— Research Project Log —

Raven Timmer - 13974920

16/04/2025

Initialized the repository.

Standard initialization

Started testing by using entity Recognition.

I have started testing the entity recognition. Firstly I will try this using the pretrained huggingface model described in the paper: Batavia asked for advice. Pretrained language models for Named Entity Recognition in historical texts.

The results seem to be good. I will show the recognised entities in the following text:

> “Erasmus werd in J apan, waar het bij aankomst slecht terecht was gekomen, vastgehouden; ten gevolge van het uitblijven van de nodige herstellingen werd het geheel onbruikbaar; het werd in 1634 voor sloping verkocht. 282 Specx, Vlack, Van Diemen en Van der Burch II, 7 maart 1631 ’t Schip den Gouden Leeuw is onbequaem bevonden omme met retouren naer ’t vaderlandt over te gaen, jaa is inwendich soo vergaen, dat, onaengesyen de handt daer extra-ordinaris aengehouden is, niet langer in ’t vaerwater sal connen continueren. Schiedam is op de Cust ende wert oudt, sulx dat Uw Edn bij ’t vorige als uuyt”

Gives the (post-processed) entities:

```
[{‘entity’: ‘PER’, ‘text’: ‘Erasmus’, ‘score’: np.float32(0.52735126)},
{‘entity’: ‘LOC’, ‘text’: ‘Japan’, ‘score’: np.float32(0.9614511)},
{‘entity’: ‘SHP’, ‘text’: ‘Specx’, ‘score’: np.float32(0.67104995)},
{‘entity’: ‘SHP’, ‘text’: ‘Vlack’, ‘score’: np.float32(0.641053)},
{‘entity’: ‘PER’, ‘text’: ‘VanDiemen’, ‘score’: np.float32(0.9959038)},
{‘entity’: ‘PER’, ‘text’: ‘VanderBurchden’, ‘score’: np.float32(0.7346338)},
{‘entity’: ‘SHP’, ‘text’: ‘GoudenLeeuw’, ‘score’: np.float32(0.9601866)},
{‘entity’: ‘LOC’, ‘text’: ‘Cust’, ‘score’: np.float32(0.90481424)}]
```

22/04/2025

Extracted all dates from the original National Archives xml file (voc_inventory.xml) and exported them to inventory_dates.txt.

Dates that are considered ranges are saved as a tuple containing the start and end year. Anything that is not a year is discarded, meaning that 1720 May 10 ⇒ 1720.

Any references to centuries are regexed to a fitting year, so: 18e eeuw ⇒ 1700

The data seems to be complete but is to be finetuned based on how it will be used later.

The program now also saves a dump of the dictionary to the file: Inventorydates.pkl which can be used to load the data back into a dictionary. This is done using the pickle library. The keys are the years, and the values is a list of each entry that corresponds to that year.

24/04/2025

A first version of the searching is working. It makes use of the native Knaw API documented here: https://gloccoli.tt.di.huc.knaw.nl/swagger#/ It is able to use the dictionary created before to search for word combinations in a range around a given year. For example “Gouden AND Leeuw” around 1633 return 155 hits and around 1674 it returns 11 hits.

Added a batch search function that can request all of the results by splitting them up into multiple smaller requests. For now I will limit the max number of results to 200, as to not overload the server. This can be changed later if needed.