Dimitri Aschbacher, B.Sc

Quick Guide/Readme
Data exploring Project

Supervisor
Dipl.-Ing Haris Felic

Graz, 03.2025

1 Definition and Scope of the Project

The aim of the project was to explore the data given in numerous journal articles in an automated fashion. Normally if you want to find information about a certain topic you look for journals about the topic and read them yourself. Because this can often be tedious an automated way was looked after.

The sought method was to explore the data via the counting of words. With the frequency of specific words an assumption of the correlation can be made. If for example the word embankment often appears together with the word permeability and assumption can be made that these two are somehow related.

To use this method an understanding of the technical background is required.

2 Setup

A python enviroment is required to use all of this. Furthermore, different libraries are needed. To make the installation of all these libraries easier a textfile called enviroment.txt was created.

To use this file to install the libraries:

1. Open CMD
2. Navigate to the folder using cd
3. Type pip install -r enviroment.txt
4. done

3 How to use these programs

In the Git-hub folder there are two main branches. The first branch are the downloaders and the second branch is the text handling/plot generation.

3.1 Downloaders

The downloaders are mostly explained in the programs themselves. For each of the different publisher you need a specific API key.

The API key for Elsevier you can get here: Elsevier API Key

The API key for Wiley you can get here: Wiley API Key

Please note that for most of the downloads either a Subscription or an access trough a University is required.

After you have got the API Key for the different publishers you can open the Python file for each publisher. At the top of the program there is a section marked in between two lines of ####.

For Elsevier:

- Add your API Key
- Add the Online-ISSN of the wanted Journals in the list

Run the program and wait. This will download the most recent articles (up to 5000) for each journal. If a Journal has more than 5000 articles and you are sure you need all of them you have to run the program again after specifying the year of the last downloaded article in line 31.

Please Note that depending on the number of articles this can take up to several hours till everything is downloaded!

For Wiley:

- Add your API Key
- Add the Online-ISSN of the wanted Journals in the list
- Add the name of the Journal to the list and make sure that the issn and name of the journal are on the same position in their respective list
- Add the preferred to and from date

Run the program and wait. This will download all articles in the given journals as a pdf file.

Please Note that due to limitations in the Wiley API after 60 downloads there is a 10-minute pause. This means depending on the number of articles the download can take several hours up to days!

3.2 Texthandling

After the downloads are finished you can explore the texts. For this the textanalysis program is used. If you run this program for the first time you have to uncomment the nltk downloads at the top of the program.

In the texthandling folder you will find four different text documents. These are called application, methods, unwantedwords, ignoredwords. They are all formatted in the same way words or sentences in a single line **without** a whitespace at the end.

In the unwantedwords you can specify words or strings of words that should get filtered out in the process. These words are removed form the json files that are the baseline for the analysis.

In the ignored words you can specify words that should be ignored for the analysis but are maybe needed for a later analysis and should remain in the dataset.

In the two files applications and methods you can specify search terms that should be evaluated.

After you ruin the program a UI appears where you can select the type of analysis you want to do.

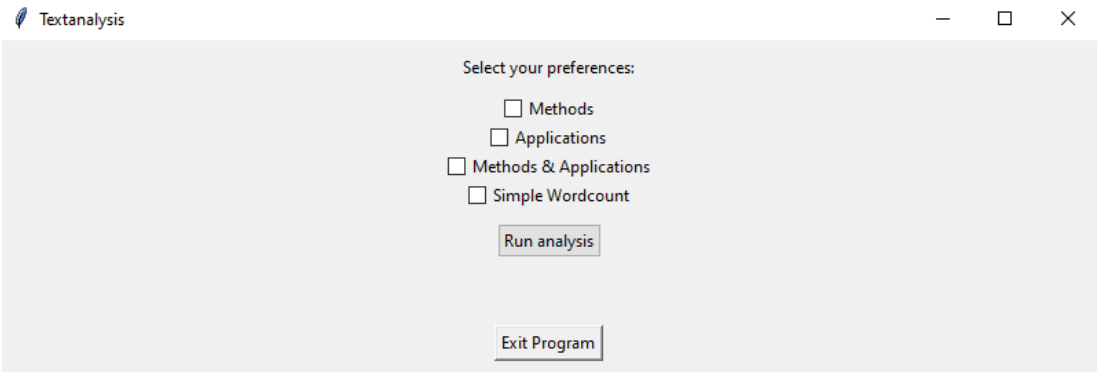

Figure 1: User Interface

Methods:

The program creates a histogram for each word in the methods file. In this Histogram the words that often appear in conjunction with the respective word are displayed.

Application:

The program creates a histogram for each word in the applications file. In this Histogram the words that often appear in conjunction with the respective word are displayed.

Methods & Applications

The program creates a histogram for each combination of a word in the methods file with a word in the applications file. In the histogram there are the often appearing words, that are in the same article as the word combination, displayed.

Simple Wordcount

The program creates a histogram for each journal where it ranks all the words that appear in each journal after the number of appearances.

All of these histograms get saved in the Plots folder.

It is recommended to first run the simple wordcount and then add all remaining filler words that were not caught by the filter to the ignoredwords list.

Please note that the UI may crash, this does not mean the program isn't running please always check if there is an error message in the terminal!

Depending on the number of articles, words in the lists and so on these runs can take some time.

3.3 Problems in the Programs

The main problem is that everything is extremely slow. The word matching for two words can take a very long time. The search for words is very basic there may be faster algorithms available.

Also, the UI runs on the same thread as the main program. This shouldn't be the case but the effort to run it in a different thread is to high, so the UI basically crashes once the program is started.

4 Findings in a geotechnical context

4.1 Example for a simple wordcount

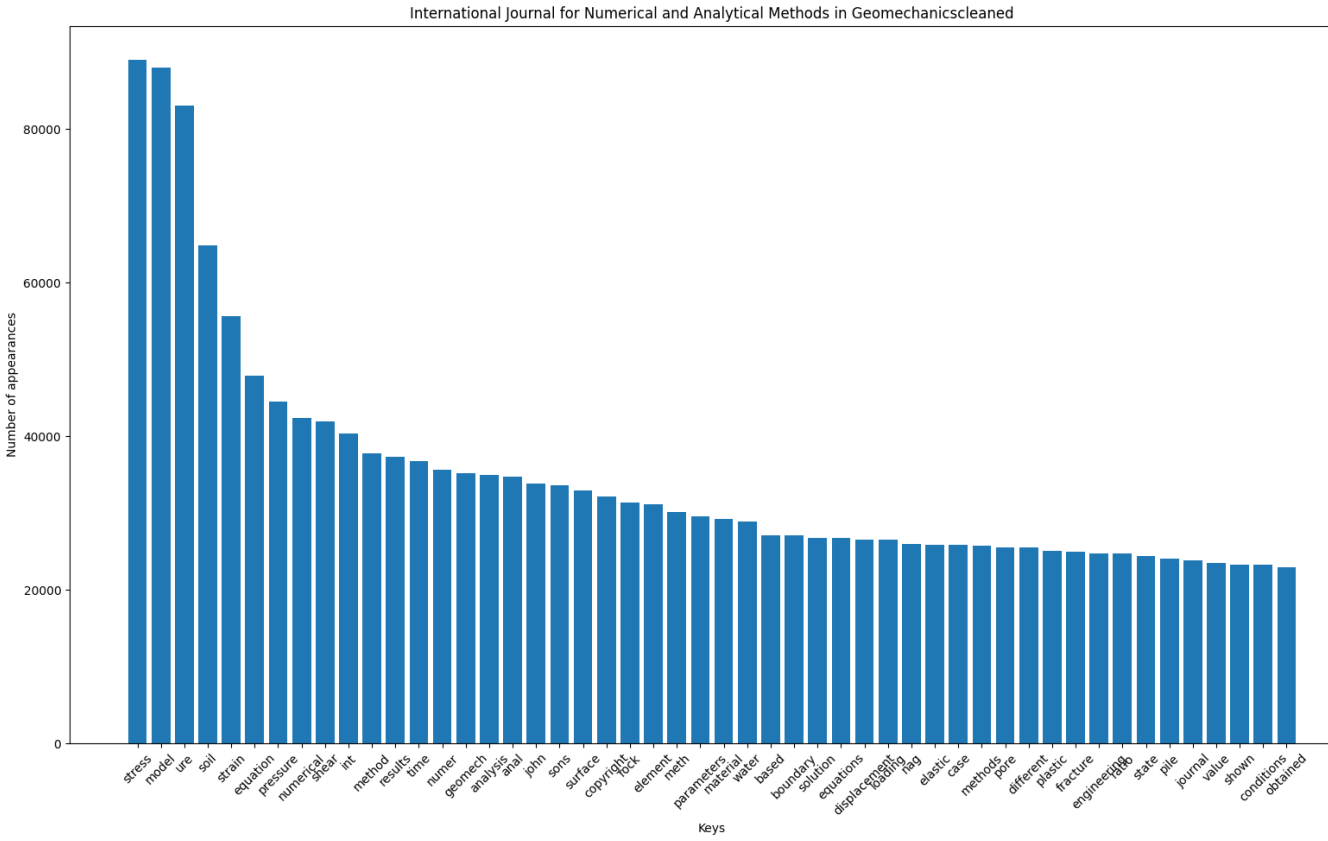

Figure 2: Simple word count for a journal

For this example, you can see that the journal of Numerical and Analytical Methods in Geomechanics deals with models a lot and not only with soils but also with rock. If this is the result of the first run you can filter out words like john, sons, shown and others that have no clear meaning. Through this iterative progress you will arrive at a histogram that is only linking geotechnical terms.

4.2 Example for a specific word

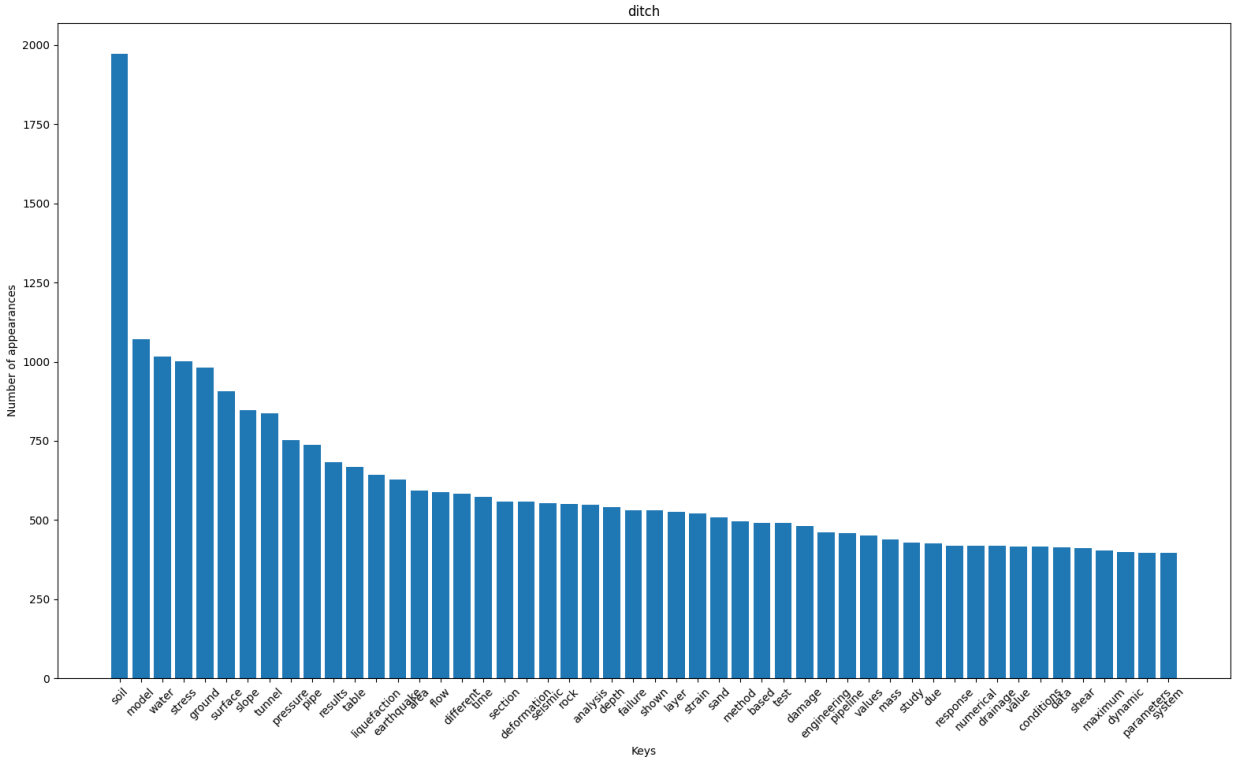

Figure 3: Example for Ditch

In this example it can be seen that for the word ditch often appearing words are liquefaction and earthquakes but also pipelines and damage. By iterating the histogram generation, you also could filter out some words and arrive at a meaningful histogram.

5 Further Work

There are many more possibilities to do with the data that is available. You could either train a ML model or simply read all the articles. A “by-product” of all this is a huge database in the form of .json files that contain all the articles so there is a large amount of text available.

Also, the correlation between different words could be calculated by matching the often-appearing words for different search words.

Furthermore, different search options would be possible, like how often does a word appear in a single article. With use of the publisher’s API also links to the metadata would be possible like which author uses which words the most, or in which year a topic was very relevant. For this maybe a simpler approach is possible.

5.1 Recommendation

The best thing would be to start with a fresh mind from the .json files and develop a new concept how to analyse the data. I’m already locked in in building these histograms and maybe a fresh view would change things up a little bit.