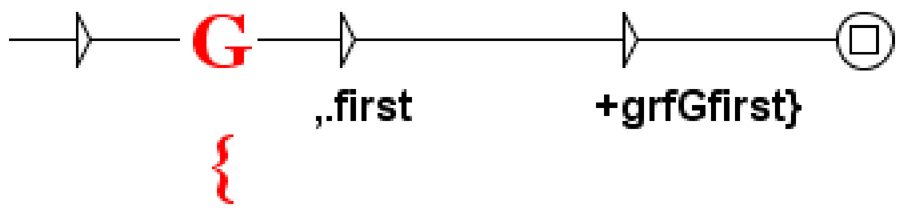
We proceed with graphs that try to recognize as many names as possible (as correctly as possible)

This graphs are already prepared, we will just use them

How to see them? We will open an already prepared cascade:

- Text→ApplyCassys cascade
- select cascade **TS_analysis.csc**
- press button Edit

We will look at next graphs, one by one (you can see each of graphs by selecting one and pressing a button View)



2. Gen_first

If something was recognized as a first name in a broader contex (graph **T_First_Upperfirst**), we assume it is correct and we tag all occurrences of that as a first name as well.

How can we see what this graph has done?

Open in Unitex the intermidiary file:

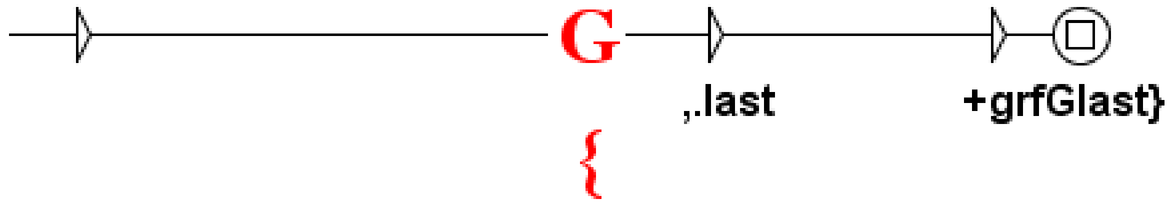
Text→ENG18900_Doyle_noHeader_xml_csc→ **ENG18900_Doyle_noHeader_xml_1_0.snt** (it is the result of the first graph in the cascade)

search for the pattern **<first>** (Text→Locate Pattern) – you will obtain 41 mathes

Now open file **ENG18900_Doyle_noHeader_xml_2_0.snt** (it is the result of the second graph in the cascade)

Again search with the pattern **<first>** – you will obtain 95 results

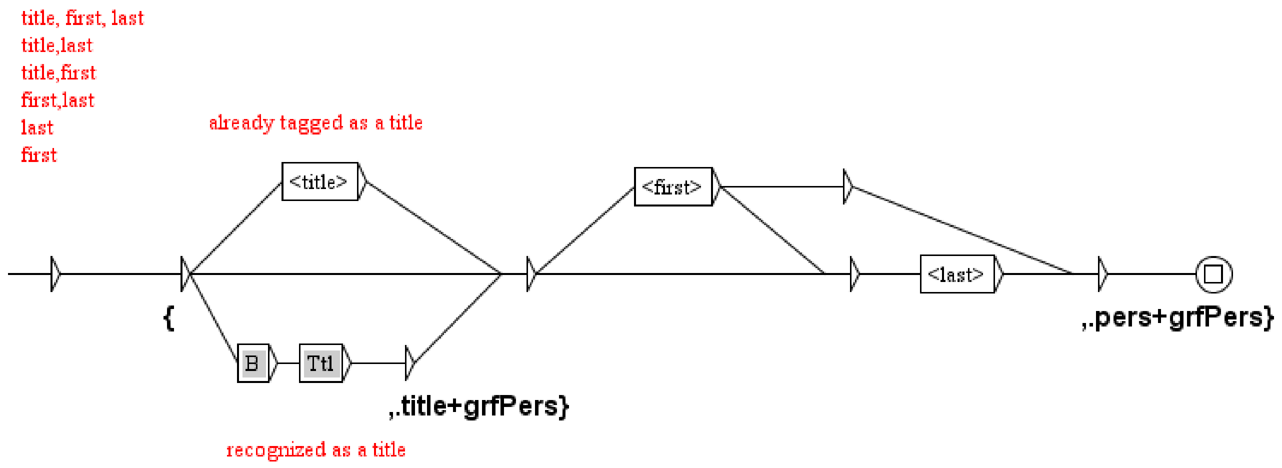
The second grpah has „generalized“!



3. Gen_last

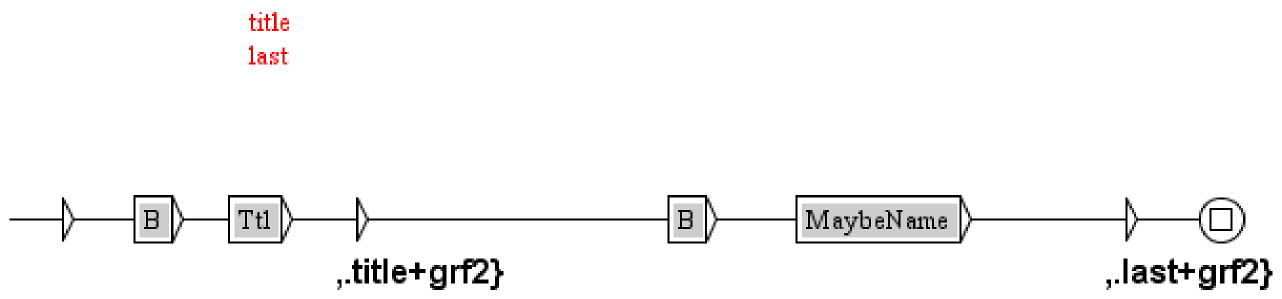
It has the same function as the previous graph but for the „last“ name

We can check what it has done in the same way as for the previous graph, only we look in the next intermediary file.



4. Pers

We tag a „person“ everything we had recognized so far



5. T_Last

We try to recognized what is left – title followed by something written in the first upper-case (probably a surname)

The end of the cascade

The last two graphs were already used:

- You can use the same graph several times in the same cascade if that suits your purpose

Gen_last generalizes the newly recovered surnames

Pers groups all elements belonging to a person named entity

We will now apply this cascade:

- Text→ApplyCassys cascade
- select cascade **TS_analysis.csc**
- press button Launch

What has been achieved?

Open the results of the cascade:

Text→Open→ENG18900_Doyle_noHeader_xml_csc.raw

(you have to go back to the Corpus folder)

„Do you want to preprocess the text“, you answer „No“

This version of the result uses lexical tags that you may use for the search

Text→Locate Pattern→Regular expression: <pers>

You will obtain 600 matches, you can produce concordnces to see what is there.

How can you obtain XML version of results?

This was a cascade for the recognition (the analysis phase)

Now you have to open another version of the result of the first cascade:

Text→Open→ENG18900_Doyle_noHeader_xml_csc.txt

It is in the Corpus folder and again you do not want a preprocessing

We will open an already prepared cascade:
- Text→ApplyCassys cascade
- select cascade **TS_synthesis.csc**
- press button Launch

What have we got?

You can open now the newly prduced file **ENG18900_Doyle_noHeader_xml_csc_csc.txt** (from the catalog Corpus) in Notepad++

You will see at the beginning:

<pers><first>Sherlock</first> <last>Holmes</last></pers>

and later on:

<pers><title>Miss</title> <first>Mary</first> <last>Morstan</last></pers>

What about false recognitions?

Many of them can be avoided, how?

First, dictionaries have to be improved to contain more proper names with more refined markers.

Even without that much can be done.

You can notice that there are lot of „urban places“ mentioned in this novel. Using appropriate triggers (street, square, etc.) and the most appropraite order of graphs can remove many ambiguities.

The same goes for toponyms (island, isles, etc.)

Conclusion

This exercise session was prepared to give you the impression how things are done and what can be achieved.

There is a comprehesive manual to help

There is a forum to answer your questions

We are also available in order you need help