Project Text Analysis – Final Project Wikification

Wietse de Vries
wietse.de.vries@rug.nl

Jasper Bos
j.k.bos.1@student.rug.nl

25 May 2020

As we discussed in class and in the previous assignments, the final project is concerned with *wikification*. Wikification is the task of detecting entities of interest in some text, and linking them to a database. The target database that we are using is Wikipedia, and the data we are using is that from American newspapers that you have previously annotated (Assignment 4).

In this document you will find some additional information regarding the data, the procedure, and what is required that you submit.

hint

For an overview of the task and its main components, and ideas on how to tackle issues that you most likely will have to deal with, check the **slides from Week 5**.

Please read carefully: I will open two assignments for this project on Nestor. On the main one, where this pdf document is, you can upload all of the project work, while there will be a separate one for the report only, as that will be graded separately for each of you. Technically on Nestor, but also in reality, the first one is a group assignment, the second one is an individual assignment.

- use Nestor for submission
- **Deadline: Friday June 11th, 2021, 23:59**
- **Presentations: Thursday June 17th and Friday June 18th**.
- hand in:
  - (on assignment "Project" on Nestor) your wikification system. This will have to be a python script which produces output files in exactly the same format as those you have produced manually. Also include the `server.properties` file you used for CoreNLP.
  - (on assignment "Project" on Nestor) the output files as produced by your system, that is those that you are comparing to the gold standard.

- (on assignment "Project" on Nestor) your evaluation script, that is a version of `measures.py` which will produce the evaluation measures required by this project (see Section 2.2 below).
- (on assignment "Report" on Nestor) a report (`.pdf`) to be written **individually** which contains your system's description, any comments you want to add, and highlights which portions you have worked on mostly. This should not be longer than 1200 words. You will received a template soon.

1. Task and Data

For Assignment 4 you had to produce some *annotated data*, including entities and Wikipedia links. As a reminder, these were the entities:

- **Country/State** — tag `COU`. All country and state names.
  Examples: France, Alaska, The Philippines, Tuscany, Burkina Faso, etc
- **City/Town** — tag `CIT`. All cities and smaller towns.
  Examples: New York, Rome, Groningen, Berlin, Zuidlaren, etc.
- **Natural places** — tag `NAT`. This includes all natural places such as lakes, mountains, volcanoes, rivers, seas, oceans, forest, etc.
  Examples: Mississippi River, Etna, Amazonia, The Pacific
- **Person** — tag `PER`. This includes all persons. Limit this to proper nouns.
  Examples: Bill Clinton, Johnny Depp, Bruce Springsteen, etc.
- **Organization** — tag `ORG`. This includes companies but also all sorts of organisations
  Examples: Google, ONU, Mercedes Benz, etc
- **Animal** — tag `ANI`. All animals.
  Examples: dog, crocodile, rabbit, cat, etc.
- **Sport** — tag `SPO`. All sports.
  Examples: football, soccer, baseball, tennis, etc.
- **Entertainment** – tag `ENT`. This includes any books, magazines, films, songs, concerts, etc.
  Examples: The Wall Street Journal, Ghostbuster, The Bible, Holes, Born to run, etc

What a file looks like is given in Figure 1.

Well done to all for making this dataset. The data that all of you have produced for Assignment 4 is now used both for developing and for testing your system, as explained below. Please, remember that you should be able to re-use all of the code that you produced for the assignments, as it was indeed intended to be useful for the final project, too.

```
0 7 1001 Burkina NNP COU http://en.wikipedia.org/wiki/Burkina_Faso
8 12 1002 Faso NNP COU http://en.wikipedia.org/wiki/Burkina_Faso
13 14 1003 ( NNP
14 22 1004 formerly RB
23 28 1005 Upper NNP COU http://en.wikipedia.org/wiki/Upper_Volta
29 34 1006 Volta NNP COU http://en.wikipedia.org/wiki/Upper_Volta
34 35 1007 ) NNP
36 44 1008 achieved VBD
45 57 1009 independence NN
58 62 1010 from IN
63 69 1011 France NNP COU http://en.wikipedia.org/wiki/France
70 72 1012 in IN
73 77 1013 1960 CD
77 78 1014 . .
79 87 2001 Repeated NNP
88 96 2002 military JJ
97 102 2003 coups NNS
...
```

Figure 1: Format of gold standard file

2. Procedure

From the whole collection produced for Assignment 4, you are getting a portion which amounts to approximately 30% and which contains all of the hand annotated information (as well as the original files, they are still in the directories). This is uploaded as part of the assignment on Nestor as a zip archive (`dev.zip`) You can use this portion to develop your system, remembering that you can always test your output against the gold standard file using the measures you already know about, to get an idea of how well you are doing. The remaining 70% will be used for testing your system, and you will only see it after you will be done with developing.[1]

2.1. Developing you system

The **input file** to your system should be the `.pos` file. This means that you do not have this time to pos-tag your data again, it's already done from Assignment 4. To that file, your system should add the classes and the links, wherever appropriate and produce the `.ent` files *automatically.*

As we said in class, your system can be very basic and produce just an output file that corresponds to `en.tok.off.pos.ent`, or, on top of the production of that file (which is compulsory as we need it for evaluation), can be integrated in a web interface where a sample text can be given, and an output annotated text is returned.

[1] In a standard evaluation setting, the test file would be stripped of the information you added manually, which the evaluator would keep to produce the measures themselves. However, in this case you can run the evaluation yourselves, so you will get the gold standard files, thus including the manual annotation, too. See Section 2.2 on how to use it.

hint

A demo for inspiration can be found here:

- `cogcomp.cs.illinois.edu/page/demo_view/Wikifier`

Please remember that your system will have to output an `.ent` file that in format looks exactly like the one in Figure 1 above.

Your system that will be tested on secret annotated *test set* should work like this:

```
$ python3 my_system.py path/to/en.tok.off.pos > path/to/en.tok.off.pos.ent
```

Once you are done, and in any case **by the deadline**, you can submit your final system and your report by uploading them on Nestor. If you upload your system before the deadline, please also send us an email to say that you're done. Also, remember that the system is jointly produced by the group, but the report has to be written **individually**. In the report you should describe how the system works, what was done to tackle the task, and also clearly specify which portions of the project you worked on most. Please note that an individual report is a document that you write completely by yourself. It's expected to about approximately two pages long (and in any case max 1200 words).

2.2. Testing your system

Once you have submitted your system and the report that describes what you have done, you will receive from us the *test set*. Because you can do the evaluation yourself, the test file will also be a gold standard file thus including the entities and the link columns. On that, you can run your system and evaluate it, using precision and recall for the entities, and accuracy for the links (correct or not correct). Note that the `.ent` files you are getting in the test set are only to be used to evaluating your system against them (they are your gold standard), but should not be in any way used as input to your system. No information contained in them can be used by the system when assigning entities. The only input to your system is the `.pos` file.

You should also produce a confusion matrix for the entities, to see what the most common mistakes were. In your presentation you will show the measures and the matrix. Please, remember that in order to compare your produced file and the gold standard file that you will get from us, including the confusion matrix, you should be able to run the same script that you developed to assess inter-annotator agreement (`measures.py`), as the format and the measures are the same. The only addition should be the calculation of accuracy for the links. To do that, please note that you will have to calculate also something which is like precision and recall. Specifically, you will need the following two measures, and report in your presentation both of them:

- the number of links you managed to get (correctly) out of the number of links you were supposed to get. You have to count how many links from the gold standard you were able to retrieve correctly (this is like recall);

- the number of links that are correct out of the total number of links that you found. You have to take all of the links your system found and count how many of those are actually correct (this is like precision).

Please, remember that at this point you shouldn't try anymore to change your system in order to get better results.

2.3. Showing your system

As a final part of this project, you will have to present your system to your fellow students and the instructors. Please, remember that each group member has to participate in the presentation. You will have to illustrate what you have done to tackle the task (what processing you've done, which tools you have used), and also present and discuss results on the test set. If you have set up a web page for interactive wikification, you are more than welcome to show a demo, too.

Presentations will last 15 minutes each. They will take place on June 17th and 18th, following the schedule you will soon find on Nestor. You will be required to attend all sessions, even after your group has already presented.