Improving readability and searchability of documents provided by the dutch government under the WOB

Author: Justin Bon
Information Studies - Data Science
University of Amsterdam
justin.bon@student.uva.nl

Supervisor: Maarten Marx
University of Amsterdam
M.J.Marx@uva.nl

ABSTRACT

Freedom of Information is an important right to give people in a democracy. It can increase transparency of a government and hold said government accountable for its actions and decisions. Documents that the dutch government provides under its version of a Freedom of Information Act are often not readable and searchable for a computer. This research will try to improve this readability and searchability of government documents. This will be done by using Optical Character Recognition to retrieve text from the documents. Then, as much as possible meta data will be extracted as well as named entities. With this, documents can be searched and connections between named entities can be found. The results will be visualized and made interactive in a web application. The data used in this thesis comes from a freedom of information request about the dutch governments handling of the Corona virus pandemic.

KEYWORDS

Named Entity Recognition, Optical Character Recognition, Meta Data Extraction, Co-occurrence Network

GitHub

https://github.com/JustinBon/thesis/

1 INTRODUCTION

The "Wet openbaarheid van bestuur" (Wob) is the dutch equivalent to the Freedom of Information (FOI) Act. It was put in place in the year 1991 with the purpose of giving citizens insight to the inner working of the government. This is done to promote participation in the democracy and it gives citizen a measure of control over the government [2]. A study done on the UK FOI act has shown that it works in practise by by showing that it has achieved its two core objectives of creating more transparency and accountability [16]. Therefore this paper will not look into the effectiveness or importance of the Wob because it is well established that FOI is important to a democracy and should be a fundamental right [9].

This thesis will try to solve a specific problem that the Wob has: the low machine readability and high quantity of data. Often times the result of a Wob request is thousands of pages of data in an un-indexable format, usually in the form of PDF documents. The purpose of this thesis, then, is to gauge the possibility to partially automate the processing of the large amounts of data that some of the bigger Wob requests return. This will be done by creating a proof of concept tool that will extract the text from the documents using Optical Character Recognition, extract relevant meta data from the documents (title, document type, date, etc.), use named entity recognition (NER) to find mentions of names, companies, government instances etc. and show potential connections between them. The goal of this is to give journalists of other interested parties a quick and easy solution to search through otherwise unsearchable data. To meet this goal, a interactive web application will be made with the results of the meta data extraction and NER. The specific data used for this thesis is the result of a Wob request about the handling of the dutch government pertaining to Coronavirus pandemic.

1.1 Research question

The main research question is: *How much can we improve the machine-readability of the documents the Dutch government provides when fulfilling requests made under the Freedom of information Act (WOB)?* This process consists of four things: extracting the text from the documents, splitting the documents into separate files, retrieving correct and relevant meta data from these separate files, and extracting named entities as a means of knowledge extraction. The focus of this thesis lies with the third and fourth point: extracting meta data and named entities. The goal is to create a program to do this automatically while remaining as accurate as possible. Two questions have been posed to help reach this goal:

(1) To what extend can relevant and correct meta data be retrieved form the data?
    - How can pattern recognition be used here?
(2) To what extend can Named entity recognition be used as a means of knowledge extraction?
    - How connections between named entities be analysed?
    - How can a co-occurrence network be used here?
    - How can the co-occurrence network be visualized?

2 RELATED LITERATURE

2.1 Meta data extraction

Meta data can be defined as "data about data"[12]. It can be a lot of things but usually it is data attributes that describe, provide context, indicate the quality, or document other object (or data) characteristics[6]. It can make it easier to sift through documents if there is meta data to search for [12]. Extraction of meta data if it is not readily available therefore, is important when working with large amount of data and documents.

Automation of the extraction of meta data can be done in two main ways: algorithmically or using machine learning. If the data has a standard format, it can be better to manually look at the structure and write a program to retrieve the meta data from the places where it should be [5]. An example of this is an HTML page. Most of the time the title of the web page can be found within the title tags. Meta data from documents in XML format can also be

retrieved this way. If there is a default place the meta data can be found, machine learning isn't necessary.

If there is no default structure to the data from which meta data needs to be extracted, machine learning can be used. A study done in 2003 used a support vector machine (SVM) found that it could improve meta data extraction performance [7]. SVM's are a form of supervised machine learning however more recent studies show that unsupervised machine learning models can also be very effective. A study done in 2020 found that their unsupervised deep neural network outperformed their rule based algorithm and SVM by a significant margin [13].

2.2 Named entity recognition

Named entity recognition (NER) is an important step in any Natural Language Processing (NLP) pipeline the purpose of which is to detect and classify named entities in a given text [15]. Entities, in this context, are named people, organizations, locations, government agencies etc. NER is a difficult task however to preform automatically. The difficulty lies in that there are few constraints for what can be a name and the relatively few labeled data sets that are available [8]. It can be challenging to generalize from the small amount of sample data. The easiest way to preform NER is to make use a "gazetteers" which is like a dictionary or index with a large number of entities already defined. Using gazetteers was early on in 1999 already identified as a potential bottleneck and that NER models without gazetteers was not only possible but compete in performance with NER models that did use it [10]. For this reason, the most current NER models use machine learning with labeled data to train and evaluate their models [8]. The model used in this paper, spaCy NER, uses a neural network to detect and classify entities. These methods have been shown to produce state-of-the-art models [8].

2.3 Co-occurrence network

A co-occurrence network is a network which shows when two entities occur together in the same context. It is used in a variety of fields like microbial research, lexical choice [1], and social structures [11]. The context of social structures is of most interest for this paper as it best reflects the named entities and documents format that will be extracted from the Wob request data. In social structures people can be represented as nodes in the network. An edge then exists between them if they co-occur in the same document [11]. Other research has been able to generate a lot of useful insight from these types of networks. It can, for example, be used to find clustering with groups of people and entities [3]. Besides that, if visualized, the network can be used to compare different nodes or reveal information about internal relationships between nodes [3].

3 METHODOLOGY

3.1 Data

The data that will be used for this research comes from a Wob request and consist of files about the dutch governments handling of the Coronavirus pandemic. All of the data comes in the form of PDF files. There are a total of 367 files of which 56 are inventory lists. The inventory lists contain information about what types of documents the files consists of. These files can be used to get an overview of the types of documents that can be found in the files and help indentify them. These files are, on average 21.804 MB in size totaling 7.8 GB for the complete dataset. Combined, there are a total of 32576 pages in the files. The contents of the PDF's have some variety. They fall into four categories:

- Information presentations of different ministries or companies
- Official government documents
- Reports
- lists of emails send or received by government officials or other smaller documents like memo's

Some of the files contain a compilations of multiple other documents, like email conversations.

In preliminary data analysis it was found that of the 367 documents 144 are completely unreadable. For these files another method needs used for text extraction. For the files that did contain readable text, there was only an average of 30.4 words per page and 3739.2 per file.

As for the types of meta data that can be extracted, what follows is a non-exhaustive list

- Title of the document
- Type of document
- Who created the document: what government agency or company
- Dates
- Sender, receiver, and subject of emails

3.2 Approach

This thesis has two main goals: To extract meta data from the documents and to extract named entities and create a co-occurrence network. Before this can be done however, two things need to happen. First the text of the documents need to be put in machine readable format, second, the documents need to be split into separate files.

As stated above, a lot of the PDF's are not machine readable. Usually this is the case because these are scans of physical documents or screenshots of digital documents like emails. These need to be converted to normal text. This will be done by using Optical Character Recognition (OCR). An state-of-the-art OCR engine named Tesseract will be used for the process of text extraction, specifically the Python wrapper of Tesseract: pytesseract. Tesseract doesn't allow normal PDF files to be processed so the files first need to be converted to images. The python library pdf2image can do this automatically. Some of the documents are machine readable so for these cases the text can be extracted using the python library pypdf2. Alternatively, Tesseract can be used for all documents even if normal text extraction is a possibility if it proves accurate enough to be justified.

Once the text is extracted from the PDF files, it needs to be split into separate documents. As stated in the data section, most of the files are compilations of several other documents. For example, some appendix files only contain hundreds of emails. If any meaningful knowledge extraction needs to take place, these larger files need to be split into separate documents. Once the documents are split, the data is ready to be used. The OCR and the document

splitting will be done by other parties within the project, but they are included here anyway for the sake of clarity.

3.2.1 Data analysis. Now that the data is machine readable, the process of NLP can begin. This will be done using spaCy. SpaCy is a very powerful NLP engine for python. It allows the user to do a number of important steps in the NLP pipeline automatically. When the extracted text is fed into a trained NLP processing pipeline it will first split the text into tokens after which it will assign part-of-speech tags to said tokens. Then a parser assigns dependency labels to the tokens and last the named entity recognizer detects and labels named entities. There is also a lemmatizer and a text categorizer but these are not necessary for this research. Custom components can also be added like pattern matcher. With this feature some standard format meta data can be extracted together with the NER. Dates and times always have a easily recognizable pattern to them. If the document is an email the sender, receiver and subject are usually preceded by a keyword that indicates what follows. For these, a pattern match will be made to extract them from the text. If the pattern matching and NER prove to be not accurate or precise enough, spaCy also allows users to add custom training data to their NER pipe. A number of examples from the text can be manually labeled and fed into the training data to increase its accuracy. Another type of entity that will have to be extracted with these patterns are articles of law. spaCy NER does come with a entity category specifically for articles of law but preliminary testing of the NER model showed that this doesn't work in the with the dutch language pack. Fortunately dutch articles of law have a very clear structure to them so a pattern can be created to match them. What follows is a list of named entities that will be extracted.

- Numbers
- Dates
- Event
- Locations
- Languages
- Articles of law
- Money
- Organizations
- Percentages
- Persons
- Products
- Time
- Work of art

The results of this process consist of a list of tuples per split document. The first item in the tuple will be the entity or found pattern match and the second entity will be the label given to the entity or patter. The label for the pattern matcher will be a custom pattern that describes what the pattern is. Before this data can be used to create a co-occurrence network, its accuracy and precision need to be evaluated. See the evaluation section on how this will be done.

3.2.2 Co-occurrence network. The co-occurrence network will be made with the results of the NER. The network consists of a number of nodes and edges. The nodes, in this case, are the found named entities. If these entities occur in the same document they will also share an edge. The edges will also contain some data. First, a number that indicates the amount of times the two named entities it connects have co-occurred in a document and second a list of documents in which the two entities occur together. Once the network is visualized that first number will dictate the thickness of the edge. Creating and analysing this network will be done with the python library NetworkX.

The visualization will be made as a python web app using the Flask library. This allows for a quick solution to make a web application. On the web side of the application, JavaScript will be used to visualize the network. The JavaScript library eCharts can be used for the purpose of visualizing a network. All of the network analysis and the data processing will still be done with NetworkX and python respectively. The reason for doing it this way instead of something like a distributable, command line program or python notebook is that making it a website makes it a lot more accessible to anyone who might want to make use of it.

3.3 Evaluation

The most important part of this project to evaluate the performance of is the Named Entity Recognition. Previous research into the performance of NER models can give a baseline to compare the model used in this thesis to. Lample et al. (2016) compiled performance scores of a lot of different studies and showed that, although the English NER models score very high with an F_1 scores around the 0.90 mark, the Dutch models do a bit worse [8]. The average F_1 score for the eight compiled models is 0.776 with one of their own models scoring the lowest at 0.699 and the heighest model being at 0.828 [4]. An earlier study showed two English NER models with scores of 0.715 and 0.824 [14]. This gives a baseline of an F_1 score between 75 and 80. The spaCy NER model can be compared to the baseline set above. However, as there is no labeled data, this has to be done manually by choosing a number of documents at random and evaluating them by hand. With this, the F_1 score can be calculated.

4 RISK ASSESSMENT

There are three major risks in this project. They are: a too inaccurate NER model, too little data for the co-occurrence network, and not enough time to finish the project. Starting with the NER model, it could turn out that an accurate enough model cannot be created. This could have a couple of reasons, one of which being that dutch NER models are inherently less accurate than their English equivalents as stated in the evaluation section. An other reason can be that the data the model was trained with is too different from the data it will have to process. For both of these problems there is an easy solution: manually add more labeled data to update the model to be more inline with the data it needs to process.

In regards to the co-occurrence network, the biggest risk lies in that the data doesn't have enough co-occurrences to get any meaningful information from the network. This would be the case if the network has too many separate components that are not connected to anything else. This can happen if most named entities only occur in one document or if most document only contain less than two entities. When this is the case it would make little sense to still make the network. The plan B then would be to use the data that was actually collected and make more of a search engine out of it where people can search all of the documents based on the meta data. Searching on named entities would also be quicker than searching through the entire text as they have already been found.

The last major risk has to do with time constraints. If it happens that there is not enough time to complete everything that is laid out in this thesis design, things need to be cut. The least important part of this research is the web app. If there is not enough time to finish the web app, it will be scrapped for plan B. Plan B in this case is to make a more simple version of parts of the functionality of the web app. This can be done as either a command line application or as a python notebook.

5 PROJECT PLAN

Week	Achievements
Week 1	In-depth analysis of the data, literature study
Week 2	Start testing accuracy of NER model. Start writing introduction and related work
Week 3	Finished introduction and related work. Start writing methodology
Week 4	Start on meta data extraction patterns and tuning NER model
Week 5	Finish methodology. Keep working on NER model and meta data patterns
Week 6	Finished NER model and meta data extraction patterns
Week 7	Finished code for co-occurrence network
Week 8	Make web application
Week 9	Finished with all programming tasks. Start writing results.
Week 10	Finished writing results. Start writing conclusion and discussion
Week 11	Finished writing the conclusion and discussion. Start with last revision of entire thesis
Week 12	Finished thesis and thesis defense

REFERENCES

[1] Edmonds, P. Choosing the word most typical in context using a lexical co-occurrence network. *arXiv preprint cs/9811009* (1998).

[2] Enthoven, G., Wiemers, S., Uijl, S. d., Nouwen, A., Kuilman, E., Jorissen, R., and Vos-Goedhart, T. *Ondraaglijk traag Analyse afhandeling Wob-verzoeken* (Jan 2022).

[3] Feicheng, M., and Yating, L. Utilising social network analysis to study the characteristics and functions of the co-occurrence network of online tags. *Online information review* (2014).

[4] Gillick, D., Brunk, C., Vinyals, O., and Subramanya, A. Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103* (2015).

[5] Greenberg, J. Metadata extraction and harvesting: A comparison of two automatic metadata generation applications. *Journal of Internet Cataloging 6*, 4 (2004), 59–82.

[6] Greenberg, J. Understanding metadata and metadata schemes. *Cataloging & classification quarterly 40*, 3-4 (2005), 17–36.

[7] Han, H., Giles, C. L., Manavoglu, E., Zha, H., Zhang, Z., and Fox, E. A. Automatic document metadata extraction using support vector machines. In *2003 Joint Conference on Digital Libraries, 2003. Proceedings.* (2003), IEEE, pp. 37–48.

[8] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360* (2016).

[9] Mendel, T. Freedom of information as an internationally protected human right. *Comparative Media Law Journal 1*, 1 (2003), 39–70.

[10] Mikheev, A., Moens, M., and Grover, C. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics* (1999), pp. 1–8.

[11] Özgür, A., Cetin, B., and Bingol, H. Co-occurrence network of reuters news. *International Journal of Modern Physics C 19*, 05 (2008), 689–702.

[12] Riley, J. Understanding metadata. *Washington DC, United States: National Information Standards Organization (http://www. niso. org/publications/press/UnderstandingMetadata. pdf) 23* (2017).

[13] Safder, I., Hassan, S.-U., Visvizi, A., Noraset, T., Nawaz, R., and Tuarob, S. Deep learning-based extraction of algorithmic metadata in full-text scholarly documents. *Information processing & management 57*, 6 (2020), 102269.

[14] Savova, G. K., Masanz, J. J., Ogren, P. V., Zheng, J., Sohn, S., Kipper-Schuler, K. C., and Chute, C. G. Mayo clinical text analysis and knowledge extraction system (ctakes): architecture, component evaluation and applications. *Journal of the American Medical Informatics Association 17*, 5 (2010), 507–513.

[15] Schmitt, X., Kubler, S., Robert, J., Papadakis, M., and LeTraon, Y. A replicable comparison study of ner software: Stanfordnlp, nltk, opennlp, spacy, gate. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (2019), IEEE, pp. 338–343.

[16] Worthy, B. More open but not more trusted? the effect of the freedom of information act 2000 on the united kingdom central government. *Governance 23*, 4 (2010), 561–582.