

ML2022: Web scraping theme group report

Michael Reusens¹, Bilal Kurban², Klaudia Peszat³, Bartosz Grancow³, Emilia Murawska³

¹ Statistics Flanders; ² Turkish Statistical Institute; ³ Statistics Poland

Objectives

The official statistics community is looking at new data sources and techniques to complement traditional methods for statistics production. The web is one of those data sources. It contains an immense amount of information on almost any policy domain, making it a data source with a large potential for statistics production.

However, transforming web data to trustworthy statistics is not straightforward. Statistical agencies need to tackle technical and methodological challenges. These challenges include software development for web scraping, natural language processing to correctly interpret web data, legal and ethical aspects and statistical challenges related to the quality of resulting statistics.

In order to boost the adoption of web scraped data for official statistics, ML2022 dedicated a theme group on this topic. By collaborating and sharing knowledge on web scraped data for official statistics, statistical agencies should be encouraged and empowered to start using web scraped data for official statistics.

Activities

The theme group focused on two types of activities: knowledge sharing and parallel implementation of experimental statistics using web scraped data.

First, during the monthly meetings presentations were given on various use cases and techniques for web scraped data for official statistics. The following topics were discussed: Existing resources for web scraping in official statistics, such as prior ESSnet projects; Use cases of web scraping in official statistics, presented by speakers from different statistical organisations; Advanced natural language processing techniques, such as Top2Vec and Lbl2Vec; Quality issues related to web scraped methods; and various other topics.

Second, the group consisted of members of three statistical institutes that during the project year implemented experimental statistics using web scraped data in parallel: Statistics Poland, the Turkish Statistical Institute and Statistics Flanders. During each monthly meeting, the implementing organisations updated the members of the theme group on their progress, followed by a discussion on the challenges faced and successes achieved.

The group members of Statistics Poland explored web data sources and new machine learning techniques to recognize the possibilities to augment official statistics on enterprises. The experimental research was focused on Corporate Social Responsibility activities.

The group members of Turkstat worked on three different applications of web scraped data: measuring social media presence, extracting insights from responses to an open-ended question in the biotechnology survey, identifying public organizations that do research and development.

The members of Statistics Flanders worked on a method to automatically categorise companies based on the text scraped from their websites. They made a proof of concept of this method by applying it to categorising companies to being active in artificial intelligence, and in bioeconomy.

A complete overview of the work done by each of the group members can be found in the individual reports, included at the end of this report.

Conclusions and lessons learned

Next to the individual web scraping achievements made by the implementing organisations, the theme group takes away the following lessons learned.

The organised knowledge sharing sessions have helped participants to hit the ground running. By identifying and discussing existing work, such as prior ESSnet work, the implementors were quickly able to start developing web scraping solutions. Furthermore, by sharing relevant work, members of the group were convinced of the added value and feasibility of web scraping data for official statistics, and were inspired to try them out in their organisations.

By implementing similar web scraping projects, the implementors often faced the same challenges. During the monthly update sessions, solutions to these challenges were discussed resulting in the challenges being overcome much more easily than if implementing organisations were working alone.

Finally, it was a valuable learning experience for the members of the group to see how these types of experiments are organised in the different implementing organisations. This very practical 'peek behind the curtains' was a great experience for members to see what their organisation could learn from the approaches presented in the group.

Appendix. Individual reports

Individual report: Turkish Statistical Institute

Machine Learning Group 2022
Theme Group: Web-scraping Data
Bilal Kurban – Turkish Statistical Institute

My Projects and Experience

Introduction

The ONS-UNECE Machine Learning Group is a platform for knowledge exchange, research collaboration and capacity building in the use of machine learning in official statistics. It brings together members from national and international statistical organisations around the world to explore the value-add of machine learning for official statistics as well as how best to integrate it into existing production systems.

I currently work in Turkish Statistical Institute (TurkStat) and I have a strong interest in how machine learning can help improve statistical output and be integrated into production systems. For this reason, I joined Machine Learning Group to contribute to the group's activities. ML Group is a dynamic community and a great place to connect with and learn from other colleagues working on similar ML challenges.

I signed up for the web scraping data theme in this year's activity programme of ML Group 2022. I chose this theme because I have an interest in web scraping and I have projects in mind that can be accomplished through web scraping. In our web scraping theme group, there were two other statistical offices as implementers: Statistics Poland and Statistics Flanders. Web Scraping Data theme group coordinator was Dr Michael Reusens from Statistics Flanders. The implementing members of the web scraping data group meets up monthly to discuss web scraping of statistical unit information. Besides informative presentations associated with web scraping, meeting agenda includes presentation of each implementing member on progress, methods, tooling, roadblocks, etc. and discussion on differences and similarities in approach.

My Projects

I ran three projects in the web scraping theme group as an implementer:

1. **Scrape an ICT variable:** Obtaining an ICT variable, namely social media presence by web scraping and comparing it to the TurkStat ICT Usage in Enterprises survey results.

2. **Gain insights from an open-ended question:** Extracting insights from responses to an open-ended question in biotechnology survey without examining individual responses one by one.
3. **Create a framework for government R&D survey:** Through web scraping, identifying public organizations that use R&D-related words or phrases on their web pages at least once and creating a government R&D survey framework with them.

Project 1 – Scrape an ICT variable ([Link](#))

In the ICT Usage in Enterprises Survey of TurkStat, the variable "whether the enterprise website has links to their social media profiles" has been obtained via web scraping this time and compared both results.

Review existing work

What I did during the preparation was that I first reviewed ESSNET Bigdata II WPC deliverables ([link](#)). In the link it can be found Workpackage C (WPC) of ESSnet Big Data deliverables which focus on enterprise characteristics. Its aim is to use web-scraping, text mining and inference techniques for collecting and processing enterprise information, in order to improve or update information held by the national business registers. The implementation involves massive scraping of company websites, collecting, processing, analysing unstructured data and dissemination of national-level experimental statistics. The enterprise data collected by WPC combined with existing data from multiple other sources, such as ICT usage surveys.

And then I visited Jacek Maślankowski (Ph.D., Assistant Professor, University of Gdańsk, Poland) WP2-Social-Media-Presence GitHub repository and examined the code he wrote and adapted the relevant code for TurkStat case. The application is used to scrap all the links related to social media from websites ([link](#)).

Work steps followed

After the preparation, I followed the below work steps:

- First, I got a list of enterprise web addresses to scrape: In the 2021 TurkStat ICT usage survey, there was a question asking enterprises to specify what their web addresses are. The responses to this question created the web addresses framework (URL list). I did not have a problem getting the list of enterprises' web addresses because we already have this information since we asked this question to enterprises in the 2021 ICT usage survey. As you can see from the below screenshot we asked enterprises if they have a website or not and for those who have a website we wanted them to indicate it.

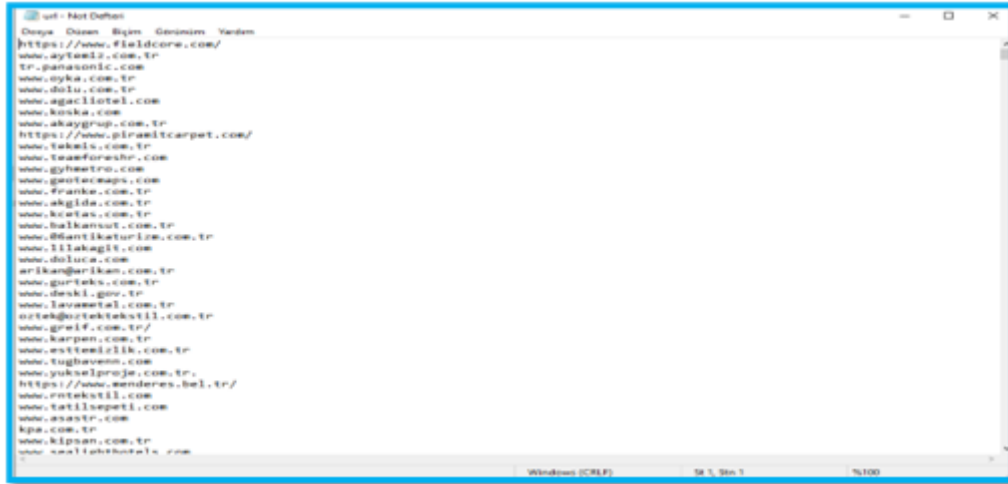


The image shows a survey interface with two questions. The first question, labeled '1.5', asks 'Does your enterprise have a website?' and provides two radio button options: '1.Yes' and '2.No'. The second question, labeled '1.5.a', asks 'Please specify' and features a text input field with the placeholder text 'www.'.

Figure 1. TurkStat ICT usage survey question asking enterprises for their websites

- Next, I created a text file of these web addresses (as shown below) that would be the input for the code I was going to run. A text file was created from the list of web addresses of

enterprises with 250+ employees. And this file became the input file of the code. ICT Usage Survey in TurkStat is a complete census for enterprises with 250+ employees. Therefore, I limited this study to only enterprises having 250 or more employees in Türkiye. The python code was designed to take a text file with web addresses written one after the other as input. Therefore, I also created my file in this format. It is also possible to just copy/paste the ready text file.



```
Notepad - Not Defter
Çözge Özgen Şişir Gönümlü Yavuz
https://www.falidcore.com/
www.aytasel.com.tr
tr.gamaonline.com
www.oyku.com.tr
www.dulu.com.tr
www.ayacilotel.com
www.kaska.com
www.akaygrup.com.tr
https://www.prasitcarpet.com/
www.tahsil.com.tr
www.taamfarshe.com
www.gybasir.com
www.garfacmaps.com
www.franka.com.tr
www.ahgida.com.tr
www.kocbas.com.tr
www.bakarsat.com.tr
www.distantkatorlan.com.tr
www.illakoglu.com
www.dulu.com
arlikanharlikan.com.tr
www.garbas.com.tr
www.deshil.gov.tr
www.lacemetal.com.tr
oztekg@oztekgokull.com.tr
www.grelf.com.tr/
www.karpar.com.tr
www.ortemizlik.com.tr
www.tugbavem.com
www.yukselproje.com.tr
https://www.marderes.bel.tr/
www.ortekull.com
www.tatlisport.com
www.ozantr.com
kpa.com.tr
www.kipart.com.tr
www.ozt1atvortak1a.com
```

Figure 2. “url.txt” input file consisting of URLs

- Then, I used Jupyter notebook in the Anaconda distribution to execute the code which is an open-source IDE (Integrated Development Environment). Very basically the purpose of the code is to find social media links on enterprise websites. Input is a list of URLs and output is a “csv” file containing domain names and found social media links. I executed the ready-made python code ([link](#)) with very minor changes using Jupyter Notebook. The application scrapped all the links related to social media from websites.

```

In [5]: import requests
import re
import string
import csv
import sys
import json
import os
import os.path
from bs4 import BeautifulSoup
from collections import OrderedDict
from datetime import datetime

# Global variable to store the list for JSON purposes
jsonlist=[]

# HTMLParserSS class is used to find all URLs that reference to Social Media
class HTMLParserSS:
    output_list=[]
    # method extractURLs is to extract all URLs and return them as the urls[] list
    # it goes through the website to find all anchors as hrefs
    def extractURLs(self,page):
        # all links are lowered because sometimes the same link is written differently, e.g., ug.edu.pl or UG.edu.pl
        soup = BeautifulSoup(page.text.lower(),"html.parser")
        urls = []
        for a in soup.find_all('a', href=True):
            url = a.get('href')
            urls.append(url)
        return urls
    # method extractAllURLs is to extract all URLs that are not in anchors as
    # it uses regular expression that search for any http or https, even not included in anchors
    def extractAllHTTP(self,page):
        URLS=re.findall("https?://[\w\.-]+?[\w]{1,256}.*", page.text.lower())
        return URLS

# class SocialMediaDeep contains a function responsible to divide the links into:
# (1) internal (used for the second search and external)
# (2) external (not used for the second search of social media)
class SocialMediaDeep:
    website=""
    # this method is responsible to do the second search on subpages
    # from internal links that are present on the main page of the website
    def goDeeperToFindSocialMedia(self,website,URLS):
        print("Preparing to scrape subpages...")
        for url in URLS:
            try:
                if url:
                    # the difference between InternalURL_type2 and InternalURL_type1

```

Figure 3. Running code for web scraping in Jupyter Notebook

- And finally, I compared the results generated by the code with the results from the survey for the social media presence variable. As the identifier, the web address field in the output file could be used for matching and linking scraped data and survey data. I had to make some adjustments in the web addresses to bring them to the same format to achieve the maximum match rate. After combining the two datasets, all that remains was to make a comparison which was made with the scraped data and the enterprise responds to the question shown below figure in the 2021 ICT survey questionnaire. The results obtained with web-scraping were compared with those obtained from the 2021 survey.

A7. Does the website have any of the following? <i>-Optional</i>		Yes	No
*6	a) Description of goods or services, price information	<input type="checkbox"/>	<input type="checkbox"/>
	b) Online ordering or reservation or booking, e.g. shopping cart	<input type="checkbox"/>	<input type="checkbox"/>
	c) Possibility for visitors to customise or design online goods or services	<input type="checkbox"/>	<input type="checkbox"/>
	d) Tracking or status of orders placed	<input type="checkbox"/>	<input type="checkbox"/>
	e) Personalised content on the website for regular/recurrent visitors	<input type="checkbox"/>	<input type="checkbox"/>
	f) Links or references to the enterprise's social media profiles	<input type="checkbox"/>	<input type="checkbox"/>

Figure 4. “social media presence” question in the TurkStat ICT usage survey

The code took a total of 4 hours and 44 minutes to run and produced 1,566 pages of output. The execution of the code was interrupted only once. Then I just rerun the code for the remaining URLs and combined the two outputs when it has finished.

Results

As a result, web scraping was successfully performed for 1,087 out of a total of 3,640 enterprises. This corresponds to a rate of 30%. The comparison summary results are below:

Table 1. Summary of the comparison between web scraping and the survey results for the social media presence variable

		Web Scraping		Total
		Yes	No	
Survey	Yes	392 (36%)	325 (30%)	717
	No	89 (8%)	281 (26%)	370
Total		606	481	1,087

As can be seen from the table above, the percentage of enterprises stating in the survey that they have a link to their social media platforms and having at least one social media link in the scraped data is 36%. The percentage of enterprises stating in the survey that they don't have any link to their social media platforms and having no social media link in the scraped data is 26%. The percentage of enterprises stating in the survey that they have a link to their social media platforms but having no social media link in the scraped data is 30%. And lastly, the percentage of enterprises stating in the survey that they don't have any link to their social media platforms but having at least one social media link in the scraped data is 8%.

Eventually, 62% accuracy was achieved while the inconsistency was 38% between two different data acquisition methods.

Project 2 – Gain insights from an open-ended question ([Link](#))

Introduction

With the Turkish Statistical Institute (TurkStat) Biotechnology Statistics Survey, it is aimed to create statistical data in the field of biotechnology. All enterprises engaged in biotechnology activities have been covered in the survey and the data is collected directly from enterprises via web survey.

In the 2020 Biotechnology Statistics Survey, unlike previous years, an open-ended question was added to the end of the questionnaire in addition to the routinely asked questions. The purpose of adding this question was not to miss anything about this very technical domain. Because open-ended questions allow the respondent units to freely express the points that the survey designer missed on the subject. The question was: “Briefly inform us about the biotechnology activities carried out by your enterprise and the techniques and applications it uses”. All enterprises carrying out biotechnology activities were required to fill in this open-ended question at the end of the survey questionnaire. Eventually; all 499 enterprises carrying out biotechnology activities in Türkiye filled the free text box provided to them with information about their activities.

This paper presents an experimental study to analyse these free texts and try to understand in a few words what biotechnology enterprises in Türkiye are mentioning about most in these texts in 2020. Of course, these free texts would not be reviewed and analysed one by one, which was not possible anyway. For this reason, a different analysis method based on Natural Language Processing (NLP) and deep learning techniques has been developed using Python to gain insights from these free texts. Natural Language Processing—or NLP for short—in a wide sense to cover any kind of computer manipulation of natural language [1]. The reason for choosing Python programming language is that it has an excellent functionality for processing linguistic data.

Methods

In order to extract insights from the responses provided by enterprises to the open-ended biotechnology survey question as free text, the following processes were carried out in Python programming language:

1. Preparing data by pre-processing,
2. Finding collocations,
3. Finding most common words and
4. Detecting topics present in text with Top2vec algorithm

The second, third and fourth steps mentioned above were repeated once more after both removing stop words and lemmatization.

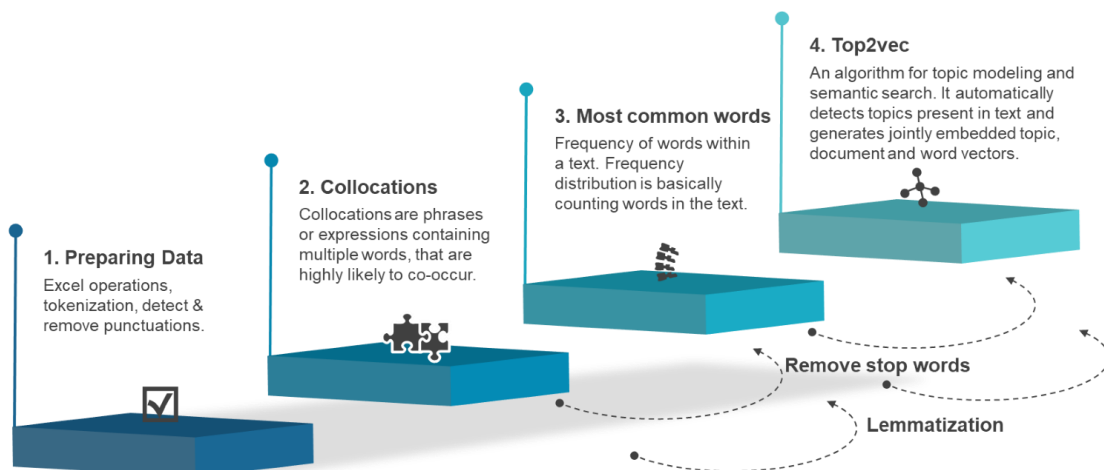


Figure 5. Process steps

Preparing data by pre-processing

Before insights could be gained from the free texts, first of all, some operations needed to be done on the raw data. These operations covered deleting nonsense records (the records with only "." or "x"), standardizing some words (such as removing the hyphen between words) and converting all letters to lowercase. After these operations performed on the raw data, the data was ready for pre-processing. Pre-processing stage included tokenization and detecting and removing punctuations.

The analyses carried out to extract meaning from the free texts in this study were also repeated by removing stop words and performing lemmatization. Stop words are basically a set of commonly used words in any language and in NLP and text mining applications, they are used to eliminate unimportant words. As for the lemmatization, it is the method to take any kind of word to that base

root form with the context. It groups together the different inflected forms of a word so they can be analysed as a single item [2].

Finding collocations

After the pre-processing was completed, analyses were started to extract meaningful information from the word list obtained. For this purpose, collocations in the word list were searched first. Collocations are expressions of multiple words which commonly co-occur [3]. Natural Language Toolkit (NLTK) library in python has been used to find collocations. NLTK is a leading platform for building Python programs to work with human language data [4]. NLTK contains *collocations* module having tools to identify collocations within corpora.

Finding most common words

One way to find out what is most frequently mentioned in free texts is to look at the most frequently used words in these texts. For this purpose, number of occurrences of each individual of the word/word group were calculated through the *FreqDist* module in NLTK.

Detecting topics present in text with Top2vec algorithm

In this study, it has also been tried to find how many different topics can be produced and what these topics can be by combining similar words in free texts using the *Top2vec* algorithm. Top2Vec is an algorithm for topic modelling and semantic search. It automatically detects topics present in text and generates jointly embedded topic, document and word vectors [5].

Results

First; the bigram and trigram collocations were obtained before removing stop words, using three different measure of association, namely Pointwise Mutual Information (PMI), Likelihood Ratio and Raw Frequency. The results are shared with the figures below.

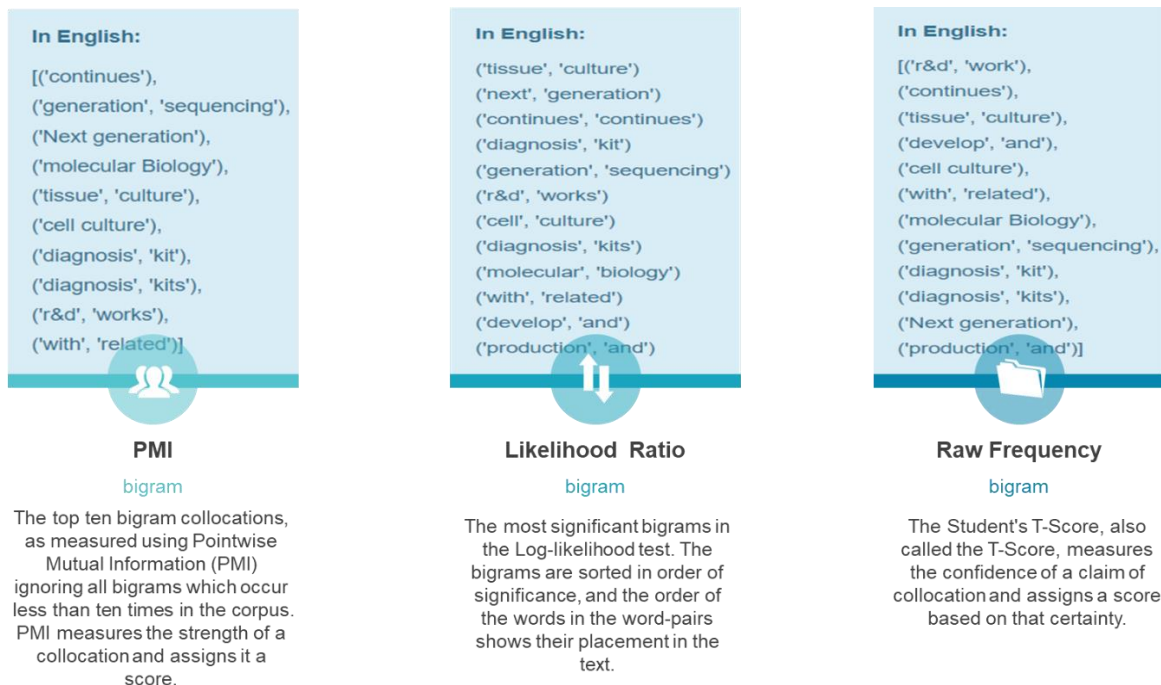


Figure 6. Bigram collocations

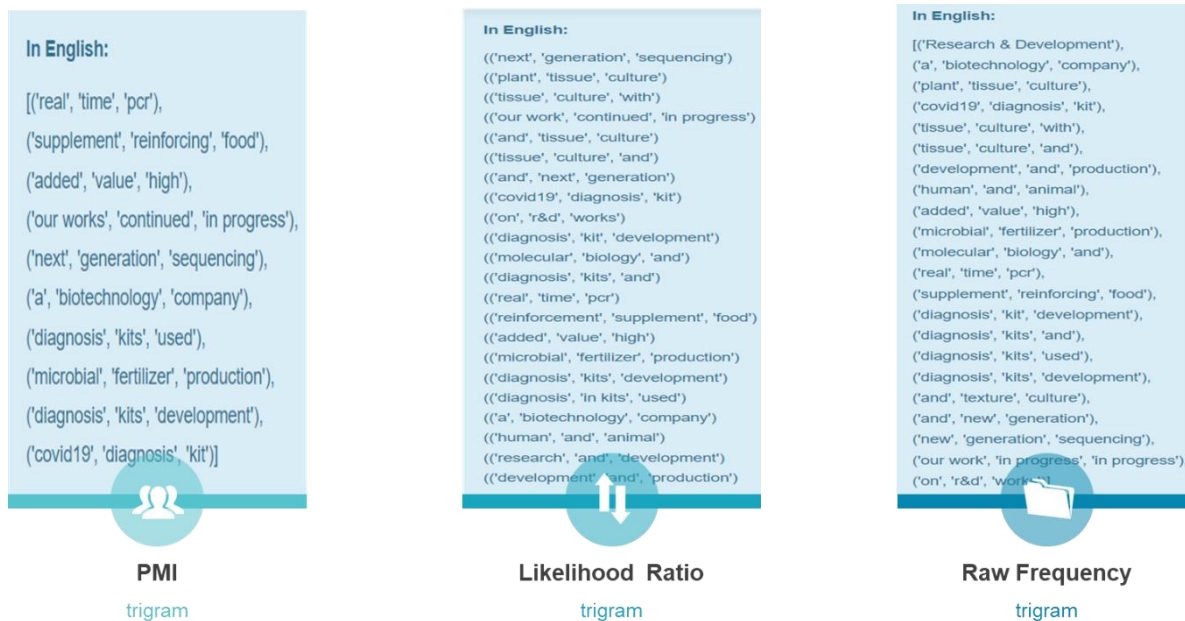


Figure 7. Trigram collocations

Then, the most used words in the texts entered into the open-ended question, were found as single, double and triple before and after removing stop words. The most important finding here was that after removing stop words, the three-word phrase "plant tissue culture" was repeated six times. The results of word counts are shared below figure. Note that, as a result of one-word count, since stop words have dominated the frequency table they are not included in the below figure.

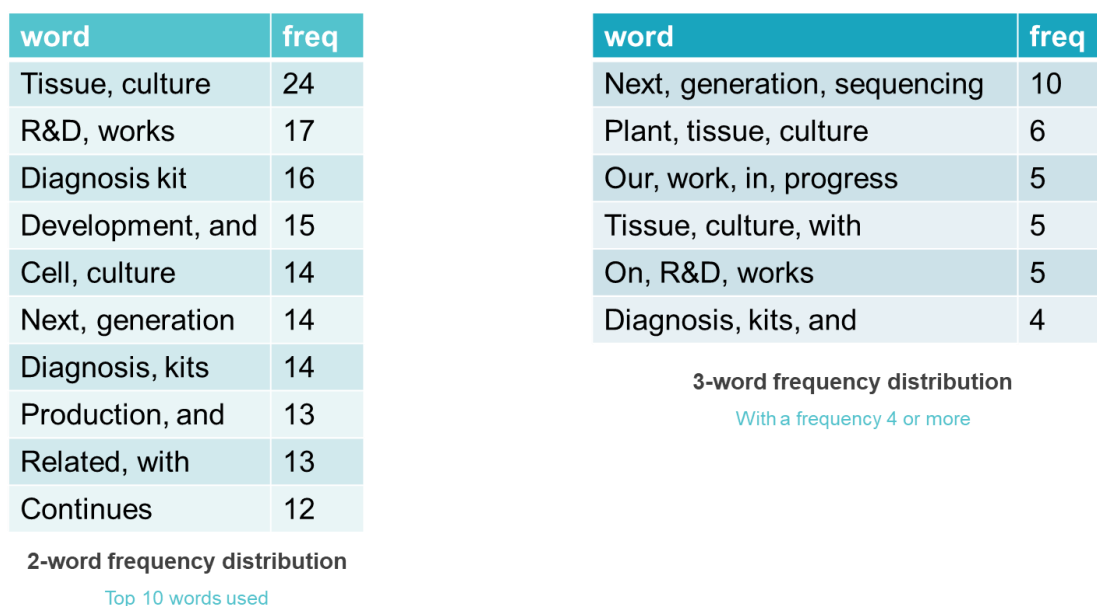


Figure 8. Word counts

Collocations and word counts were obtained again after stop word elimination:

PMI bigram	PMI trigram	Word Count 1-word	Word Count 2-word
<p>In English:</p> <p>[('generation', 'sequence'), ('molecular Biology'), ('tissue', 'culture'), ('test', 'kit'), ('cell culture'), ('agricultural', 'biotechnology'), ('diagnosis', 'kit'), ('diagnosis', 'kits'), ('r&d', 'activities'), ('plant', 'tissue')]</p>	<p>In English:</p> <p>[('real', 'time', 'pcr'), ('supplement', 'reinforcing', 'food'), ('added', 'value', 'high'), ('diagnosis', 'kits', 'used'), ('microbial', 'fertilizer', 'production'), ('diagnosis', 'kits', 'development'), ('covid19', 'diagnosis', 'kit'), ('plant', 'tissue', 'culture'), ('diagnosis', 'kit', 'development')]</p>	<p>In English:</p> <p>(production,) 95 (r&d) 80 (development,) 70 (diagnosis) 67 (tissue) 49 (biotechnology,) 49 (dna) 45 (plant,) 45 (culture,) 45 (production,) 45</p>	<p>In English:</p> <p>(tissue, culture) 24 (diagnosis kit) 16 (cell, culture) 14 (diagnosis, kits) 14 (molecular, biology) 10 (generation, sequencing) 10 (agricultural, biotechnology) 9 (test, kit) 8 (R&D, activities) 8 (genetics, diagnosis) 7 (plant, tissue) 7</p>
Only bigrams that appear 7+ times	Only bigrams that appear 3+ times	Top ten	With a frequency of 7 or more

Note: the most frequent 3-word is (plant, tissue, culture) 6.

Figure 9. Collocations and word counts after eliminating stop words

Finally, the top2vec algorithm was used to find similar words used in free texts and to derive topic titles from these word groups. Top2vec derived four topics and also visualized the similar words it grouped.



Figure 10. Top2vec results (in Turkish)

The words in the free texts were also lemmatized. But this did not work very well in this case as the package used could not find the root words well enough.

Conclusion

Based on their responses to the open-ended question “Briefly inform about the biotechnology activities carried out by your enterprise and the techniques and applications it uses” in TurkStat Biotechnology Statistics Survey, in 2020, the enterprises that carry out biotechnology activities mostly mentioned the words on the below word cloud.

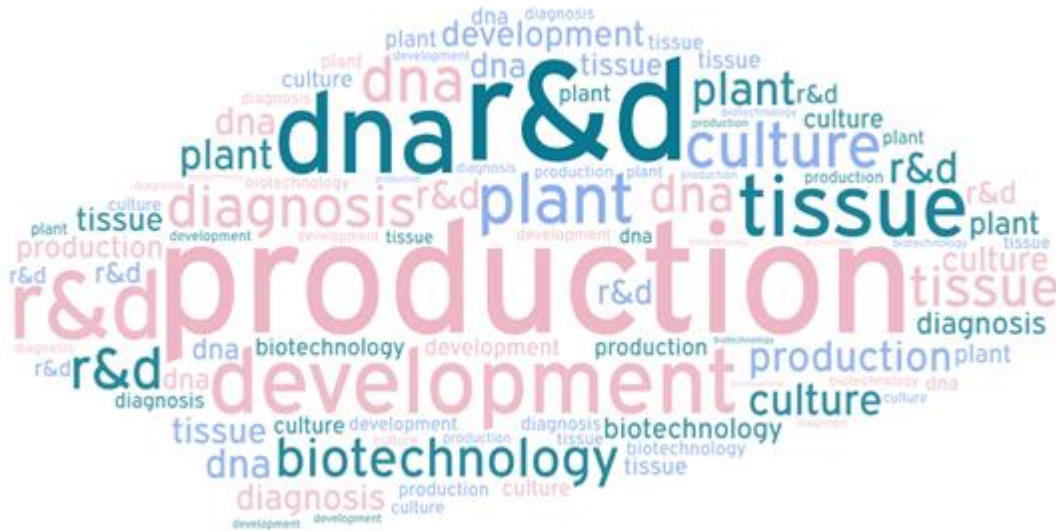


Figure 11. Most emphasized single words in open-ended texts

When we dig a little deeper, biotechnology enterprises mention these two words the most in order of emphasis): molecular biology, tissue culture, diagnosis kit, cell culture, R&D activities, test kit, agricultural biotechnology, plant tissue and genetic diagnosis.

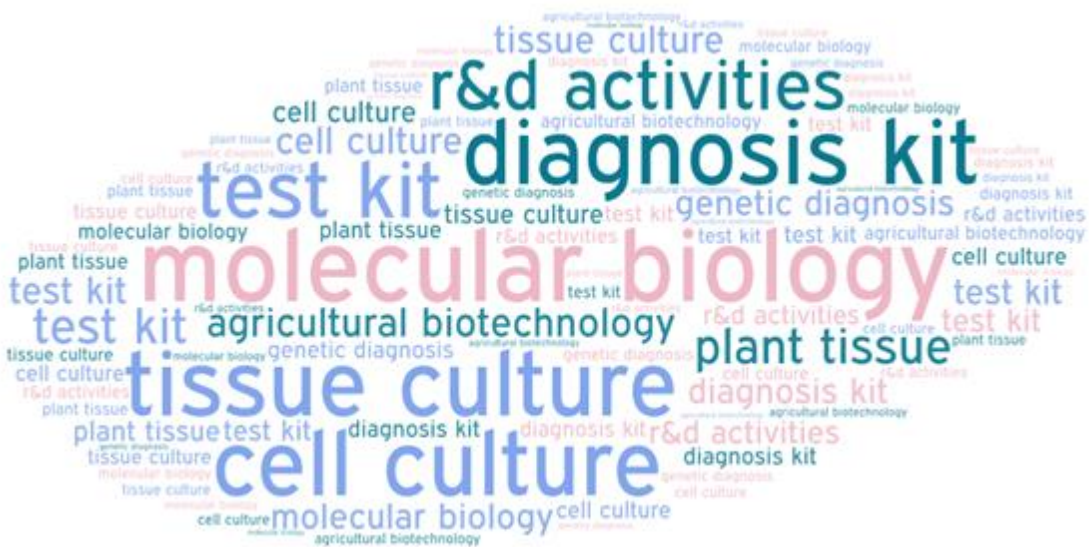


Figure 12. most emphasized two words in open-ended texts

NLP code outputs give a deeper insight into the biotechnology activities of the enterprises. As for three-word expressions, the most emphasized expressions in open-ended texts are: plant tissue culture, research and development, next generation sequencing, real time PCR, high value added, food supplement, diagnosis kit development, microbial fertilizer production, covid19 diagnosis kit, human and animal and development and production.



Figure 13. most emphasized three words in open-ended texts

Since the reference period of the Biotechnology Statistics survey is 2020, when the pandemic is more intense, the effect of covid-19 on the results can be easily seen (such as real time PCR, diagnosis kit development, covid19 diagnosis kit etc.).

As a result of this work, it was concluded that NLP and deep learning techniques can be used to reduce human effort in extracting meaning from responses to an open-ended free text. They can automate and speed up an otherwise laborious or infeasible task

REFERENCES

- [1] S. Bird, E. Klein and E. Loper, Natural Language Processing with Python, (2009), ix.
- [2] <https://jaimin-ml2001.medium.com/stemming-lemmatization-stopwords-and-n-grams-in-nlp-96f8e8b6aa6f>, accessed 23 September 2022
- [3] <https://www.nltk.org/howto/collocations.html>, accessed 23 September 2022
- [4] <https://www.nltk.org/>, accessed 23 September 2022
- [5] D. Angelov, Top2Vec: Distributed Representations of Topics, (2020), arXiv.org, accessed 23 September 2022

Project 3 – Create a framework for government R&D survey ([Link](#))

Introduction

I tried to find out if Research and Development (R&D) was mentioned on the web pages of the government units via web scraping. So that I can add them to the Government R&D survey framework. Main purpose was to create an evidence-based Government R&D Survey framework. This framework is currently created by including government units that have the potential to carry out R&D activities in the framework.

It is relatively easy to set up the survey framework for financial companies because the list of business enterprises receiving R&D support is available in the administrative registers. However, we do not know which institutions and/or organizations carry out R&D activities on the government R&D side. For this reason, I am trying to find out those who use the term “R&D” on the internet pages of public institutions and how many times this R&D expression is mentioned. So that; if the

Research and Development phrase is mentioned on their web pages, I can include them in the government R&D framework. Thus, both the response burden and the cost are reduced.

Figure 14. Project code summary

Work steps followed

For this purpose, I developed a project with web scraping in python that searches for keywords from a list of URLs. The project consists of two main parts: searching the links and scraping the keywords in the found links. In the first part, I took advantage of [the article](#) by Kelvin Kramp (accessed 18 October 2022) and by modifying it I performed the following steps in order:

- Import necessary modules/libraries
- Write a function for getting the text data from a website URL
- Write a function for getting all links from one page and store them in a list
- Write a function that loops over all the subpages
- Import the URL list and create the loop to get all “a href” marked links.

In the second part, the code executed the following steps in order:

- Import previously produced json data containing all the links found
- Get links from imported json data
- Import the keyword list and scrape the list of words from URLs
- Print the url's scraped and cumulative number of found words
- Show and export only the links containing keywords with their frequency

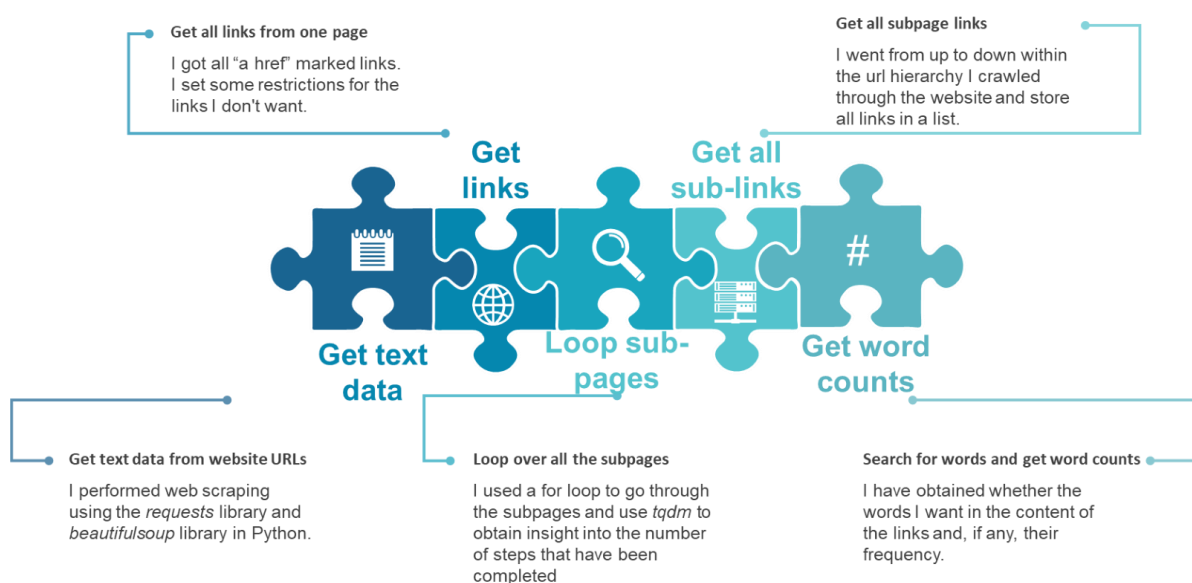


Figure 15. Process steps

The code takes a list of URLs as input and performs web scraping on the web pages and their subpages in that list for a list of words/phrases, which is also a separate input. This is actually a generic application: It reports how many of the word/word groups you entered as a list were detected within the URLs you entered as a list, at the URL level. By default, the program uses two iterations to detect subpages. This iteration number can be changed as some URLs might require more iterations than others.

Conclusion

In conclusion, with this project, it is possible to learn which government units (ministries, general directorates, hospitals, municipalities, etc.) mention R&D on their websites and how often. Thus, we have an evidence-based knowledge of whether to include them in the R&D survey framework or not.

At the same time, since this is a generic application, it can be used easily when certain words or word groups are searched on specified web pages.

Overall Conclusion

The ever-growing supply of data and demand for timely and high-quality statistics are challenging governments to transform the way they produce official statistics. Traditional methods and resource alone are not enough to exploit the vast potential of this data. Significant investment in both technology and capability will be needed to harness the opportunities from administrative data and other sources of big data. National Statistical Offices around the world have stepped up their response in recent years, with more organisations establishing their own specialist data science functions to conduct exploratory research and build capability.

As more statistical organisations develop their own data science strategy and functions, it is important for them to learn from each other's experiences and identify ways to drive forward development.

Joining the ML group 2022 and being an implementer in the web scraping theme group provided a great opportunity for me and representatives from different organisations and job functions to reflect on the progress that has been made in building data science capability in official statistics in the past years, discuss common challenges, and learn about different approaches to tackling them.

In the past twelve months it has been run a varied programme of research, knowledge exchange and capacity building activities in the use of ML for official statistics. I found this group a valuable and enjoyable platform for connecting with colleagues working on ML across the world. And the skills and knowledge I built here have direct benefits for my work and organisation.

Individual report: Statistics Poland

Statistics on companies undertaking activities in the field of corporate social responsibility (CSR) using web scraping and machine learning

Organisation: Statistics Poland

Authors: Bartosz Grancow,
Emilia Murawska,
Klaudia Peszat

Date: 15.11.2022

Version: 2.0

Introduction

The traditional data sources do not provide information on many new and fast changing socio-economic phenomena, therefore national statistical offices more often reach for new data sources, such as web data.

Statistics Poland within the ML2022 web scraping theme group made an attempt to explore web data sources and new machine learning techniques to recognize the possibilities to augment official statistics on enterprises. The experimental research was focused on Corporate Social Responsibility activities.

Corporate Social Responsibility (CSR) is defined as „a company's sense of responsibility towards the community and environment (both ecological and social) in which it operates.

Companies express this citizenship:

- (1) through their waste and pollution reduction processes,
- (2) by contributing educational and social programs, and
- (3) by earning adequate returns on the employed resources” (*the Business Dictionary*).

The report presents the results of the research which aimed at the classification of companies into two groups: those which carry out activities in a CSR field (and inform about it) and those which more likely do not have any CSR strategies or at least do not communicate their activities in this area. The type of such activity was not the subject of the project.

Data

Input data

The data source used to build the population of companies was the Business Register – Database of Statistical Units (BJS), including name of enterprise, address, e-mail address, URL address (if available). Due to the fact that CSR-related activities are most often carried out by large and medium-sized companies (small companies often do not have websites), we focused our attention on this group of enterprises (employing more than 10 employees).

The first step was the extraction from the Business Register of the population on enterprises with URL addresses. It is worth mentioning that, the URLs can be also obtained from email address. The methodology for obtaining URLs among others from email addresses has been described in detail in the report delivered within the ESSnet Trusted Smart Statistics – Web Intelligence Network project (Kuhnemann et al. 2022). Taking into account the exploratory purpose of our research and the fact that the database of large and medium companies covered over 20 thous. enterprises with URL addresses, we did not decide to obtain additional URLs via email addresses. The total number of URLs in our database consisted of 6 thous. URLs for large enterprises and nearly 14 thous. for medium ones from all over Poland.

Next, we checked validity of the URLs using web scraping tools. There were 11,380 valid URLs (with response code 200). Inactive URLs were removed from the database.

Search engine results

We have decided that in a scope of our interests are only websites held by the companies to provide information on their business. The enterprise should have control over the content of such websites. External web pages providing basic information on the companies, such as yellow pages, government domains etc. were excluded.

Information on CSR activities can be found in various places on the website. Some enterprises post them on the home page, others have a dedicated tab or subpage. The descriptions of the CSR activities conducted by companies are also published in various ways. Some enterprises publish them as a plain text on the website, others attach pdf reports.

Taking into consideration the enormous variety of websites, we decided to use an internet search engine to find texts from the official company's web sites referring to their activities in the CSR field. Only search results (snippets) were used as data source and no additional scraping was done. This approach has been implemented first by CBS (Delden et al. 2019). This enabled to save time and resources needed to web scrape the entire content of webpages.

The choice of the search engine and the search term have significant impact on the results. However, due to the exploratory purpose of the research and payment requirements for using popular search engines, such as Google or Bing for mass queries, we decided to use the less known search engine – Duck Duck Go¹. Its advantage is not collecting data about users, and thus not profiling the results. Each user can get the same search results, which made it easier to work on the project.

We also tested various search options by entering company's URL and keyword or name of a company and keyword. The first variant brought more accurate results. When entering a company's name, contact details were usually in the first places of the search.

To prepare a catalogue of keywords that appeared most often in the descriptions of CSR activities we also searched manually on companies websites. The directory consisted of words and phrases such as: 'csr', 'społeczna odpowiedzialność', 'odpowiedzialność biznesu', 'społeczna', 'zaangażowanie', 'środowisko', 'ekologiczne', 'zrównoważony rozwój', 'csr raport', 'charytatywny'. We tested all above mentioned keywords, however the search term 'URL address' + keyword: 'csr' gave the best results.

We also noticed that if a company conducts CSR activities, information about it appears in the first five search results (most often in the first three descriptions, but there are some cases when it appears in the fourth or fifth ones). For this reason, we limited the search to the first five results.

Below is an example of the search results – URL address and keyword entered manually in the search engine Duck Duck Go and the results of descriptions retrieved and saved in an Excel file.

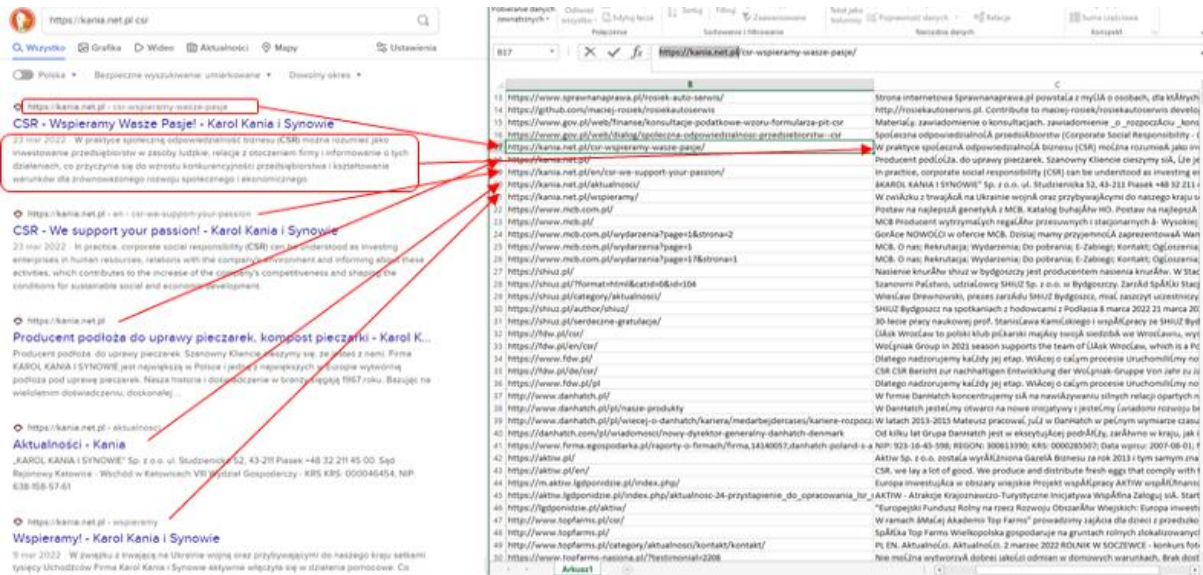


Figure 1. Search engine results

Data preparation

Data stored in an Excel file had to be further processed. At this step, results which did not come from company's webpages were removed from the database. It was: yellow pages, phonebook websites (e.g. panoramafirm.pl, oferteo.pl, fimy-pl.com), job offer pages (e.g. gowork.pl, pulshr.pl), social media (e.g. Facebook, LinkedIn), social campaign pages (e.g. kampaniespoleczne.pl), non-governmental organizations pages (addresses with the org.pl domain), as well as government and educational institution pages (addresses with the gov.pl and edu.pl domain).

In the next step, the pre-processing techniques, such as tokenization, reduction of words to their basic form by lemmatization and stop words removal were applied.

Machine learning solution

Model Top2Vec

Once the data was prepared, we attempted to detect proximity of texts from our database with the CSR topic. For this purpose, we applied a new unsupervised machine learning algorithm – the Top2Vec. The Top2Vec algorithm is used to automatically detect topics present in texts and documents. This algorithm was chosen as it enables working on short text (such as snippets), does not require removing stop words and pre-processing (e.g. stemming/lemmatization)².

Despite the fact that the Top2Vec algorithm does not require previous text processing, some of the pre-processing steps were done, mostly to compare the model results applied on the raw data and pre-processed ones.

The default embedding model applied in the Top2Vec algorithm which enables creation jointly embedded document and word vectors is Doc2vec. This is not a pre-trained model, it learns from scratch and thus it can be used for different languages. The algorithm provides also a possibility to use pre-trained embedding models, such as: universal-sentence-encoder, which currently supports 16 languages, including Polish. For the experimental research purposes different embedding models were tested, however universal-sentence-encoder-multilingual-large model proved to give the best results.

Software used

The implementation of the Top2Vec model was carried out in a Python programming language. The installation of the library seemed not to be very challenging, however several adjustments of the IT environment were necessary. First of all, Microsoft Visual C++ 14.0 or greater was required, and secondly – installation of hdbscan. The latter problem can be easily solved in Anaconda environment by running the code: `conda install -c conda-forge hdbscan`.

Results

Raw data

The Top2Vec algorithm was applied on the dataset consisting of 13.4 thous. raw texts. The model detected 73 topics, including one very closely related to the CSR concept (Topic 0). It consists of 477 semantically similar texts, which also makes it the largest cluster. The wordcloud for the topic 0 is presented below.

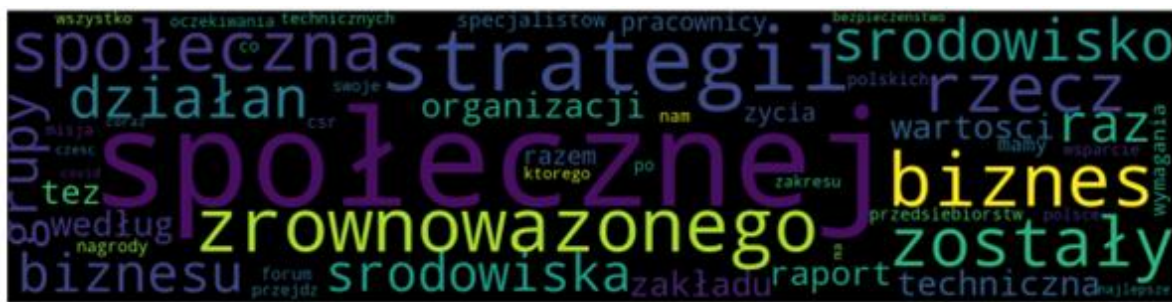


Figure 2. The wordcloud for the topic 0 (in Polish)

In the next step, we calculated the document scores, which indicate a semantic similarity of documents to the assigned topic. The table below presents the document scores for the texts clustered in the topic 0 (in descending order).

id	url	text	topics	topic_scores	document_scores	
0	1	https://kania.net.pl/csr-wspieramy-wasze-pasje/	w praktyce społeczną odpowiedzialność biznesu ...	0.0	0.519214	0.687659
1	2	https://kania.net.pl/en/csr-we-support-your-pa...	in practice corporate social responsibility cs...	0.0	0.506223	0.675753
2	3	https://kania.net.pl/de/csr-wir-unterstuetzen-...	in der praxis kann die soziale verantwortung v...	0.0	0.521032	0.673456
3	4	https://www.czatkowice.pl/zrownowazony-rozwoj/csr	csr zrównowazony rozwój kopalnia wapienia czat...	0.0	0.470100	0.673364
4	5	https://www.czatkowice.pl/zrownowazony-rozwoj	kwestię csr traktujemy bardzo kompleksowo dlat...	0.0	0.477510	0.671222
...
472	473	https://about.puma.com/en/sustainability	social compliance against modern slavery and h...	0.0	0.436546	0.247481
473	474	https://annual-report.puma.com/2020/en/sustain...	t puma for sustainability targets performance ...	0.0	0.250929	0.245022
474	475	https://www.csrhub.com/CSR_and_sustainability_...	cnh industrial n v description open who uses c...	0.0	0.395547	0.239758
475	476	https://www.dailycsr.com/tags/CNH%20Industrial/	daily csr companies environment economics poli...	0.0	0.538010	0.230428
476	477	http://joyevent.pl/party/csr-w-naszym-domu	csr corporate social responsibility jest konce...	0.0	0.624386	0.217756

477 rows × 6 columns

Figure 3. The results for the topic 0

The results can also be easily assessed in a scatter plot where the x axis is the document id and y axis is the document score.

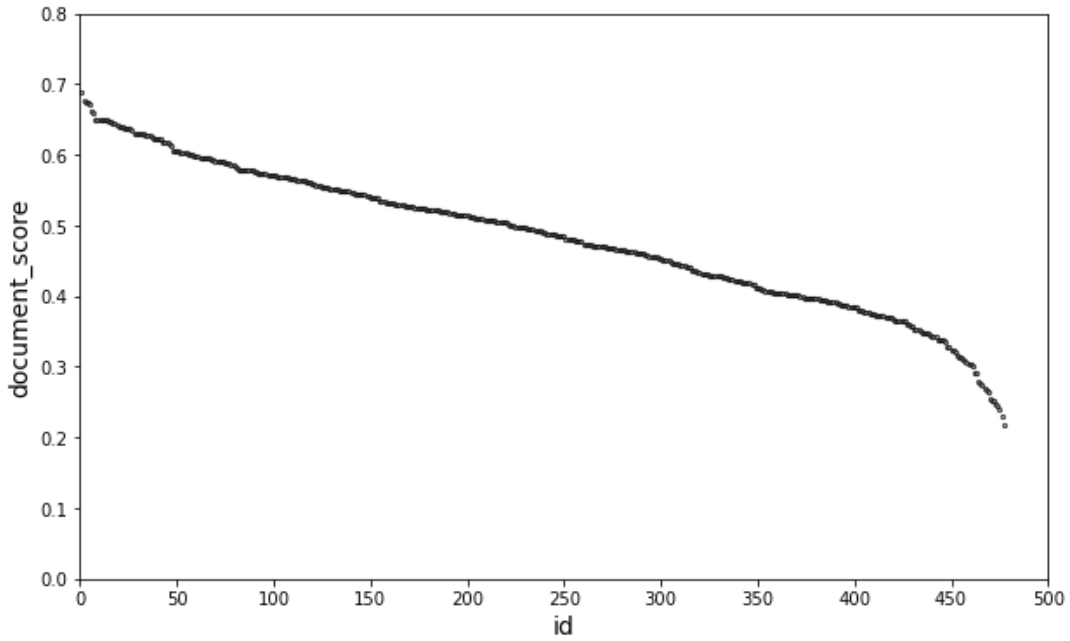


Figure 4. The scatter plot for the topic 0

The table and scatter plot contain the number of texts from companies websites, thus to calculate the number of companies which more likely conduct CSR activities we had to deduplicate information.

The number of unique companies which are assigned to the topic 0, closely related to the CSR concept is 280 (2,5% of all companies).

Pre-processed data

The same process was carried out on the pre-processed data. In this case, the model detected 93 topics, from which topic 2 seemed to be the most related to the CSR concept. It was the third topic taking into account its size (379 documents).



Figure 5. The wordcloud for the topic 2 (in Polish)

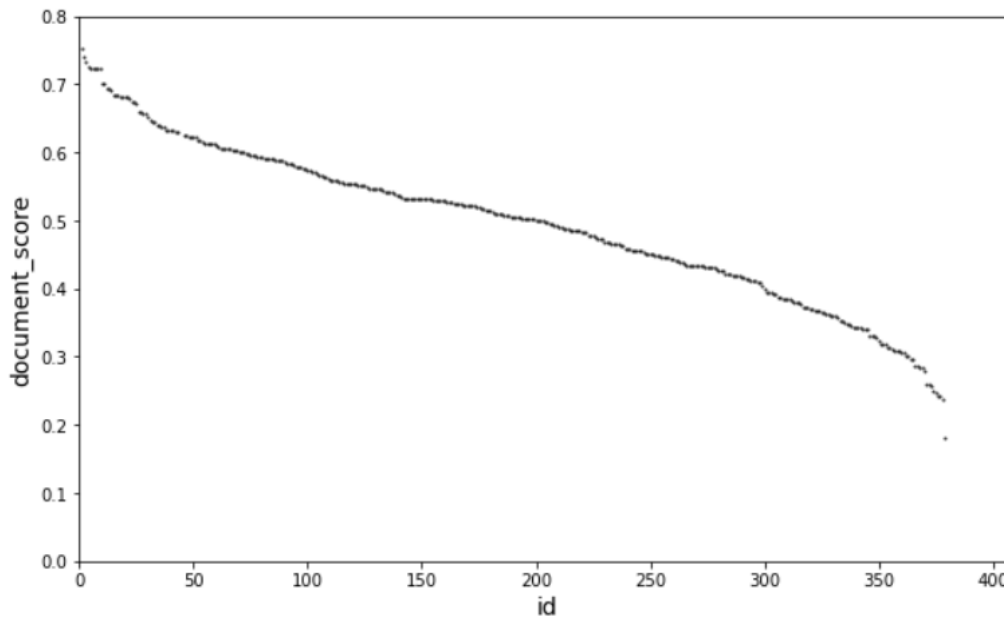


Figure 6. The scatter plot for the topic 2

The number of unique companies which are assigned to the topic 2, closely related to the CSR concept is 246 (2,2% of all companies).

The above means, that the results of the model trained on the raw data and on the pre-processed dataset are very similar. The random assessment of the assignments of texts to the CSR-related topics confirmed these outcomes, however further research as well as more advanced validation methods should be used when considering production of official statistics.

Reproducibility of the results

Basically, the Top2Vec model is stochastic, thus to achieve the reproducibility of the results for the same dataset it is necessary to set a random seed in the UMAP algorithm. Unfortunately, the Top2Vec library does not provide the possibility to set this parameter via API and it can be only done in the source code³.

References

Delden, Arnout van; Windmeijer, Dick; ten Bosch, Olav (2019): *Finding enterprise websites*. Bilbao (European Establishment Statistics Workshop).

Kuhneman, Heidi; van Delden, Arnout; Summa, Donato; Gussenbauer, Johannes; Ils, Alexandra, Loytynoja, Katja (2022): *URL finding methodology*. Joint report for Work Package 2 (Online Based Enterprise Characteristics) and Work Package 3, Use Case 5 (Business register quality enhancement) V.5.0. ESSnet Trusted Smart Statistics – Web Intelligence Network. https://ec.europa.eu/eurostat/cros/content/url-finding-methodology_en

Individual report: Statistics Flanders

The individual report from Statistics Flanders is written in paper-form, and was submitted for presentation at NTTS 2023.

Unsupervised ranking and categorisation of companies using web scraping and machine learning

Michael Reusens, michael.reusens@vlaanderen.be

Keywords: web scraping, natural language processing, official statistics, machine learning

INTRODUCTION

This paper presents a method to automatically categorise companies based on the text scraped from their website. The method is demonstrated by applying it to categorising companies as being active in the domain of artificial intelligence (AI) or not.

This study has been set up to evaluate if the text scraped from company websites can be used as a complement to existing data sources to produce official statistics. Today, most official statistics are produced using survey data (such as the Community Innovation Survey [1]) and administrative data sources. The web scraping methodology proposed in this paper has the following advantages compared to these traditional data sources. First, scraping instead of surveying alleviates the response burden of companies. Second, using the web scraping method, companies can be surveyed at any desired frequency. This leads to more up-to-date data, resulting in increased quality and timeliness of the company statistics. Finally, our approach can be generalised to any categorisation of interest, resulting in the ability of statistical organisations to create new company statistics relatively quickly.

In recent years, there have been other studies that show the opportunities of web-scraped company information for use in statistics production [2,3]. Our study contributes to these existing studies in the following ways. First, we apply and evaluate a method that to the best of our knowledge has not yet been applied for use in official statistics. Next, our approach allows for the ranking of companies without any labelled data, and for the categorisation of companies with only a small amount of labelled data. Existing methods of categorising companies based on their website texts require a relatively large set of labelled data. Finally, the method evaluated in this paper is generically applicable to a large amount of different company categorisations.

METHODS

As a specific use case to demonstrate the method, we discuss our experiments categorising companies as being active in AI or not. Keep in mind that the same method can be applied to other company categorisations (e.g. active in bioeconomy or not, being a transportation company, etc.).

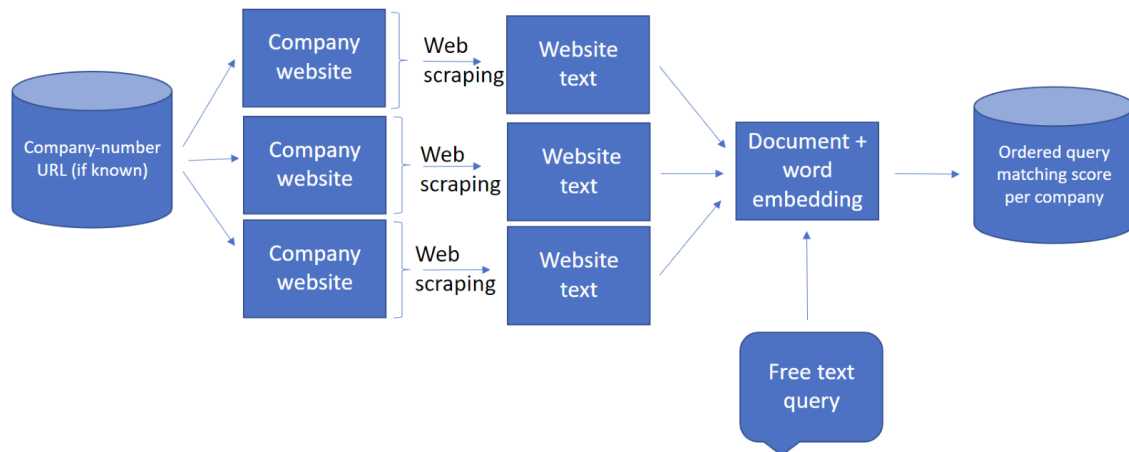


Figure 1. Overview of the method

Figure 1. shows an overview of the method. In the following subsections, we will discuss each step in the figure.

2.1 Input data

The input data for the method is a list of companies described by their company number and URL if known. For our experimental dataset we used the list of all companies with a legal entity in Belgium, excluding one-person businesses. The one-person businesses are excluded for privacy reasons. This results in a dataset of 914,000 Belgian companies, of which 320,000 had a known URL. The URLs in this dataset were purchased from a business partner. The challenge of automatically finding and validating company URLs is out of the scope of this study.

2.2 Web scraping and data cleaning

For each of the URLs, the visible text from the homepages was scraped using Python in combination with the requests [4] and beautifulsoup [5] libraries. Scraping text from deeper webpages (such as the ‘about us’ page) is an improvement we will tackle in future work. The following cleaning steps are performed on the resulting texts. Only texts in English and Dutch are retained. Language detection are done using the langdetect [6] library in Python. Only texts with more than 50 characters are retained and stop words and a custom list of web-technology words are removed from the texts. The goal of the cleaning is to obtain texts that are dense in information on company activities. After scraping and cleaning, the dataset is reduced to 200,000 clean texts.

2.3 Document and word embedding

The cleaned texts and individual words are jointly embedded using a fine-tuned version of a pretrained multilingual transformer model [7]. For the implementation of this joint embedding the Top2Vec [8] library is used. The resulting embeddings of company texts and words lie close together if they are semantically similar and far apart if they are dissimilar. The embedding model used in our demonstration allows for the combination of 16 different languages for which the model is pretrained. This makes it trivial to deal with different

companies using a different language on their website (as long as the languages are included in the pretrained set of languages).

2.4 Query selection and embedding

Next, a free text query is defined that describes the categorisation of interest. For our example use case, AI, we concatenated the Wikipedia introductions of 'artificial intelligence' and of 'machine learning' with a description of 'data science' found on an IBM webpage. There are infinitely many options to define a query. Designing a method to find the optimal query for a given categorisation will be tackled in future work. Once it is defined, the query is embedded using the same model as the company texts.

2.5 Company ordering and categorisation

The distance between the embedded query and each embedded company text is calculated. This allows for a ranking of companies with the first company having a website text that has the shortest distance to the query and the last company having the website text with the highest distance to the query. If a sorted list of companies given a specific activity is the desired output, the method can stop here and is completely unsupervised. For business-facing government agencies this is already a valuable outcome. If a binary categorisation is needed, such as for statistics production, some labelled data is necessary, making the approach semi-supervised. To go from ordering to categorisation, a cut-off score must be decided. Companies with a shorter distance to the query than the cut-off are considered part of the category, the others as not part of the category. The choice of cut-off can be made by optimising recall@N and precision@N, with N the number of companies being included by the cut-off.

RESULTS

In order to validate the approach, a dataset of 50 known AI companies was created.

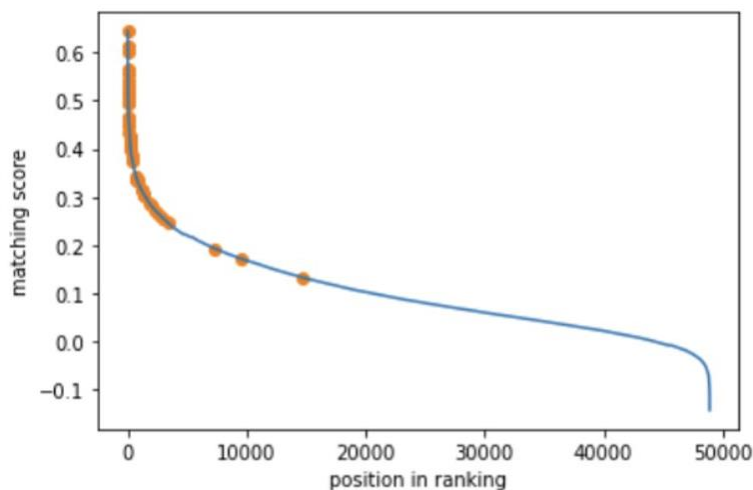


Figure 2. Matching score per company (blue) and placement of known AI companies (orange)

Figure 2. shows the sorted matching scores per company given its position in the ordering. This figure also shows that the known AI companies are placed highly in the ordering, which is

desirable. The quantitative performance of the method can be seen in Table 1. The median ranking of the known AI companies is 199, with median score of 0.41.

Table 1. Quality of ranking known AI companies

	Mean rank	Median rank	Mean score	Median score
Known AI companies	1290	199	0.40	0.41

Inspection of 50 randomly selected unknown companies that were ranked higher than the median rank showed only 1 company not active in AI. An inspection of 50 random companies with matching score lower than the lowest-scoring known AI company showed no companies active in AI. This indicates desirable false-positive - and false-negative rates.

CONCLUSIONS

The method presented in this paper demonstrates a new way of complementing traditional company information with website texts. Our first experiments show that the method is successful in ordering companies active in AI. Following these promising results, we identify the following gaps in this paper for future research. First, a more elaborate validation approach needs to be set-up to assess the quality of the ordering and give guidance to the choice of cut-off point for categorisation. In order to do so, the ordered list of AI companies is currently being used by business-facing government consultants, who provide further feedback on the quality of the results of this method. Next, the performance of the method for other categorisations should be verified. Besides AI, we are investigating categorising companies as being active in bioeconomy and circular economy. Finally, further research should be done on the general properties of website texts for the production of company statistics. For example, bias in the type of companies having a website could be of concern.

References

- [1] <https://ec.europa.eu/eurostat/web/microdata/community-innovation-survey>, (2022)
- [2] P. Daas and S. van der Doef, "Using Website texts to detect Innovative Companies.", (2021).
- [3] https://ec.europa.eu/eurostat/cros/content/WPC_Enterprise_characteristics_en, (2022)
- [4] Python Software Foundation, <https://requests.readthedocs.io/en/latest/>, (2022)
- [5] L. Richardson, <https://www.crummy.com/software/BeautifulSoup/>, (2022)
- [6] <https://github.com/Mimino666/langdetect> , (2022)
- [7] <https://tfhub.dev/google/universal-sentence-encoder-multilingual/3> , (2022)

[8]D. Angelov, "Top2vec: Distributed representations of topics.", arXiv preprint arXiv:2008.09470, (2020).