Retro-digitizing Turkish dictionaries using GROBID-dictionaries

Emrah Özcan

Retro-digitizing Turkish dictionaries using GROBID-dictionaries[1]

Emrah Özcan[2]

Dictionaries are invaluable resources about languages. Beginning from the famous Dīwān Lughāt al-Turk, many Turkish dictionaries were printed until now, and today, we still value these resources as they contain the traces of the history of the language, the society's situation at the time of the publication which also, directly or indirectly, reflected the policies imposed upon speakers. With this in mind, different editions of the same dictionary, published at different times, could bring new perspectives at how we look at the language. The fact that today's technology allows us, the researchers, the necessary tools to investigate more about the language we are studying, we need digital versions of dictionaries in order to do analyze deep into those resources; by searching and comparing the content. With the current technological tools in hand, we can deal with huge data as if we can deal with couple of print pages. We need couple of minutes, to do tasks that need days or weeks, if done by hand.

Thus, at this year's Lexical Data Masterclass 2018, I attended the GROBID-dictionaries workshop given by the core developer of the project, Mohamed Khemakhem, to use GROBID-dictionaries for retro-digitizing Turkish dictionaries.

GROBID-dictionaries work with OCRed PDF files. Thus, you need a good OCRed PDF version of the dictionary. Otherwise, PDF2XML engine will fail to recognize the characters or the page layout. So, I recommend you to choose a good OCR software, or try and see whatever options you have.

After providing the PDF, you need a decent machine, with enough RAM to handle the process. Choosing Linux or MacOS as the operating system will help you a lot, as Windows machines come with a bunch of alternations among different editions. You can also install Linux on top of your existing OS via dual boot.

After choosing the OS, you need to install a couple of applications on your machine; oXygen XML Editor and Docker.

---

[1] This paper is presented at the Lexical Data Masterclass 2018 Participants' Symposium, 3-7 December 2018, Berlin, Germany.

[2] Yildiz Technical University, Istanbul, Turkey - eozcan@yildiz.edu.tr

After you follow the steps in the given instructions, you may start experimenting with the GROBID-dictionaries by writing this command in you terminal (I will be using MacOS instructions here);

docker pull medkhem/grobid-dictionaries

When you *pull*, you will see your terminal downloading the necessary files from Docker container, aka *pulling* the Docker image. Then, you can get into the GROBID-dictionaries with your first command:

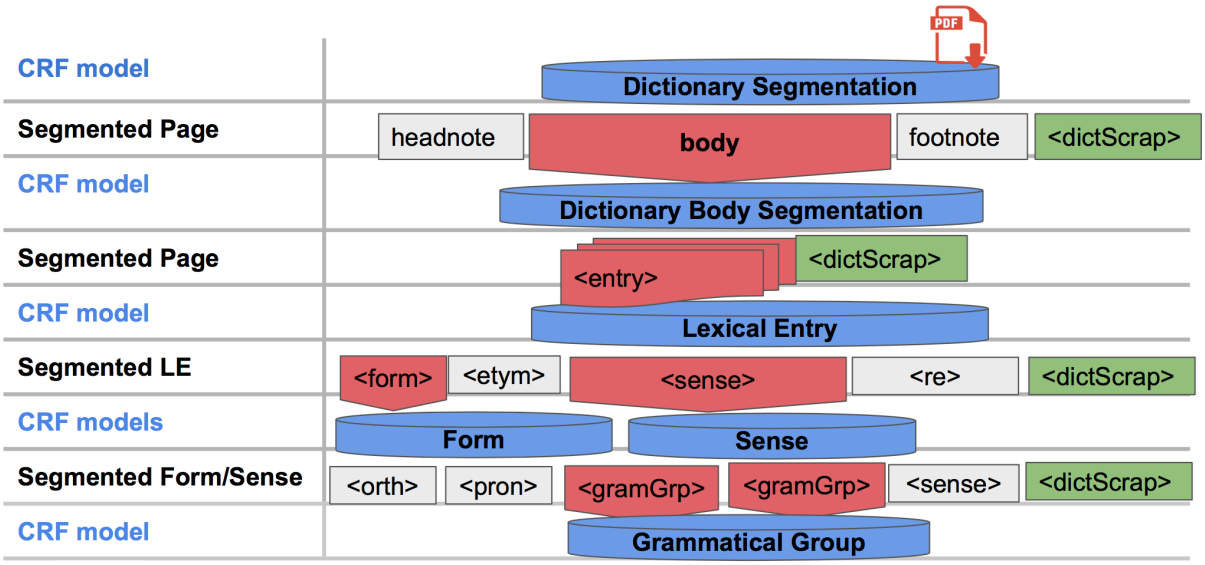
docker run -it medkhem/grobid-dictionaries bash

After you put the above command in your terminal, your prompter will look like this.

root@someRandomNumbersAndLetters: /grobid/grobid-dictionaries#

This shows you that you are in the GROBID-dictionaries image now as a user.

GROBID-Dictionaries is based on cascading CRF models. The diagram below presents the architecture enabling the processing and the transfer of the text information through the models.



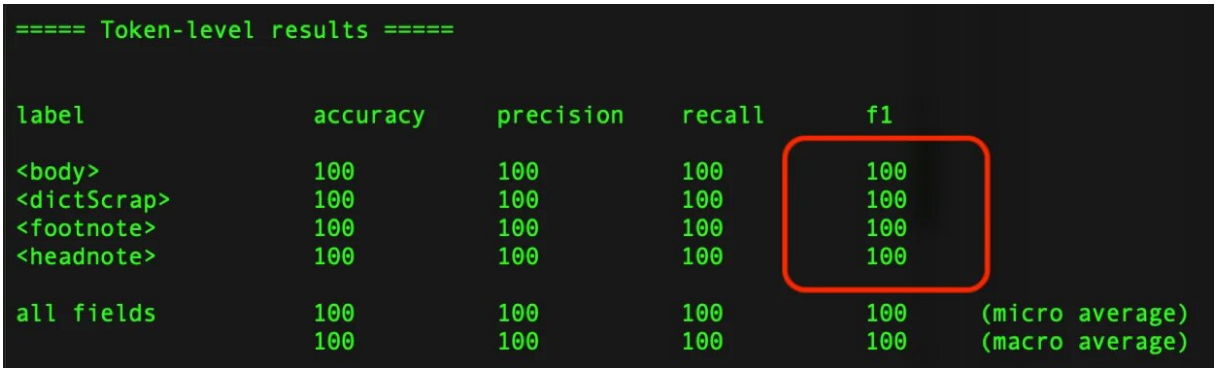
Cascading CRF models used in GROBID-dictionaries

Interestingly, the levels for models in GROBID-Dictionaries reflects the actual levels from the TEI in libraries recommendation. Models can be seen below:

- Dictionary Segmentation model,
- Dictionary Body Segmentation model,
- Lexical Entry model,
- Form model,
- Sense model,
- Etymology model,
    - For the first stage model of processing etymology information (EtymQuote model)
    - For the second stage model of processing etymology information (Etym model)

These models introduce the system the parts of the page layout in a dictionary and as you proceed within the models, you start annotating. After annotating the models, the system should be trained with 2/3 of the annotated pages, and the rest of the annotated files, 1/3, should be used for evaluation. Therefore, the system will use the files used as evaluation as a reference point to calculate an *f1* score. The higher the score, the better the results are. In other words, if results are calculated less than 97, then more annotated pages are necessary, such as pages with a variety of content, so the system will see more diverged samples and the results will get better after this training.

Here are the results from our training data. The first one is from the first model, Dictionary Segmentation.

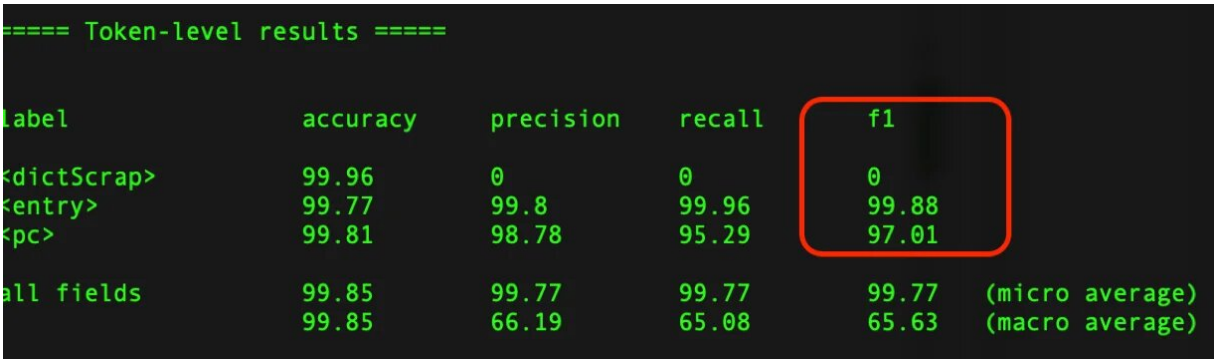


```
===== Token-level results =====

label                accuracy     precision    recall       f1

<body>               100          100          100          100
<dictScrap>          100          100          100          100
<footnote>           100          100          100          100
<headnote>           100          100          100          100

all fields           100          100          100          100     (micro average)
                     100          100          100          100     (macro average)
```

Dictionary segmentation *f1* score

As can be seen in the above picture, *f1* score is 100%. This show that the system was able to understand <body>, <dictScrap>, <footnote> and <headnote> tags correctly.

Next model is Dictionary Body Segmentation, and this model is a little bit less complicated than the first model as it contains only two tags; <entry> and <dictScrap>. Here is the result from the second model:



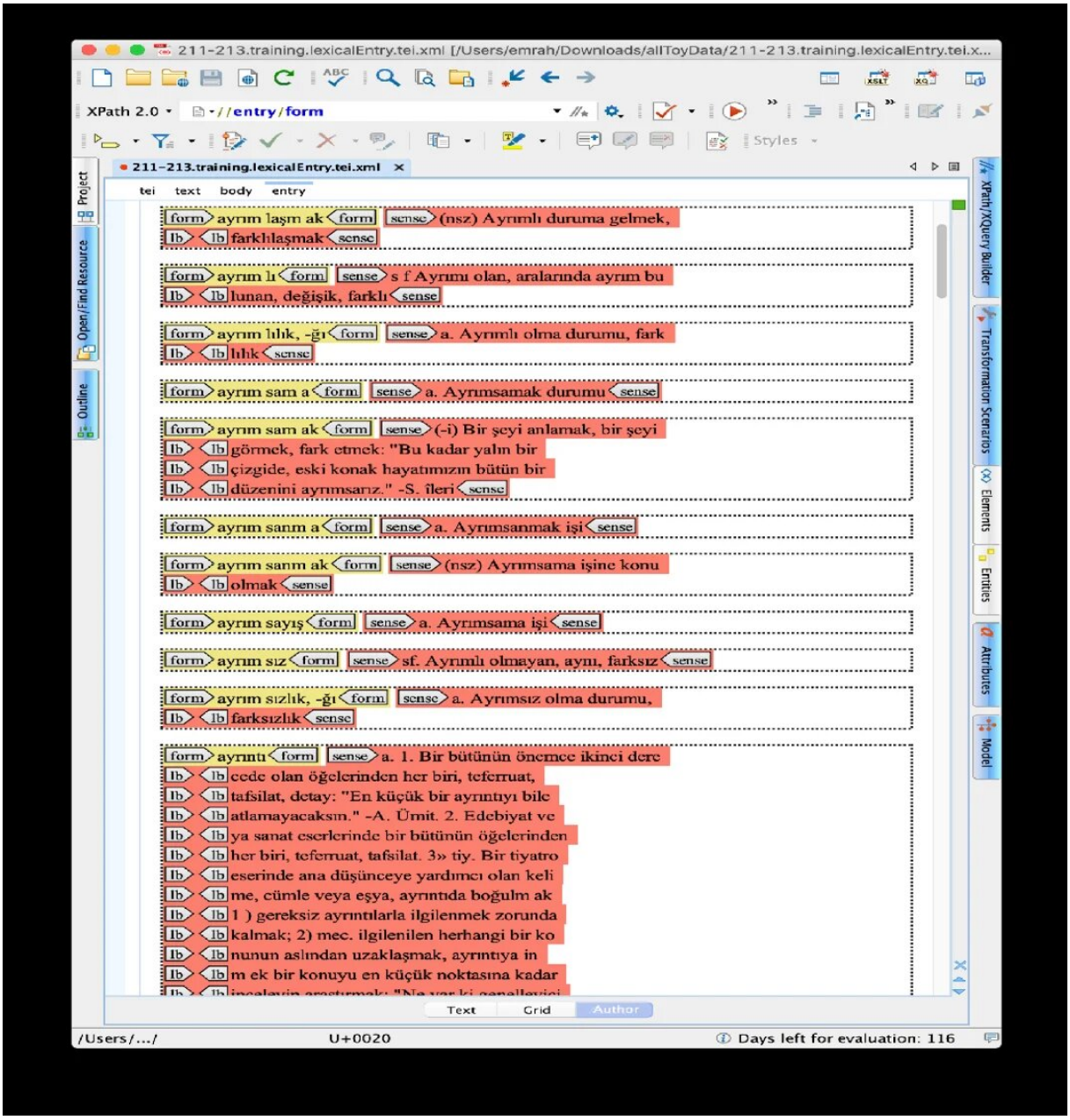
Dictionary Body Segmentation *f1* score

As it can be seen in the screenshot, *f1* score for the second model needs more varied annotation that contains <dictScrap>, since the score is 0. As it has been mentioned above, this indicates that the evaluation data contains no <dictScrap> tag, therefore system is unable to train. The solution for this is to add more annotation in the evaluation set that contains the mentioned tag.

Repeated procedure for each model will take place and after each procedures, model should be checked whether it is successful enough or not.

Another important point that needs to be taken care of while using GROBID-dictionaries is the quality of the OCRed PDF file. If the OCRed PDF contains problematic issues, the system will have hard time to understand to recognize the file correctly. This problem will make it harder for the system to work properly as one continues to the next models. As the model number increases, the system looks for more detailed information on the page layout. Therefore, it is advisable to use more than one OCR software to test and see if the OCRed PDF contains problems. Depending on the software, sometimes one type of problem could be seen in another OCR process and the vice versa. In our case, we used Adobe Acrobat and ABBYY FineReader softwares, and each had its own problems with the OCR process. Re-OCRization is needed if different softwares produces different results.

In the third model, lexical entry is being annotated. oXygen software's author mode is used for the annotation process. This mode helps the annotator as the system already provides the necessary CSS and NRG files. In the below picture, the author mode in oXygen can be seen during the annotation process.



oXygen Author mode is used for the annotation process

In case one annotates enough training data and evaluation data for GROBID-dictionaries, thinking that the OCRed PDF is clean, you could retro-digitize a dictionary containing thousands of pages by annotating dozens of pages for each and every model and help you to have a fully TEI compliant digital version of the print dictionary.

Even in cases where it is hard to have a clean OCR PDF, due to the physical appearance of the print dictionary, GROBID-dictionaries can still be used to have just the entries out of the print dictionary. Especially, if the project is only interested in comparing different versions of the print dictionaries, from the perspective of entry selection.

I would like to thank Mohamed Khemakhem for his invaluable help and patience for numerous questions during the GROBID-dictionaries session.

References:

- Mohamed Khemakhem, Axel Herold, Laurent Romary. Enhancing Usability for Automatically Structuring Digitised Dictionaries. *GLOBALEX workshop at LREC 2018*, May 2018, Miyazaki, Japan. 2018. 〈hal-01708137v2〉
- Mohamed Khemakhem, Luca Foppiano, Laurent Romary. Automatic Extraction of TEI Structures in Digitized Lexical Resources using Conditional Random Fields. *electronic lexicography, eLex 2017*, Sep 2017, Leiden, Netherlands. 〈hal-01508868v2〉