ESSnet Trusted Smart Statistics – Web Intelligence Network

Grant Agreement Number: 101035829 — 2020-PL-SmartStat

Work Package 3

New Use-cases

Deliverable 3.11:

UC5: Report on methodology and results to use online data for business register enhancement

Version, 2025-02-22

Prepared by:

UC coordinator:

Arnout van Delden (CBS - Netherlands, a.vandelden@cbs.nl) (until 1-08-2022)

Olav ten Bosch (CBS - Netherlands, o.tenbosch@cbs.nl) (from 1-08-2022)

Contributors:

Ville Auno – Statistics Finland

Olav ten Bosch – Statistics Netherlands (editor) o.tenbosch@cbs.nl

Arnout van Delden – Statistics Netherlands

Johannes Gussenbauer – STATA, Austria

Pär Hammarström – Statistics Sweden

Alexandra Ils – Statistics Hesse

Remy Kamali– Statistics Sweden

Heidi Kühnemann – Statistics Hesse

Katja Löytynoja – Statistics Finland

Naomi Schalken – Statistics Netherlands

Pieter Vlag – Statistics Sweden

Wictoria Widén – Statistics Sweden

Nick de Wolf – Statistics Netherlands

This document was funded by the European Union.

The content of this deliverable represents the views of the author only and is his/her sole responsibility. The European Commission does not accept any responsibility for use that may be made of the information it contains.

Web Intelligence Network

Funded by the European Union

Contents

1 Background

This document is part of the Work Package 3 (WP3) *New use-cases* from the ESSnet Trusted Smart Statistics – Web Intelligence Network project (TSS-WIN). The overall objective of WP3 is to explore the potential of new types of web data sources for official statistics. The work is organised in a number of use-cases (UCs), each focused on a specific application. The use-cases being explored are:

- **UC1** Characteristics of the real estate market
- **UC2** Construction activities
- **UC3** Online prices of household appliances and audio-visual, photographic and information processing equipment (and generalising the data-collection to other activities)
- **UC4** Experimental indices in tourism statistics (hotel prices)
- **UC5** Business register quality enhancement
- **UC6** Faster Economic Indicators using new data sources

This deliverable focuses on UC5 specifically and describes the topic starting from a high-level perspective down to the many building blocks that were developed in the project and could be useful for other statistical organisations. This report combines all the work performed by the UC5 members on different topics, organised by topic.

The report starts with a high level introduction into the subject in chapter 2, identifying two main subtopics: URL finding and enhancing the business register. These topics are explained in more detail in chapters 3 and 4 respectively. Each chapter starts with a more detailed explanation of the generic concepts that are to be identified and the possible choices to make, followed by a description of the achievements of the partners in this project in that particular topic. Chapter 5 addresses some cross-cutting subjects. Finally, in Chapter 6, a general discussion is presented keeping an eye on the future.

2 Introduction

National Statistical Institutes (NSIs) and Other National Authorities (ONAs) worldwide maintain statistical business registers (SBRs), which are comprehensive databases of information of statistical units such as enterprises within their respective jurisdictions. For example, these registers contain detailed information about each enterprise, including its size, location, economic activity, and administrative details. They also capture the relationship with the legal unit and other important relationships among enterprises. The SBRs serve as an essential sampling frame for statistical surveys and are indispensable for producing official economic statistics. It is a central production component, not just another statistical product. Improvements in this register pave the way to control representational errors in all business statistics and contribute to overall quality in official statistics. This underlines the relevance of the use case described in this document: to use online data to enhance SBRs by incorporating better, more detailed, or new information that may be difficult or impractical to acquire through traditional methods. Enterprises leave a trail of digital footprints or traces across the internet, such as websites, media advertisements, product listings, customer care interactions, and job postings, which may offer valuable insights into enterprise operations and characteristics. Moreover, these digital traces need not be created by the enterprise itself; descriptive information maintained in community media such as Wikipedia entries or in articles about the statistical unit may also be valuable. An important challenge for this particular use case is to find relevant traces and interpret them as well as possible in a statistical context keeping in mind that data sources are of a different nature than their traditional administrative counterparts and thus should be explored with new methods.

Although all digital traces are potentially interesting and could be explored further, for practical reasons in this project we initially focused on one obvious starting point: the website(s) of the statistical unit itself. Unfortunately, the URL of this website is not always known from the SBR. If missing, or not reliable enough, the missing ones must be found in the so-called URL finding phase (A). In the next phase (B), statistical variables can be derived or improved from the data found at the online data sources. Online data can be used in both phases. Figure 2.1 shows this graphically.

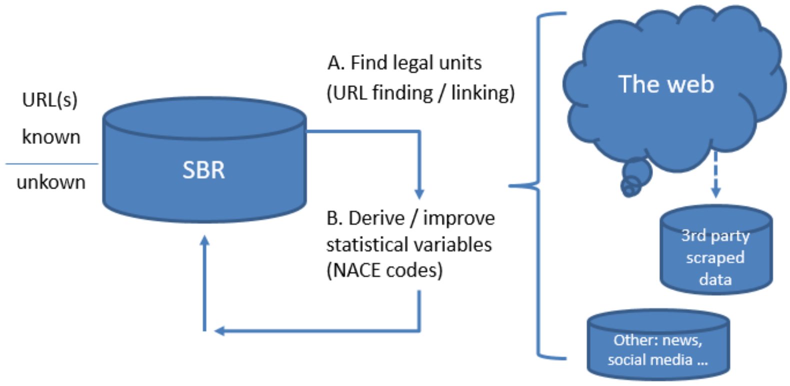

Figure 2.1: High-level view on business register enhancement

The process of SBR enhancement and in particular finding URLs matching enterprises in the SBR can be – or maybe should be – an iterative process, see ten Bosch et al. (2023). In a first phase, a certain percentage of candidate URLs are retrieved from various online data sources, which then have to be validated before they can be added to the SBR. Validation can be done in different ways, based on information retrieved in the search phase, like the short description that comes with the search item (typically called snippet), or by an extra scraping step, or otherwise. In an iterative setting with periodical search requests executed over time, the link from SBR units to URLs can gradually be increased, but there will probably always be a category of units for which no URL can be found. This may be due to the fact that such units do not need a website, use other digital communication channels or simply have no online presence. This has to be taken into account when deriving indicators on the target population. Figure 2.2 sketches the iterative process in more detail.

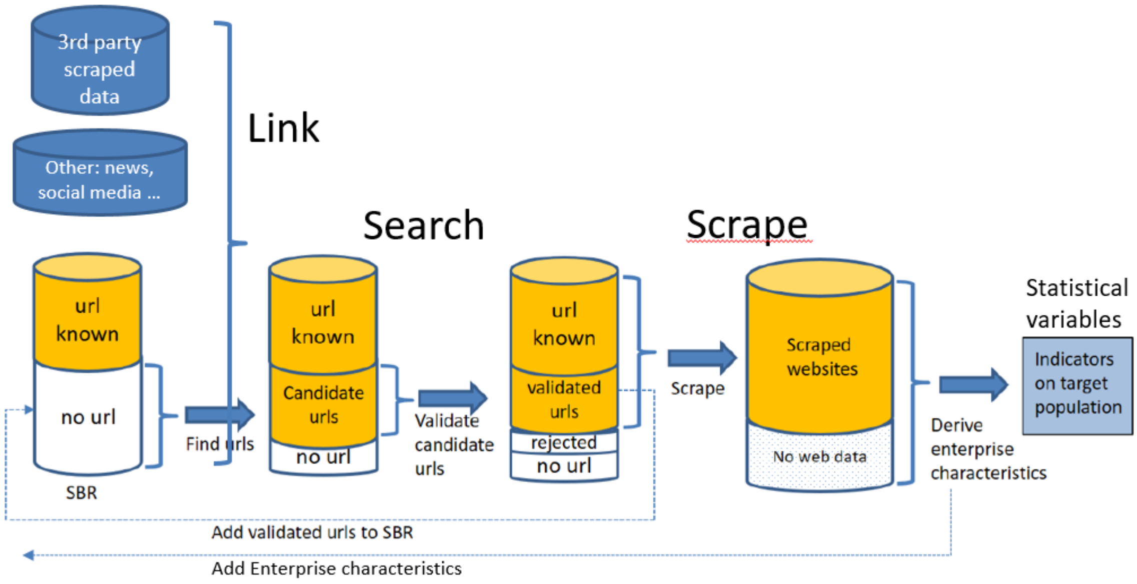

Figure 2.2: High-level process of business register enhancement

This chapter shows the highest level view on the art of business register enhancement. Obviously all steps to be performed, choices to be made, and implementation challenges to face are highly dependent on the state and completeness of the SBR in the statistical organisation and on country-specific availability and use of online data. Therefore, in the next two chapters on URL finding and enhancing the business register respectively, apart from the generic aspects, we also present a number of national experiences of members of the WIN consortium with the approach.

3 URL finding

3.1 Introduction

In the URL finding phase (Figure 2.1, phase A) multiple online data sources or services can be used to locate URLs for statistical units or to verify known URLs. One way is to use search engines to find or verify URLs. This can be done by executing a search query containing the name of the unit, optionally supplemented with contact information such as address or a Chamber of Commerce (CoC) or tax identification number. Executing multiple search queries per statistical unit with different composition and possibly on multiple search engines can increase the results, but should be weighed against the costs or resources. Search engines should be used with caution, as identifying information contained in the query could be identified in web server logs (search engine leakage). This risk can be reduced by carefully designing the queries, spreading them across different search engines, or entering into a non-disclosure agreement with a search provider. Hence search engine leakage is a serious but manageable concern.

To automate the search process, it is recommended to use a (paid) application programming interface (API) for searching rather than interpreting a human-readable search result pages, although both approaches are possible. In any case, search results must be interpreted to select the best match. This can be done using the snippet (a short textual extract of the results page) or by scraping the URLs returned by the search engine. However, the latter approach involves an extra step which can increase costs and slow down the process. In countries where enterprises are required to include identification numbers on their websites, these can be used for direct and exact linkage to the business register. In the absence of such legislation, machine learning techniques have been shown to be useful for selecting relevant search results. Based on a labelled training set of valid and invalid search hits, a model can be trained to capture the search engine behaviour for a particular search engine. The set of legal units in the business register with known URLs can serve as a training set. Since search engines evolve over time, the model must be retrained periodically.

Another source of URLs is web data collected by third parties for their respective business goals. Reusing such data can save resources. However, a (paid) agreement must be made with the third party, and the dependence on the third party must be managed. Therefore, using third party data is usually only useful if the added value of the data is considerable. Experiments have shown that that using third-party data for URL finding is valuable, but should be monitored over time, as the third-party might change scraping and processing methodologies based on their own business strategy.

Another online source for URL finding is domain registry data, which is a register of domain names and IP addresses that exists in all countries and for cross-country domains such as .com. This data can be useful for deducing domain ownership, but the degree of openness of this data varies per country and domain.

Yet another online source for URL finding is data that is present on online maps. It is of great importance for companies to have their location, opening times and contact information correctly present at the most popular mapping apps in their respective countries. This also holds for the additional information to be found on such map items, and in in many cases there is also a direct link to the website of the enterprise.

For all techniques applied in the URL finding phase it is important to note that the link between the online data source and the legal units can be many-to-many. For websites this is easy to see, after all a business may operate multiple websites depending on its business activities and vice versa smaller businesses may choose to advertise their services via business portals that offer hosting to many similar enterprises instead of setting up and maintaining their own websites.

Funded by the European Union

Finally, for all methods it is also important to measure the quality of the relation between the URL and the legal unit.

At the start of this ESSnet, a literature study was performed on the topics of interest. Concerning URL finding methods two papers were identified, which were taken as an input for the work done. More details on the literature review can be found in Section 5.1.

3.2 URL finding methodology

In general, URL finding involves executing automated requests to a search engine and interpreting the results to identify the correct enterprise URLs. The methodology of this concept has been described in the WP2 report of Kühnemann et al. (2022), publicly available on CROS portal at the WIN page: https://cros.ec.europa.eu/book-page/web-intelligence-network-reports. Here we summarise the main findings. We refer to enterprises as the units to search for, but this includes legal units.

If an NSI has hardly any URLs to start with, there are three ways to obtain a train set for the URL finding model:

1. Manually search for (correct) URLs of enterprises.
2. Manually classify URL search engine results whether obtained URLs are correct or not
3. Determine whether scraped URLs can be matched by exact (direct) linkage or not

Option 2 has the advantage that one directly labels obtained search results, whereas in option 1 there can be a mismatch between the correct URLs and the results from the search. The disadvantage of option 2 is that the search results can be somewhat selective and they depend on the search engine used. In practice, option 2 is often used. The training data obtained by option 3 is usually very biased since it only considers URLs that can be matched by exact linkage as being a correct URL. Finally one should be aware that the train (and test) set should be representative for the target population.

Usually, this URL finding approach contains four parts:

1. sending search terms to a search engine,
2. scraping the resulting URLs,
3. extracting the relevant information from the scraped data and
4. creating a machine learning or rule-based model to link websites to enterprises

More specifically, these four parts consist of:

Part 1. Both selecting the right search terms and the right search engine has a great impact on the accuracy of the obtained URLs, see examples in the WP2 report. What the best search terms are actually varies somewhat per country because, depending on the country, enterprises may store different sorts of identifying information on their website. Different search engines also gave different results, but one may prefer not to rely on a single commercial search engine because it may store all the entered information. Note that some obtained URLs are not relevant because they do not concern dedicated business websites. One can make a block list of such URLs and block or remove them from the search results. Usually, this block list has to be manually maintained.

Part 2. The found URLs should then be scraped to obtain the available contact information. It is obligatory for EU enterprises to include this information on the website, which is usually found on the landing, the contact or on the imprint page. Before scraping these pages, one should check for the exclusion protocol on the robots.txt page which should be respected. The search results can contain different links to the same URLs, so one should try to deduplicate the search results, including checking for URLs that are redirected to other URLs. Scraping HTTP errors may occur, and it is best to record them in order to evaluate the frequency of these errors. Scraping can either be done using HTTP GET requests or JavaScript rendering services. HTTP GET requests are quicker and need less bandwidth. However, on some websites, the complete content may only be loaded when rendering JavaScript, e.g. with an automated browser. The choice of scraping methodology therefore means deciding on a trade-off between resource usage and data completeness.

Part 3. Depending on the country, different identifying variables in the business register will be available for comparison with website data. Enterprise data often consists of the name, address, register/tax IDs and contact information (phone number, email). This information can be compared with the scraped website information by either trying to do exact string matching or by some string distance function. Before doing this, it is important to pre-process the data (removing duplicate whitespaces, lowercasing words and letters, removing language specific characters and so on) because the spelling in the business register may differ from the one on the website. Besides agreement on the identifying variables of website and business register, one might use other characteristics for the model such as search position. The search snippet can also be used for the comparison.

Part 4. Because some enterprises do not have a URL, one can consider three different options for building the model:

a) limit oneself to the cases of part 1 where actual URLs have been found and train a model which predicts whether URLs are correct or incorrect
b) build two models: one model that predicts whether an enterprise has a URL and a second model that finds the correct URL
c) include in part 1 cases where URLs were found and cases where no URL was found and train a single model that can also predict if ‘no URL’ is correct

Option b) is ideal in the sense that different features can be used for both models. One can use rule-based models, classical models or neural-net type models.

There can be some variations to URL finding. First, an alternative to the scraping in part 2 is to only use the information that is shown in the snippet. Second, if email addresses are known for the enterprise, this can also be very useful information to find the URL of the enterprise because it regularly occurs that the domain name of the website can also be found in the email address. Third, the procedure to search and scrape URLs can also be used to derive characteristics of enterprises such as whether they have e-commerce or not.

3.3 Common elements of URL finding

We have looked into how we could best report about URL finding results in such a way that the different reports of the countries have a number of common elements.

With respect to the linkage of URLs to units in the statistical business register we agreed:

- to specify to what unit type a URL is linked to (legal unit, enterprise, …)
- to report which linkage variables are used (name, phone number, business ID, business name, domain/URL from business register if one exists) and so on
- how to decide whether we accept or reject a potential linkage as being a true match and what threshold to use?

If participants use a linkage method that involves non-unique linkage variables, they should report the importance of each of the variables in the linkage process. For situations in which a partner used both unique linkage variables (like VAT ID) and non-unique linkage variables, it is interesting to know what proportion can be linked through unique linkage variables and what proportion to non-unique linkage variables.

Furthermore, on the result of the URL finding we agreed to report

- The proportion of 'businesses' without URLs and with one or more URLs after applying the method (this can be done for businesses within a specific domain if only URLs of a specific domain are scraped)

When a partner applies the "data collected by others" approach, we also agreed to report the proportion of unlinked records of the external source.

Finally, we agreed on a few optional elements to report. With respect to the "do it yourself" approach, we agreed to optionally report:

- what proportion of scraped websites we were not able to access (due to HTTP errors, sites for sale, robot exclusion protocol)
- how to deal with reCHAPTAs or information that could not directly be accessed by a scraper
- the proportion of accessed sites where one could extract identifying information

With respect to the "data collected by others" approach, we agreed to optionally report:

- The proportion of missingness in linkage variables of the external source and in the business register. This can be done for the full sets or only for the units after linkage.
- After linkage of the units one can report on disagreement of linkage variables of linked units (if that is allowed in the linkage procedure).
- For domain registry data: how many domains are found in the registry and are they correct? To what extent is there overlap between domains found in domain registry data and those found with the „do it yourself" approach.

3.4 URL finding by search and ML at Statistics Austria

Statistics Austria has developed their own URL finding procedure. It uses the Google Search API and afterwards scrapes potential enterprise websites using the R programming language and Selenium.

Google Search API

In an initial step, each enterprise is searched by using the Google Search API which results in a set of candidate URLs. The search string for the API contains the name and address of the enterprise taken from the Statistical Business Register (SBR). For each enterprise two search queries are conducted. The first contains the name of the enterprise, municipality, street, and street number, the second contains only the name and street. In the case of one-man businesses the name of the business is removed from the search string and more details on the business address is used instead. The API configuration includes an extensive list of URLs, which are blocked from the search results. The URLs received from the Google Search are processed further to remove duplicates and keeping country-coded folders for regionalized websites. After pre-processing we receive for each enterprise e_i,$i = 1, \dots, N$ a list of possible URLs $u_{1(i)}, \dots, u_{n(i)}$ to match with.

Given the ICT 2021 population which contained 41430 legal units, the number of URL, after pre-processing, yields roughly 91000 .

Scraping potential enterprise websites

Each of the URLs received from the API is visited using Selenium, through the R package RSelenium (Harrison 2020), and the website source code is collected. With the use of Selenium, a browser can be simulated and JavaScript, embedded on a website, can be rendered. At Statistics Austria a chrome browser is simulated and apart from the user agent, the following browser options are specified:

```
##  [1] "--headless"                   "--disable-gpu"
##  [3] "--lang=de"                    "enable-automation"
##  [5] "start-maximized"              "--no-sandbox"
##  [7] "--disable-infobars"           "--disable-browser-side-navigation"
##  [9] "--disable-blink-features"     "--window-size=1080,1920"
## [11] "--disable-popup-blocking"     "--disable-dev-shm-usage"
```

During the scraping process the robots.txt exclusion protocol is respected and the crawler scrapes the main page and up to 25 subpages. Table 3.4.1 shows the number of links and sublinks scraped as well as the number of errors, timeouts or disallowed links during the webdata scraping of the ICT 2021. Both the share of links visited with an error encountered or prohibited by the robots.txt protocol was very low with less than 1%. The first case only occurred if a website or link was not reachable. The low share of prohibited links is due to the fact that the scraper is looking for specific links on a website and from this selection only few where prohibited by the robots.txt protocol.

Table 3.4.1: Overview scraping

Number Links	Encountered Error (%)	Encountered Timeout (%)	Robots-txt not allowed (%)
954000	0.1292	5.974	0.1539

The crawler is instructed to look for certain subpages, like “imprint”, “contact”, “impressum”, which can contain the name, contact information as well as VAT or commercial register numbers (CRN) of the enterprise who owns the URL. In Austria businesses are legally obliged to identify themselves and in many cases also list their VAT or CRN on their webpages. These register numbers are available in the SBR and represent a reliable source for linking an enterprise to a URL. After a website has been scraped the collected html source code is further processed. Html tags or embedded code is removed and only the text elements are kept. Text snippets are further processed in order to extract any VAT or CRN using regular expressions. In addition, the name of the enterprise in different types of writing and detailed address information like, municipality, municipality code, street name and street number is searched for in the collected website text. These results are saved in a 0-1 variable each and used for probabilistic linkage if no VAT or CRN can be found on the website.

Linking enterprise and website

With the use of VAT and CRN found on a website it can deterministically be linked to an enterprise. This assumes that the VAT or CRN found corresponds to the enterprise which owns the URL. In roughly 41.11% of the cases a VAT or CRN number was found on a webpage which corresponds to an enterprises in the ICT population. This lead to a total of roughly 26500 enterprises being linked to a website.7890 enterprises were linked to more than one URL and for 1381 URLs the VAT or CRN of more than one enterprise from the ICT population were found.

According to the ICT 2021 survey roughly 90.99% of enterprises in the target population have their own website, which indicates that the deterministic linkage falls short of linking almost 1/3 of the enterprises. The reasons for this are diverse and range from the website owner listing incorrect identifiers, listing company name, address, and identifiers from a country different than Austria or the website owner does not list any identifying information. It is assumed that the first scenario only occurs in very few cases. In order to link additional websites to enterprises we use a machine learning model. This ideally requires a manually pre-labelled set of training data to train the model on which was, due to lack of resources, not available. Instead, a training and test set was artificially created using enterprises which can deterministically be linked to a URL. The training data is composed of the set $S_1 = \{(e_i, u_i): u_i \text{ was deterministically linked to } e_i\}$ and the Set $S_2 = \{(e_i, u_i): u_i \in \{u_{1(i)}, \dots, u_{n(i)}\} \wedge u_i \text{ was not deterministically linked to } e_i\}$. S_1 represents the positive and S_2 the negative cases. This assumes that an enterprise which was identified on a website with either the VAT or commercial register number will be identified on all its websites through the same logic. The predictor variable used is composed of the boolean variables which indicate if name and or parts of the address where listed on the URL. The random forest algorithm, see Breiman (2001), using the R-package ranger, see Wright and Ziegler (2017), is applied for this model based linkage. With k-fold cross validation on the training data a reasonable cut off value was determined that describes if a potential link is a true match or not. The results of the cross validation, average scores on accuracy, precision and F1 score, are shown in Table 3.4.2. An average precision of 0.95 and average F1 score of 0.92 can seem high at first glance but one has to keep in mind that the training data is quite selective and it does not seem plausible that a complex task like linking enterprises and web pages can be modelled adequately using such a simple set of features. Thus, one should interpret these values with caution.

Table 3.4.2: Average scores after cross validation

	Value
Accuracy	0.8919013
Precision	0.9546828
F1 Score	0.9156977

After applying the random forest model, an additional 6548 URLs can be linked to 5074 enterprises. According to the deterministic and model base linkage around 76.76% of the ICT population owns a website. Respecting enterprise groups, e.g. an enterprise also “owns” a website and it belongs to an enterprise group where one or more enterprises own a website, about 80.01% of enterprises own a website. This result is roughly 10% below the ICT survey results which lists 90.99%.

3.5 URL finding by search and ML at Statistics Hesse

The URL finding phase is especially important to Statistics Hesse, since there are no URLs available from German Official Statistics data sources. This section is structured as follows: first, we provide a methodological overview of the URL finding process; next, we discuss how we obtained training data and compared our results to third party data; and finally, we detail the IT infrastructure that supports the process of URL finding.

Methodological overview

The URL finding methodology at Statistics Hesse, see Kühnemann (2023), closely follows the approach explained in section 3.2. We will give a short summary on our implementation of each of the four parts mentioned above:

1. sending search terms to a search engine,
2. scraping the result URLs,

3. extracting the relevant information from the scraped data and
4. creating a machine learning or rule-based model to link websites to enterprises

Part 1: Google search API is used with the name and municipality of the legal unit as search terms. Up to 10 results per search query are obtained. Additionally, an extensive list of URLS, known as blocklist, are excluded from the search results either using the API configuration or after scraping. These URLs contain enterprise data without being correct enterprise websites (e.g., e-commerce platforms, yellow pages).

Part 2: Scraping is done with the JavaScript rendering service Splash. German enterprises are legally required to have an imprint (“Impressum”) page on their website, which lists basic contact details and identifying information about the enterprise. Using regular expressions, we search for links within the website that contain strings like “Impressum” and scrape those subpages since it is assumed that they will give the best information for linkage. Each scraping request is logged, so that the frequencies of errors during scraping can be evaluated.

Part 3: After minimal pre-processing (removing css stylesheets, JavaScript code and multiple whitespaces as well as lowercasing), the following features are created from the data:

- Regex and exact match features: Available information from the SBR is compared with the website text using regular expression or exact matches. For this purpose, we use the name, address and several register numbers (Chamber of Commerce, European VAT ID, German tax ID) of legal units.
- String similarity features: We compute the Jaro-Winkler similarity of the name of the legal unit and the HTML title of the websites.
- Search meta data: The Google search position of each URL, if the query has been corrected by Google (e.g. if an abbreviation got rewritten to its long version), the total number of search results and the query time are additional features.
- Company meta data: These enterprise level features indicate if a legal unit has missing values for the VAT ID, German tax ID or trade register ID.

All features are aggregated to domain and legal unit level using maximum, minimum and mean (only continuous features) aggregation.

Part 4: The features are first used as input to a machine-learning model that predicts if an enterprise has a website in general. The resulting model is calibrated using Platt scaling and the scores are used as an input to a second model to predict which of the domains were correct. See Figure 2.5.1 for an illustration. We use each time a five-fold cross-validation with 80% training and 20% test data. The crossvalidation consists of an inner loop to select the best features with recursive feature elimination and to tune hyperparameters and an outer loop, which compares the model performances of the best model-features-hyperparameter combinations. After the best model is identified with this procedure, the model is retrained on the full training data and used to predict websites with unknown URLs. We compare the following ML models Logistic Regression, Support Vector Classifier, Random Forest, Extra Trees, Gradient Boosting, Adaboost and XGBoost (scikit-learn and xgboost Python packages).

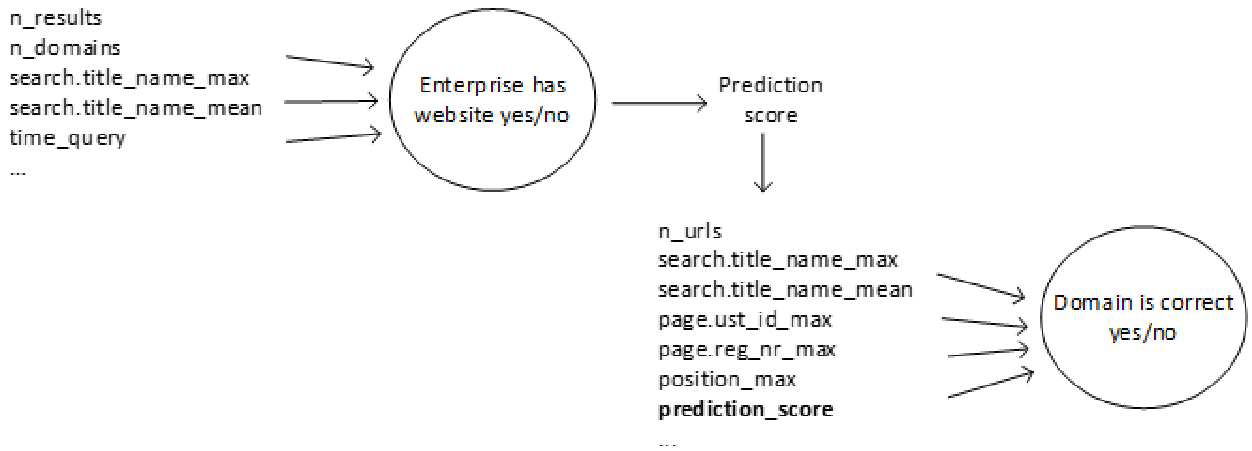

Figure 2.5.1: Two-step ML approach for URL finding

Training Data and Evaluation

To obtain training data and evaluate our URL finding approach, we used manual annotation and a comparison to third party data.

Evaluation can be done in two ways:

- On the website level: was a website returned by Google correctly classified?
- On the enterprise level: were the correct website(s) assigned to the enterprise?

The website level can be evaluated by the usual evaluation criteria for binary classification problems, e.g. precision, recall and F1 score. However, this is not possible on the enterprise level, since there are three possible errors:

- An enterprise with a website was assigned the wrong website
- An enterprise with a website was no website assigned
- An enterprise without a website was assigned a (wrong) website

We therefore calculate the percentage of correct enterprises: the percentage of enterprises where either the correct website(s) were assigned or no website was assigned because the enterprise does not have a website. If an enterprise has more than one website but not all of them were found, this is not counted as error.

Training data from manual annotation

As a starting point, we created a training set of 2000 legal units manually. We decided to focus on retail trade enterprises and excluded complex enterprises (enterprises that consist of more than one legal unit). Therefore, these 2000 legal units are a sample from the population of legal units in retail trade (ca. 30,000), stratified by 4-digit NACE codes. Units with 10 or more employees were oversampled.

Of these 2000 legal units, 1466 had at least one website according to the manual search. We first tested if the VAT ID could be used as sole criteria for linking legal units with URLs. In this deterministic approach, if the VAT ID was found on the website, the website was assumed to be correct. The results show that this approach achieves a high precision, but relatively low recall. Only 64% of the correct websites contain the VAT ID. The best performing ML algorithm, Gradient Boosting, achieved a substantially higher recall (77%)

than the deterministic approach and even a small increase in precision (93%). In Table 2.5.1 columns "Retail Trade" show the evaluation results on the Retail trade training set.

Comparison to third party data

In a cooperation with Destatis, we used data from the Dutch company DataProvider to evaluate how well URL finding performs in comparison to third party data.

The German federal Statisticsl Office (Destatis) received DataProvider data of websites with German-identified enterprises. The data contained information on name, address, type of legal entity, trade register number and VAT ID. It was Destatis' requirement that every entry in this data must have either a VAT ID or a trade register number. In total, DataProvider was able to deliver 1.5 Mio URLs that fit this requirement. Destatis linked this data to the business register using deterministic linkage with trade register number and VAT ID as linkage variables. They were able to link URLs to 12% of all legal units in the business register. Of all 1.5 Mio URLs from the DataProvider delivery, ca. 0.75 Mio URLs could be linked this way.

In order to evaluate the quality of DataProvider data and compare it to URL finding, business statistics experts from Destatis manually checked websites for ca. 1700 legal units. Results show that if DataProvider found a URL, it was often correct. However, the number of found URLs was rather small. Of all enterprises in the evaluated data, for 44.9% at least one URL could be found, while the dataset from DataProvider showed no found URL. A wrong URL was assigned to 2.1% (given a URL exists) and 1.5% were assigned a URL even though they do not have one. In Table 2.5.1, the column "DataProvider" shows the evaluation results for the manually checked sample.

Our URL finding approach returns slightly more errors (because it assigns more incorrect URLs to enterprises without URL), but is much more complete. We used the manually checked sample for the DataProvider data to create new training data for all NACE codes. The classification results for this new training data can be found in Table 1, columns "All NACE codes".

In conclusion, the URLs returned by DataProvider contained slightly fewer incorrect URLs than the results of URL finding. However, DataProvider data was much less complete than URL finding results. This is in part explained by the linkage process: not all enterprises have a trade register number or VAT ID. Also, DataProvider only delivered URLs where a trade register number or VAT ID could be extracted. Despite the legal obligation to do so, not all enterprises in Germany list this data on their website.

Table 2.5.1: Evaluation of URL finding and DataProvider results

Evaluation		Retail Trade		Data-Provider	All NACE codes	
		Deterministic approach	ML		Deterministic approach	ML
URL level	Precision	0.91	0.93		0.86	0.90
	Recall	0.64	0.77		0.61	0.86
	F1	0.76	0.84		0.71	0.88
Enterprise level	% Correct	0.66	0.88	0.52	0.67	0.85
	Wrong URL	0.04	0.01	0.02	0.04	0.01
	No URL found	0.29	0.07	0.45	0.27	0.09
	URL assigned to enterprise without URL	0.01	0.04	0.02	0.01	0.04

IT Infrastructure

As Statistics Hesse set the goal of implementing web scraping into its production processes, several new demands to the IT infrastructure emerged in the following areas:

- Internet access: Unrestricted internet access with sufficient bandwidth is essential to ensure that the scraper can reach all websites. Additionally, the scraper should also have its own IP address, so that, in case it gets blocked by a website, this does not affect other internet users at Statistics Hesse. Blocking could happen because webscraping might be perceived as a possible DDoS attack.
- Automation: The data collected via web scraping needs to be automatically transferred to the internal environment. Furthermore, the initiation and termination of the web scraping process should also be automated.
- IT security: The infrastructure must be protected from external attackers and from potentially malicious websites.

Establishing and administrating this webscraping infrastructure requires close cooperation between classical IT and data scientists. The data scientists needed to familiarize themselves with the fundamentals of IT security and architecture to accurately define the requirements.

As a result, a dedicated webscraping environment was established at our IT service provider, separate from the internal infrastructure. A demilitarized zone (DMZ) with a proxy server, through which all webscraping requests are routed, protects the environment from external threats. All webscraping servers are virtual machines running Red Hat Enterprise Linux, with regular backups managed by the IT service provider. The environment includes a testing, preproduction and production area. Each area consists of an application server running the webscraping applications and a database server for storing the data. Data transferral is automated by servers within the internal infrastructure, e.g. using cron. Cron is also employed to automate the webscrapers themselves.

Setting up this infrastructure presented some challenges. One significant challenge was identifying suitable software for the proxy server that our IT service provider could support. While nginx, a commonly recommended choice for webscraping, was not available in their software catalogue, the initially installed Apache Reverse Proxy proved inadequate for massive webscraping. Ultimately, we reached a consensus on using Squid as the proxy software.

Another still ongoing challenge is selecting a suitable browser for scraping. The first browser we used, PhantomJS, has been discontinued. We then switched to Splash, a lightweight web browser written in Python. However, since the latest Splash release dates back to 2020, we are currently evaluating other options, including Playwright, Chromote and Selenium Grid.

3.6 URL finding by search and ML at Statistics Sweden

The initial plan in Sweden was to utilize the Sweden’s Business Register (SSBR) for scraping business websites. However, it was soon discovered that the system did not contain URLs. Due to limited resources and legal constraints related to using registry data in Google web searches, Statistics Sweden (SCB) sought commercial companies capable of providing a dataset of URLs linked to business organization numbers. The requested dataset was ultimately provided by DataProvider.

Pre-Feasibility Study

Based on this dataset, a pre-feasibility study was conducted with the following steps:

- Names and organization numbers of enterprises were randomly selected from the SSBR
- A Python-script using the 'googlesearch[1]' package was developed.
- The information was analysed.

The study revealed that the dataset from DataProvider did not meet quality expectations required for statistical linking and had a low coverage rate of about 11%. However, the method of using business name and address in Google searches proved promising for accurately finding URLs, with an average accuracy rate of 80-85% based on a sample of 1224 businesses. Key findings from the study included:

- Most enterprises in Sweden have their organization number mentioned on their website, allowing for validation of the found URLs.
- In about 5% of web searches, the found URL was not directly linked to the correct enterprise but rather to websites of branch organizations or information-collecting websites. This issue was particularly prevalent among small enterprises in traditional sectors (e.g., construction, sports, independent health workers).
- Approximately 70% of the URLs found had some legal or information security protection against scraping, raising concerns about the legality and ethics of this method. Further investigation into the legal implications is necessary. Due to this grey area, the focus was on the 'main page' and 'about us' sections, which have fewer legal and ethical considerations.

Project Phase and Further Developments

During the project phase, URLs were introduced as a variable in the SSBR. As of Dec. 4th 2023, it contains 1 332 640 operating businesses but only 309 (0.02%) URLs, most of which represent the public sector.

There is a clear need to improve this coverage and collect more business URLs. One evaluated approach was using the Google Custom Search API to find URLs based on the SSBR variables 'company name' and 'location'. At the time, unclear internal Open-Source policies prevented the use of an already developed Python package by Statistics Netherlands[2] for this purpose. Consequently, Statistics Sweden developed an initial proof of concept using Python.

Preliminary Results

The preliminary results of the proof of concept were positive, indicating that this approach may be viable. It was tested with a sample of 100 URLs from the SSBR and 95% of them were correctly identified.

3.7 URL finding by linking 3rd party data at Statistics Netherlands

Although Statistics Netherlands (CBS) has developed a URL finder using search engines, it is not currently used on a large scale. The URL finding method is still available, but since June 2019 CBS has paid access to a data set with URLs of Dutch businesses collected by the external company: DataProvider (DP) which – at this moment - offers a reasonable effective way to find missing URLS. In addition, CBS has URLs obtained

[1] PyPi library: https://pypi.org/project/googlesearch-python/

[2] Software for finding websites of enterprises using search and ML: https://github.com/SNStatComp/urlfinding

from the CoC because legal units (LUs) are asked to report their URL (if they have any) when they register to the CoC. Finally, a sample of enterprises receives the yearly ICT survey. One survey question is to report the URL, so that information may be used in the near future.

Table 3.7.1. Identification variables in DP and in the SBR (Legal Units) and scoring weights

Dataprovider	**SBR (Legal units)**	**Original weight**	**LR weight**	**Updated LR weight**
CoCnumber	CoCnumber	500	5.02	5.04
Hostname	Website	500	1.67	1.67
Domain	Website	400	5.20	5.15
Primary Email	Email	200	1.86	1.80
Secondary Email	Email	100	1.16	1.12
Zipcode	Zipcode	100	0.71	0.71
Zipcode + Housenumber	Zipcode + Housenumber	X	X	0.49
Zipcode + Housenumber + Additions	Zipcode + Housenumber + Additions	X	X	0.42
Primary Telephone	Telephone	200	1.00	0.94
Secondary Telephone	Telephone	100	-0.48	-0.53
Primary Telephone	Mobile	200	1.78	1.78
Secondary Telephone	Mobile	100	0.42	0.40

The DP-URLs are linked to LUs by linking identification variables DP scrapes from the websites to those of the SBR. Those identification variables are listed in Table 3.7.1. DP distinguishes primary from secondary email addresses and phone numbers. The primary phone number is the number that is found on the main page and/or the number that is mentioned most. The primary email address is the address that is located on the contact page and/or the address that is mentioned most. There are considerable gaps in the identification variables of DP (see left panel of Figure 3.7.1), and of the SBR (right panel of Figure 3.7.1). Note that there is some variation in the variable names in the two sources. The consequence of these gaps is that not all URLs that are listed in the DP files can be linked to our SBR.

In 2021 of the ESSnet we studied the quality of the original linkage method between DP-URLs and LUs in the SBR. Next, in 2022, we improved the estimation of the linkage probability. In 2023 we studied the dynamics of the DP data sets which can be used in the long run to improve the number of linked records. Finally, in 2024 we improved the number of DP ULRs that could be linked.

The linkage method is based on agreement between DP and SBR linkage (=identification) variables, (see Table 3.7.1), and for each agreement for a pair of records scoring weights are obtained. Let X_{jk} (with $j = 1, \dots, J$) denote the agreement status between of identification variable j for (pair) k, with value 1 when the values in DP and SBR are identical and 0 otherwise. We denote that the probability that a pair is a link by P_{Lk}. This probability is a function of the linkage score S_k: $P_{Lk} = f(S_k)$. This score is as a function of the agreement vector $\boldsymbol{X}_k = (1, X_{k1}, X_{k2}, .., X_{kJ})'$, and of a vector of scoring weights $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots., \beta_J)$. For the original linkage method, $S_k = \boldsymbol{X}_k \boldsymbol{\beta} = \beta_0 + \beta_1 X_{k1} + \beta_2 X_{k2} + \cdots + \beta_J X_{kJ}$ and $P_{Lk} = 47.5 \ln(S_k) - 234$. Rounded-off this led to the following schema for the linkage probability (in %): 100-300 point (0%+), 400 points (50%), 500-800 points (75%), 850-900 points (85%), 1000 points (95%), 1100 (97%) and ≥ 1200 points (100 %); where the '0%+' indicates the true probability is expected to be slightly larger than 0. These linkage proportions were determined by sampling a minimum of 20 linked units per group and checking whether they were a true match.

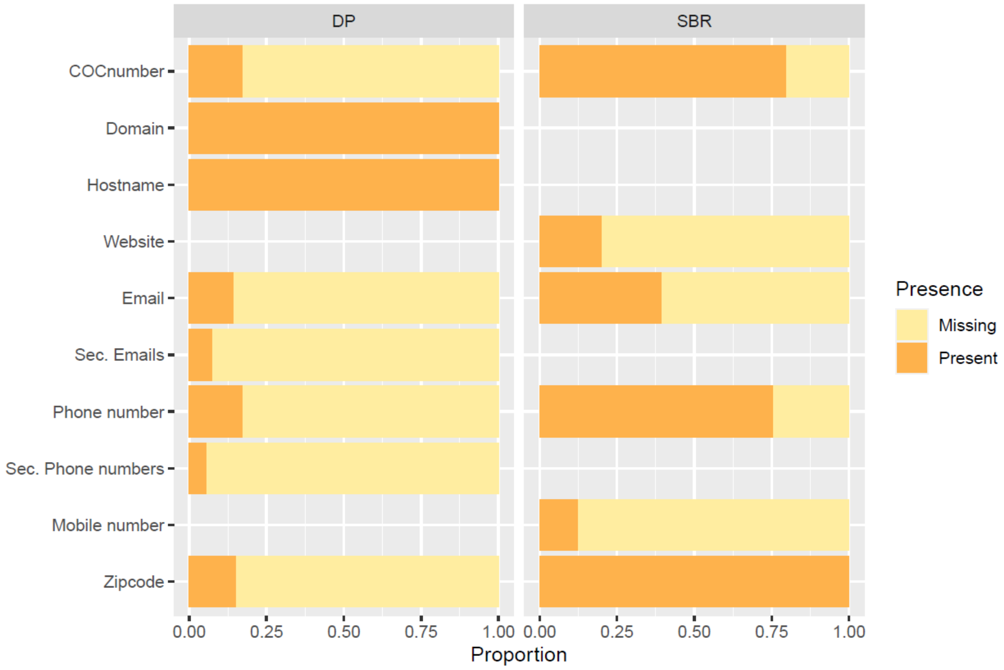

Figure 3.7.1: The proportion of units missing or present in the DP source and in the SBR for the variable in either source that are used for linkage (illustrated for a preliminary delivery of 2018 data).

For a better overview of the number of URLs available at CBS, several counts were made on the data set by DP of October 2020. Note that there are far more *DP-URL LU links* than *DP URLs with a link* and then *LUs with a link*, because the link between a URL and a LU can be one-to-one (1:1), one-to-many (1:n), many-to-one (n:1) and many-to-many (m:n). In total, there were 5 702 455 DP-URLs while there were 4 630 836 LUs (not shown). Unfortunately, 5 057 922 DP-URLs could not be linked to the SBR, because either the identifying information was missing for those DP records, or the identifying information did not agree with those in the SBR. One of the reasons that identifying information does not agree with SBR is that the URL does not belongs to Dutch business (but to a foreign business).

In order to find all possible DP URL-LU links, the cartesian product had to be computed between these 5 702 455 DP-URLs and 4 630 836 LUs. With the regular hardware available at CBS this was not feasible, due to the number of possible combinations that had to be compared. This process could easily have ended up taking over a week to compute, while also applying a constant heavy load to the database server. Hence, the choice was made to leverage the Greenplum database cluster that is available at CBS. By fully converting the linking process to a series of SQL queries and stored procedures, the linking process could be run in its entirety in this environment, reducing the required time to less than one hour. The choice to use Logistic Regression as an algorithm was also partly motivated by this choice, because the components of the logistic function can be computed with relatively simple SQL queries. An additional benefit of using this Greenplum cluster is that SQL tables containing over 1 terabyte of data are still able to be queried relatively quickly, which has proven sufficient for the agreed data retention period of 2.5 years for the DP URL data.

The total number of possible DP URL-LU *links* were 2 634 383 at a linkage probability of larger than 0%. Furthermore, the total number of links decreased to 725 323 at ≥ 75% linkage probability and to 389 979 at 100% linkage probability. When one only accepts links with ≥ 95% linkage probability, then the linked DP-URLs do not provide any *additional* links to those already known from the CoC. At ≥ 75% linkage probability, which is more realistic, 32 081 new links are found by using DP data.

We also counted for how many *LUs* we have found with least one link to a URL, where we distinguished four groups: Group A) LUs with at least one URL from CoC and at least one URL from DP; Group B) LUs with at least one URL from DP but no URL from CoC, Group C) LUs with at least one URL from CoC but no URL from DP; Group D) units for which we do not have any URL. The size of the four groups is shown in Table 3.7.2 in relation to the linkage probability. As mentioned before, the total number of LUs in the SBR is 4 630 836, unfortunately this number also includes LUs that are no longer active (we could not exclude those). The total number LUs with a URL from CoC is 922 578 (group A + C) and that does not depend on the linkage probability. At linkage probabilities of ≥ 0%, 34% of the LUs have one or more URLs (group A + B + C) and 14% of the LUs only have a URL only from DP (group B). At linkage probabilities of ≥ 75%, which is a far more reasonable threshold to use in practice, only 0.6% of the LUs have a URL only from DP (group B). Note that with the original linkage probability computation, LUs that fall in group A and have a high linkage probability (≥ 95%) concern cases where the LU from CoC and DP coincide. That concerns the most reliable URL-LU links that SN has in their SBR.

Table 3.7.2: Size of the four groups within the SBR in relation to DP-URL - LU linkage probability, for the DP data of October 2020.

Groups	DP URL-LU linkage probability						
	> 0%	≥ 10-50%	≥ 65%	≥ 75%	≥ 85	≥ 95%	100%
Total	4 630 836	4 630 836	4 630 836	4 630 836	4 630 836	4 630 836	4 630 836
Group A	700 973	670 528	656 672	644 217	635 936	424 151	389 165
Group B	671 011	213 781	123 765	29 265	1 109	1	1
Group C	221 605	252 050	265 906	278 361	286 642	498 427	533 413
Group D	3 037 247	3 494 477	3 584 493	3 678 993	3 707 149	3 708 257	3 708 257

In 2022, CBS has evaluated the original weights (Table 3.7.1) and the linkage probability function. To that end, four samples were drawn:

- Sample 1: 400 URLs from DP that could not yet be linked to the SBR;
- Sample 2: 400 legal units from SBR, from legal units for which no DP URL was linked;
- Sample 3: 400 legal unit - DP URL links with a linkage probability of at least 50%;
- Sample 4: 400 legal unit - DP URL links with a linkage probability smaller than 50%;

For the first two samples, the experts did not try to find an exact match in SBR or DP. Instead, for sample 1 they checked whether a URL described a company, association or other entity that could exist within the SBR and for sample 2 whether they could find any website for a legal unit for which no match was found. For the last two samples, the experts had to check whether the potential link between the DP URL and the LU was a true match or not.

For sample 1 experts found that 177 of the 400 DP-URLs were websites of businesses that could be in the SBR (but are currently not linked). Furthermore, for sample 2, for 89 of 400 LUs in the SBR the experts could find a website, but for those units the original linkage method does not result in a link with a DP-URL. Results from samples 1 and 2 suggest that there are more true matches possible between DP and SBR than are currently found. This led to the idea to improve the linkage method by including more identifying variables.

Further results from sample 3 showed that when the linkage probability was 50% or higher, most of the potential links are indeed true links (383 of 400 DP-URLs are true links to 381 LUs of 398 LUs). Finally, Sample 4 showed that when the linkage probability was less than 50% only about 11% was a true match (45 of 393 DP-URLs correspond to 44 of 395 LUs)

CBS re-estimated the linkage probability function using logistic regression (LR). The new probability function was given by $logit(P_{Lk}) = S_{Lk}$, with S_{Lk} as defined before; which leads to $P_{Lk} = \frac{1}{1+e^{-S_{Lk}}}$. One benefit of the LR model is that adding additional variables is trivial, whereas the original method would require manually setting the weights and updating the formula to account for the new weights. The macro-F1 scores of both the original method and LR model on a separated test-set (comparing true versus predicted links) were 0.92, indicating a similar level of performance, whilst the LR model will be easier to maintain in the future. In addition, we compared the distribution of the linkage-probabilities of the original versus the LR model, by considering all possible values for $\boldsymbol{X} = (X_1, X_2, .., X_J)'$. Figure 3.7.2 shows that the LR model results higher linkage probabilities, especially when original scores were 400 or higher.

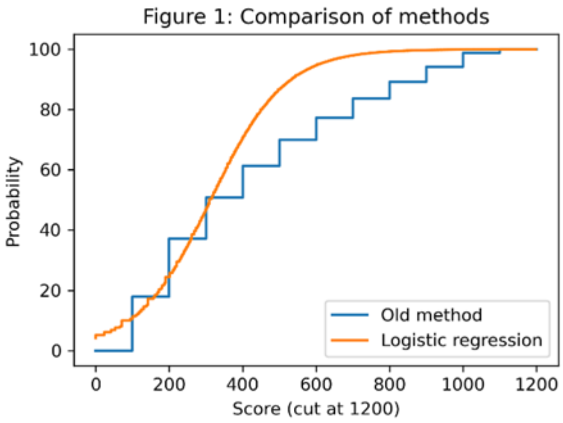

Figure 3.7.2 linkage probability for the original method versus and the LR model as a function of the original score values.

In 2024 CBS included ‘zip code + house number’ and ‘zip code + house number + house number additions’ as two additional agreement variables in the LR model. The updated LR scoring weights are given in Table 3.7.1 (final column) and updated number of linked DP URLs and LUs are given in Table 3.7.3. The lowest linkage probabilities (between 0 and 10%) usually leads to a lot of false links. When looking at linkage probabilities of 10% or more, the updated model resulted in 2.6% more DP URLs that were linked and in 0,6% more LUs with a link.

Table 3.7.3: Number of linked DP URLS and number of linked LUs in the SBR in July 2024 for the LR and updated LR model at two linkage probabilities.

	Linkage prob. 0%+		Linkage prob. 10%+	
	LR model	Updated LR model	LR model	Updated LR model
LUs	1409118	2217982	1198553	1205876
DP URLs	1364248	1487670	1234817	1266323

One issue that we encountered was that the number of URLs delivered by DP varies considerably from month to month. From January 2022 to December 2023 we counted the presence/absence patterns in the DP-URLs to better understand those dynamics and perhaps to improve the linkage method. Over that

24 months period CBS received 16.48 million unique URLs. The large difference between the 5.7 million unique DP-URLs that were delivered October 2020, is because not all 16.48 million URLs are present for each of the 24 months; there are many gaps. To analyse this further, distinguished among continuous versus intermittent patterns. A pattern is continuous when the URL is present for a continuous sequence of months that can start and end at any month, and they are intermittent otherwise. A patterns is intermittent when a URL is present then absent and then present again, for one or more times.

We found 14.45 million (87.7%) continuous patterns and 2.03 million (12.3%) intermittent patterns, see Table 3.7.4. The continuous patterns were further divided into 3.20 million (19.5%) entries, 1.89 million exits (11.5%), 3.45 million stayers for all 24 months (20.9) and 5.91 million (35.9%) temporaries, see Table 3.7.4. Note that the relative size of the patterns in this section that is given between brackets is always given as percentage of the total number of unique URLs. The intermittent patterns were broken down into groups by the number of months the URLs were absent. The top four were one month absent ('[1]') 0.90 million (5.5%), three months absent ('[3]') 0.41 million (2.5%), two months absent (['2']) 0.10 million (0.6%) and six months absent ('[6]') 0.10 million (0.6%). URLs that are one, two or three months absent, irrespective of how many intermittent periods it concerns, sums to 1.47 million (not shown) which comes down to 72.2% of the total number of intermittent patterns. Finally, we counted how many months URLs were present within the 24 months. In decreasing order: 20.9% of the URLs were all 24 months present (the stayers), 16.2% of the URLs were 3 months present, 6.8% of the URLs were one month present, 6.6% were 4 months present and all other groups were smaller than 6.6%.

The question comes up if the intermittent behaviour described above can be understood from either a SBR viewpoint or the way this web data was collected. Also, one could ask it could be expected to appear in other data sources or the same data source in other countries. At this moment we have no similar cases to compare with. However, knowing the stability of a business register and the volatility of (some of the) web data streams, it seems reasonable to assume that the effect is caused by the way DP periodically collects its data. If this is the case, it would be an indication of the necessity to have such important scraping processes under transparent control of the NSI(s).

To conclude, within the ESSnet we showed that the DP data do provide additional URLs to those that CBS already had from the CoC, but still many DP URLs cannot be linked to our SBR. By adding 'zip code + house number' and 'zip code + house number + house number additions' we slightly increased the number of linked URLS. Furthermore, we have improved the original linkage probability function by moving from a linear to a logistic regression function. We also studied the dynamics of DP URLs and found that about 7% of the URLs are missing 1-3 months and then reappear in the DP data set. We can use this finding to further improve our linkage method.

Table 3.7.4: Analysis of URLs in DP from January 2022- December 2023: (Left) presence/absence patterns and (Right) subgroups within the patterns stayers and intermittent.

Pattern type	Number	% of total	Pattern type	# months (present) [absent]	Number	% of total
Total	16485305	100,0	Continuous	(3)	2667974	16,2
				(1)	1123014	6,8
Continuous	14453869	87,7		(4)	1094110	6,6
Entry	3206753	19,5		Rest	8154002	49,4
Exit	1889437	11,5	Intermittent	[1]	901466	5,5
Stayers	3446205	20,9		[3]	412579	2,5

Temporary	5911474	35,9	[2]	104037	0,6
Intermittent	2031436	12,3	Rest	613354	3,72

3.8 URL finding from domain registry at Statistics Finland

Statistics Finland built a fully functional search engine using a web scraper in the beginning of this project. Using business name and address information we were able to retrieve domains and snippets with contact information for businesses. However, due to uncertainty of the quality of the results and snippet data being very messy we also noted the possibility of retrieving domain information from the largest .fi top-level domain (TLD) distributor in Finland, Finnish Transport and Communications Agency (Traficom). After looking into their open API we negotiated access to data not available online for more coverage and precise information on domains and owners.

As there have been multiple new data sources coming in, the need for linking to administrative data in the absence of common unique identifiers have been investigated with various different data sets. In this case we have domains and owner information that may not be the actual user but e.g. a website service catering to clients, or the domain may be used by a daughter business of a corporation with centralized domain administration, or the domain may be used by a business's establishment.

Data processing procedure

The domain register contains all registered .fi domain names and is maintained by Traficom. It contains e.g. the domain name, name of the owner company and business id. The data is publicly available, except for domain names that are owned by private citizens, and it contains over 435 000 domain names. We recently got through with negotiations with Traficom in order to get the privately owned domain data as well as new small companies may have domains registered by the name of the founder. The analysis of the privately owned data is not yet available for this document.

The data processing flow is as depicted in figure 3.8.1. First, all the domains are pinged for http response codes for indication if the domain is in use and following redirections for final domain. The data is then pruned for only domains in active use, and divided to businesses that own only one domain and businesses that own multiple domains, after which they are linked to the business register.

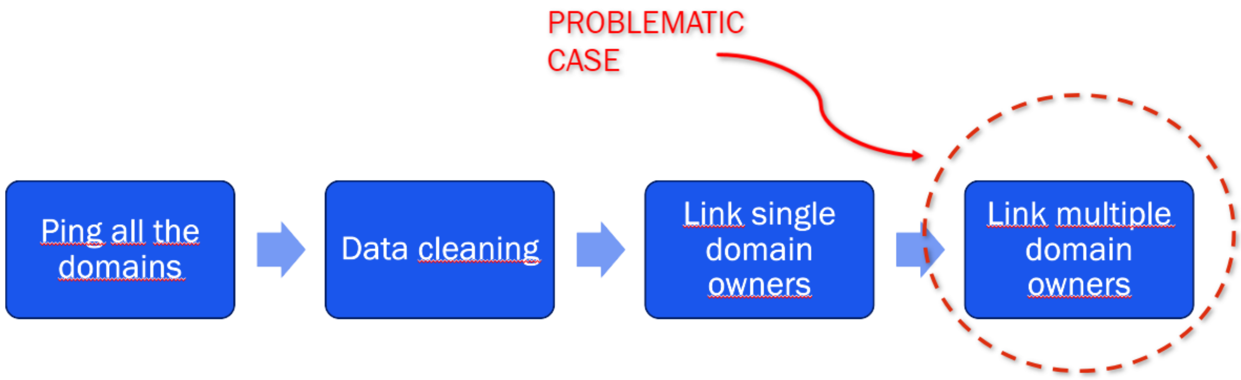

Figure 3.8.1 Statistics Finland domain data process work flow.

We used the R package httr for pinging for response codes and final URL and future and furrr for parallel processing to speed up the pinging. Respond codes tell us whether the domain name is in use. The final

URL where the domain name leads, provides valuable information to be used in the linking process and is the actual website that the enterprise uses. Response code distribution for business owned domains is presented in table 3.8.1.

Table 3.8.1: Response codes for business owned domain data.

Response code	n
Successful responses (200-299)	306 854
Redirection messages (300-399)	18
Client error responses (400-499)	21 566
Server error responses (500-599)	100 878
Other and missing	6 508

Pinged data was then cleaned with a few sequential steps. First, all the domains that are owned by foreign business id were removed. Second, all domains that led to generic “Domain reserved”, Facebook, Instagram or similar websites were removed. Some were found during manual go-through while checking the sample which confirmed there are gaps in the process. Third, all domains not receiving a successful response (200-299) were removed. Fourth, duplicate URL’s by business id were removed. Finally, data was divided into organizations that own only one domain and to those who own multiple domains. This process left about 240 000 rows of data with 140 000 rows of single domain owners and 100 000 rows of multiple domain owners.

Linking the part of single domain owners is straight-forward as the owner is likely also the user of the domain: we have the business id in both the Business Register and the pinged Domain Register. First, we got all legal units of enterprises from the Business Register and joined the corresponding legal units with domains. Next, we got all legal units of associations and proceeded as before. This process resulted in 99.9% of domains finding their match (table 3.8.3). However, we still need to fully verify this result by string distance and sampling. The remaining 134 domains were owned by enterprises that no longer existed or were so new that their registration process was not ready.

Linking multi-domain owners

Problems arise from companies that own multiple URLs. In these cases, we looked for the matches from legal units and establishments that belong to the same enterprise group. The linkage becomes more uncertain as there are no unique identifiers to link observations directly. Lack of auxiliary variables prevent from using more sophisticated probabilistic record linkage methods. Instead, we relied on string distance metrics (R package stringdist) between company names and the final URL.

Linkage was done in three phases (figure 3.8.2). First legal units belonging to an enterprise group were linked, next legal units not belonging to an enterprise group were linked, and finally legal units with no matches from phases 1 or 2 were linked with establishment level data. The linkage was done using fuzzy matching i.e. using string similarity measures between company names and URLs. We score each candidate pair with Jaro-Winkler string distance, set maximum threshold of 0.3 and then choose the best pair as a match within each enterprise group / business id.

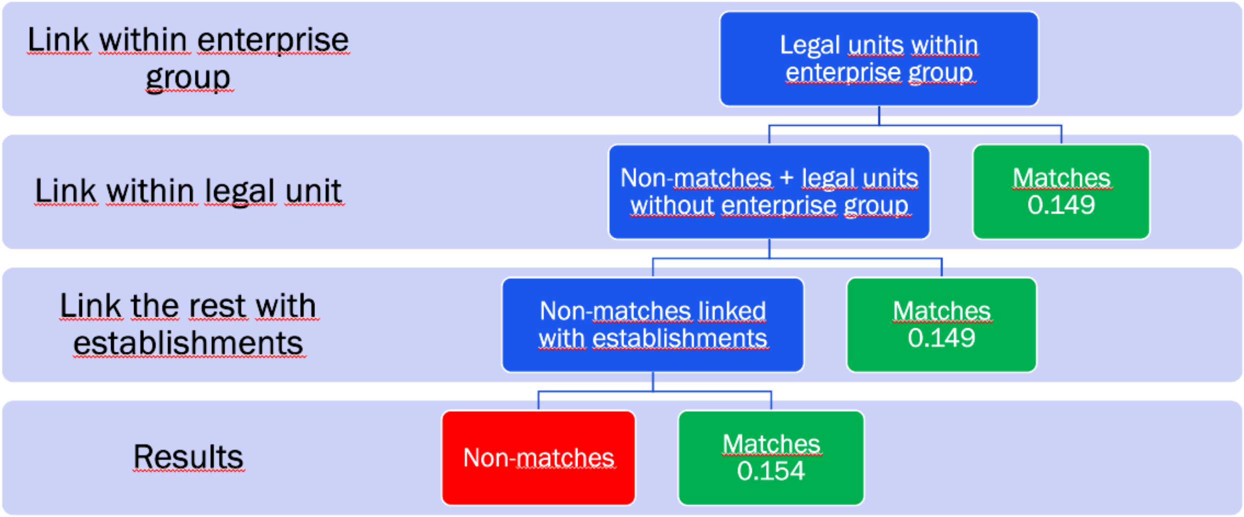
Link within enterprise group
Legal units within enterprise group
Link within legal unit
Non-matches + legal units without enterprise group
Matches 0.149
Link the rest with establishments
Non-matches linked with establishments
Matches 0.149
Results
Non-matches
Matches 0.154

Figure 3.8.1: Linkage steps of domain data.

The accuracy of our linkage was evaluated by manually inspecting a sample (n=600) from the results (table 3.8.2). The sample was a simple random sample and evaluation was done by two persons with 300 domains each. There were no overlapping domains to check as the process was fairly simple. Each person manually checked the domain to see if it was in use and if the correct business actually was using the domain. Enterprise group and legal unit matches were fairly often correct with accuracy results of 82.5 % and 86 % respectively. Establishment level accuracy was poorer with 59 % accuracy. String distance is a fairly coarse method in linkage and especially with domains, their names preferably being very short, linkage with business and establishment names can be difficult. Furthermore, establishment linkage should be done for the whole multi-domain data as there were linkages in business level to establishment level domains.

Table 3.8.2: Accuracy results of manual check of sample of linked multi-domain owners (n=600).

Link type	**Accuracy**
Enterprise group	82.5 %
Legal unit	86 %
Establishment	59 %

The number of matches for multi-domain owners were rather disappointing with overall accuracy of 75.8 % and 17.1 % share of matches (table 3.8.3). More positive was the result of over 155 000 URLs linked to the Business Register. Compared with the already existing URLs in the business register, we obtained 97 449 new ones for legal units and 4 550 were left for establishments. Traficom domain data proves to be a stable new source for getting URLs, especially for single domain owners.

Table 3.8.3: Percentage of domains in Traficom data getting a match in Business Register.

Data	**Percentage share of matches**
Single domain owners	99.9 %
Multi-domain owners	17.1 %
Total	65.5 %

3.9 Wrap-up

The country experiences show a variety of methods to check, enrich or find URLs for SBR units which are variants of the formal URL finding methodologies introduced at the start of this chapter. Austria, Hesse and Sweden use methods based on the URL finding methodology described in Section 3.2. Statistics Netherlands – at this moment – performs URL finding mainly from a third-party data provider using advanced linking strategies which are continuously improved. Statistics Finland explored also some other methods for URL finding such as the use of map data APIs. All in all, we think we may conclude that this field has great potential to be applied on larger scale within the ESS, but also that more work should be done on finding and implementing a common URL finding strategy among the ESS.

4 Enhancing the business register

4.1 Introduction

Once we have found the URLs that belong to the legal units of our interest, the second phase (Figure 2.1, phase B) is to derive statistical variables from the online data. In the statistical business register (SBR) a number of characteristics are typically stored for all „units" in the population: its name, contact information, owner(s) of the businesses, the structure of the statistical unit, its size class, its NACE code and the URL. Within this register a number of different unit types are stored, but the legal unit is typically the unit from which the statistical units enterprise and enterprise group are derived. Here we limit ourselves to legal units.

Variables of interest

We can use the online data to enhance different sorts of information on the population of businesses. For instance one can use the online data to verify and add administrative information of SBR units, such as email addresses and names of the owners of the businesses. Statistics Hesse presents results on that.

Other examples of using online data are to derive characteristics of businesses, currently not in the SBR, such as degree of innovativeness, see Daas et al. (2020) , degree of sustainability, Sozzi (2017), operating a web shop or not (studied in more detail in WorkPackage 2 of this WIN project), or belonging to the platform economy, see Daas et al.(2023)

Furthermore, a common use-case for using online data to improve the SBR is to predict NACE codes, Gussenbauer et al. (2022) Prediction of NACE codes is typically intended to support (rather than replace) manual editing at NSI's which currently takes a considerable effort. When predicted NACE code differs from the registered one, the latter may be incorrect. In that case, the editor can use the prediction as a suggestion which NACE code may be the correct one. However, the editor is in charge of the final assignment.

Predicting NACE codes typically requires interpretation of raw texts, natural language processing (NLP) and machine learning techniques. A literature study executed at the start of this project identified fifteen papers on predicting economic activity using text mining and one overview paper on feature engineering in text mining. The papers vary in the type of input sources used, in the actual economic activity classification they use, the type of features used and the models that are applied. The literature study was the starting point for the two studies presented here, which look into many of the open questions not solved in earlier studies. More information on the literature study itself can be found in Section 5.1.

Getting the right data

Because enterprise websites can vary significantly in almost all aspects, scraping is typically done using a generic scraping approach, which, unlike specific scraping, does not require prior knowledge of the site's structure. Generic scraping typically begins at the home page and recursively visits deeper pages up to a certain maximum depth. Decisions must be made about whether to store the entire website, only the text, or the derived data, or all of the above. It might be valuable to use a focused scraper. Such scraper does not follow all links but prioritizes those that are expected to contain the most valuable information for the task at hand. For example, a focused scraper might prioritize the "about us" page to identify economic activity. Both approaches have their pros and cons: a generic scraper is simpler and thus requires less maintenance, but will result in larger data volumes where a focused scraper could have better results but may needs more configuration.

Although estimating the linkage probability between URL and the legal unit has been treated as part of the URL finding already, here we underline that it is important to verify that the website visited matches the legal unit at hand using identifying information on the site, such as a chamber of commerce or tax identification number, if possible. In countries that have national legislation that requires enterprises to include this information on their websites this is straightforward. Within this project Statistics Hesse and Statistics Austria make use of such legislation. When linkages between URL and legal unit are less certain, one can either drop those cases from the model or give them smaller weights. Finally, as with the URL finding phase, special attention should be paid to many-to-many relationships between legal units and websites as online information may not be uniquely representative for the legal unit at hand.

4.2 Common elements on enhancing the NACE codes

Enhancing the business register with respect to NACE characteristic comprises (at least) four objectives:

1. **NACE prediction**: Prediction of one or more NACE codes of existing units using at least one open/online source (*)
2. **NACE determination**: Projection of one or more NACE codes of new-born units (subscribing their legal unit to a government body)
3. **NACE misclassification detection**: Predict whether a registered NACE code is correct or not
4. **NACE transitions**: Predict new NACE codes starting from current codes, when the NACE code classification system changes (in some years)

In this project we decided to focus mainly on NACE prediction, with the long term aim to support manual editors at statistical offices. In addition, since accurate NACE prediction of all 5-digit codes is quite challenging we also worked on NACE misclassification detection. Here the ambition is not to accurately predict all NACE codes but (only) to just model whether the registered NACE code is correct or not. That prioritizes the units that manual editors have to look at. Units that are likely to be misclassified can be assigned a prediction of the top-three of five most likely NACE codes following from NACE prediction.

We agreed to focus on the NACE code of *legal units* as the unit type and to focus on the main economic activity rather than (also on) secondary activities. In case of predicting the full range of NACE codes, we aimed at results at 2-digit level or even more detailed levels when the method is applied to a limited number of NACE codes. Different project partners focused on different elements on the NACE prediction. CBS specifically focused on the use of knowledge based words, Statistics Austria looked into features selection and into use of NACE hierarchy to improve predictions.

Furthermore, we considered three ways for scraping the texts:

1. **Focused scraping**. Scraping of the landing page and specific pages that are expected to contain information about economic activity. Such a scraper looks for substrings in the link description such as “home”, “welcome”, “about us”, “company” and “services”.
2. **Generic scraping**. Scraping of the landing page plus all subpages up to a maximum number of pages. It needs to be reasonably large, at least 25 or 30, to increase the chance to collect useful data for the classification.
3. **Combined focused and generic scraping.** Scraping of the landing page plus a number of specific links and adding additional pages until a maximum number of pages has been reached. This maximum number can be smaller than in the case of generic scraping

The included projects primarily use focused scraping (Statistics Austria) and combined focused and generic scraping (Statistics Netherlands).

We have also worked out ways to obtain a good training and test set. A first challenge is to deal with the fact that there are errors in the actual NACE code labels in the SBR, but we wish to obtain a suitable train and test set for NACE prediction. We considered the following options:

1. **Manual editing**. Create a "gold set" by manually checking the NACE codes
2. **Record selection**. Make a selection of available NACE codes such that the selected codes have a lower probability of being incorrect. Such a selection can be done in different ways.
3. **Use survey data**. Send out a survey or make use of an existing survey to collect information on the NACE code.
4. **Accept NACE errors**. Accept that there are errors in the NACE codes and use a machine learning model which is robust for NACE errors.

This project primarily used ‘records selection’ (Statistics Netherlands) and ‘survey data’ (Statistics Austria) to obtain train and test data. For the evaluation measures our starting point was the use of standard machine learning evaluation measures: accuracy, F1 score and recall. The actual measure(s) depended on the specific conditions of the study. For the study by Statistics Austria specific measures were developed that accounted for the hierarchical structure of the NACE classification. Furthermore, a top-5 accuracy was used which means that if one of the first give predicted NACE codes per unit (the top5 most likely NACE codes) was correct then the prediction is considered to be correct. For the F1 score, besides the micro-F1 score (each record counts for itself) averaging was also used in the form of a macro-average (F1 score per code and each code has equal weight) and weighted average (F1 score per code and the weight per code each code equals the size in the test set).

4.3 NACE misclassification detection at Statistics Netherlands

Statistics Netherlands uses texts of business websites for enhancement of the NACE codes in the Statistical Business Register (SBR). Within the ESSnet we followed two paths. First, we worked on a system to predict for which of the units in the SBR the registered NACE code is likely to be incorrect: misclassification detection. This work is a follow up of the work by Oosterveen et al., 2021. The second path is to test which features lead to the best performance for predicting the NACE codes. The first path is described in the current section, the second path in section 4.4.

The starting point is a selected set of (registered) NACE codes $\mathcal{G}$ that we use to find NACE codes. That means that we do not aim to detect misclassifications in the whole set of 5-digit NACE codes (> 900 codes), but we take some sensible selection. For legal units i in the SBR, we are interested to predict a latent variable z_i which equals 1 when its registered NACE code $\hat{y}_i$ is incorrect and 0 when it is correct. We combine two models to predict this latent variable. The first model is used to estimate the probability that true NACE code of unit i is g, for all codes within $\mathcal{G}$, using its website text and a text mining model. The second model is used to estimate the probability (propensity) that unit i is misclassified, denoted by π_i. The specific form of this second model depends on the scenario. In scenario 1 we estimate a single parameter π_i for the whole population. For scenario 2, π_i depends on the crossing of the background variables ‘size class’ (4 classes) and ‘number of legal units’ (2 classes). In an earlier study we found that the propensity to be misclassified is relatively small for small and simple units (enterprises consisting of one legal unit) and increases with larger and more complex units, except that the most influential units hardly have any misclassifications since these are already regularly checked manually. These two models are combined into a mixture model that predicts the misclassification probability $P(z_i = 1)$, also denoted as τ_i, which is explained in section 2.1 of Oosterveen et al. (2021).

The parameters are estimated with an expectation-maximization algorithm (EM) using maximum likelihood. The idea is that a gold set (either 20, 50 or 100 units per NACE code) is used to find starting values for the parameters of the two models (M-step). Next, given the model parameters, the website texts and the background information for all units τ_i is derived (E-step). Finally, the model parameters are re-estimated (M-step) given the τ_i values. These τ_i values are used as target variable for the first model and are used as weights for the second model. Finally, if a τ_i is larger than a threshold, the unit is considered to be misclassified and otherwise not. This threshold is chosen so that the percentage of misclassified units corresponds with the different estimated π_i values.

We have tested this model on a set of 25 5-digit NACE codes of approximately 45 thousand legal units, as described in Oosterveen et al. (2021). Some of these codes are easily interchanged (hair dresser and beauty salon) others are further apart (photography). We first created a set that was nearly free of errors and then purposely introduced different levels and kinds of misclassifications. As features we used a combination of two TF-IDF matrices: the first matrix contained all features (but after data processing such as removal of stop words) and the second matrix contained all knowledge based words. These are referred to as D-words. We used these features because this gave good results in two tests on feature selection that we describe separately in section 4.4.

In 2022 we have expanded the method by Oosterveen et al. (2021) so that we could plug in different machine learning classifiers (the second model). To that end we developed a generalized version of the EM algorithm, since the ML classifiers are not necessarily estimated by maximum likelihood. Using a gold set of 100 units per NACE code, we tested the performance of different classifiers (Naïve Bayes, Support Vector Machine, Random Forest, Gradient Boosting) and for each of them different hyper parameters by using a grid search and cross-validation approach. For each of these models we selected the best hyper parameters and used those in the GEM algorithm to find misclassifications. Surprisingly, the best performing model was a Naïve Bayes model. The reason was that the Naïve Bayes model was the most robust against the presence of misclassifications within the set.

Furthermore, in 2022 we have worked on calibrating the confidences given by the Naïve Bayes model, in order to accurately estimate the probabilities for all codes g within $\mathcal{G}$. These confidences are calibrated on the gold set, including some weights. We found that either using the inverse of the inclusion probabilities or the use of post-stratification weights yielded more accurate results than giving each unit the same weight.

Finally, we computed the impact of the size of misclassifications for a gold set of 100 for the psi model. The error pattern in the simulated data was based on the error pattern that occurs in the SBR. Also, we tested two scenarios, without (scenario 1) and with (scenario 2) the use of background variables. We repeated the procedure (generating misclassifications) 10 times to also estimate standard errors, see Figure 4.3.1.

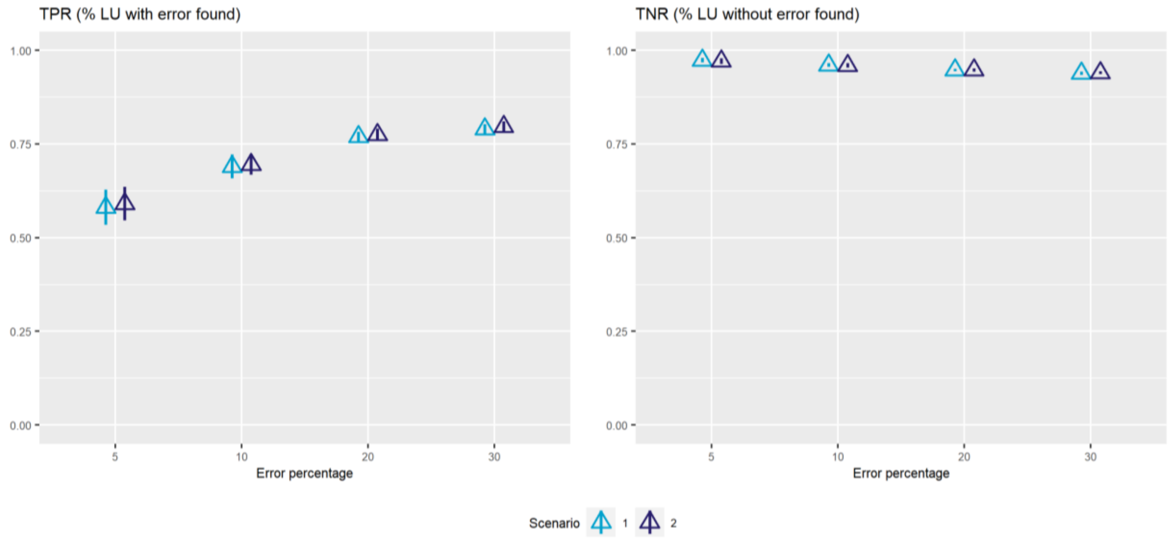

Figure 4.3.1: True Positive Rate (TPR: fraction of legal units with an error that is classified as erroneous) and True Negative Rate (TNR: fraction of legal units without an error that is classified as correct)

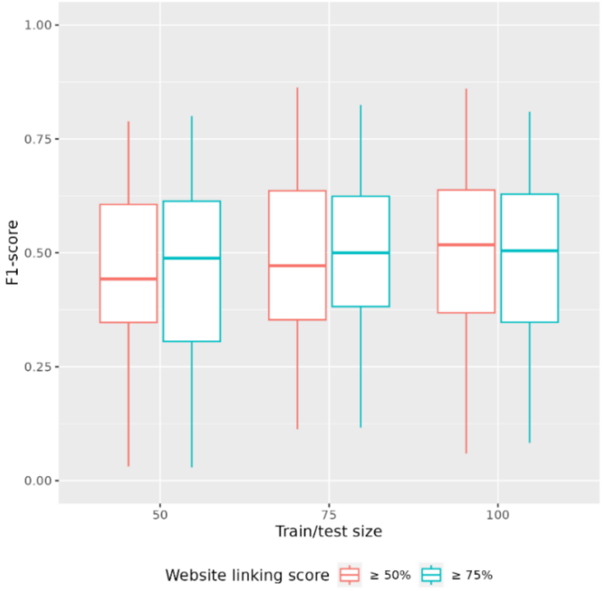

Figure 4.3.2. Boxplot of the F1 scores over 41 5-digit codes as a function of the DP URL- LU linkage probability and as a function of the train/test size.

Figure 4.3.1 shows that the higher the misclassification rate (error percentage) in the data, the higher the TPR becomes. Also, the error bars (2 times the standard error around the mean) decrease with an increased misclassification rate. In other words, if there are less misclassifications in the data, the model has more difficulty in finding them. In general, the TPR shows that the model is able to find 58 to 80 percent of the units with erroneous codes. Besides, the model can easily find the legal units without an error, which is shown by the TNR. When the percentage of misclassifications increases, the TNR slightly decreases. For both the TPR and the TNR, we see hardly any difference by adding background variables to the model (scenario 2 versus scenario 1).

In 2023 and 2024 we worked on applying the method to a real situation of observed NACE codes in section R (Art, Entertainment and Recreation). In this section, 16050 legal units were manually checked and corrected at CBS Netherlands. We first selected 64 5-digit codes in section R and 18 5-digit NACE codes outside section R. The true codes of units with a registered code within section R sometimes lies outside section R. We selected the codes outside section R that occurred 20 times or more as the true codes outside section R; together it accounted for 86% of all units with a true codes outside section R. We refer to misclassifications among these 64 codes as subtle errors, as they occur relatively often. All other errors are referred to as obvious errors. We made some further selections, one of which was that the population size per code should be ≥ 110, see the description of data set 2 in section 4.4. After the selection we ended up with 41 5-digit codes: 23 codes within section R and 18 codes outside section R.

We first explored how well we could predict these 41 5-digit NACE codes. To that end we randomly selected 100 LUs within each observed NACE code (that could contain errors) and used a train/test split where repeatedly one unit was left out to predict (like in cross-validation) until all 100 units per code were predicted. The median F1-score over the 41 5-digit NACE codes was close to 0.5, see Figure 4.3.2. The F1-score slightly increased with size of the train/test set from 50 to 100. For the train/test set size of 50 and 75 the median F1-score was slightly higher for linkage probability ≥ 75% as compared to ≥ 50%. For a train/test size of 100 the median F1-scores were nearly equal for both linkage probabilities but the range over the 41 5-digit NACE codes was smaller for linkage probability ≥ 75% as compared to ≥ 50%. When we looked further into the results, we found that some NACE codes were very difficult to predict while the prediction of other codes went very well, see Table 4.3.1. This difference is mainly due to the information on the websites. For instance, website texts of museums are usually very clear so museums were easy to predict wile website texts of artists tend to be not as informative.

Table 4.3.1: NACE codes with lowest and highest top 5 F1 scores, for train/test set of 100 and linkage probability ≥ 75%.

Top 5	NACE	Description	Size	F1
low scores	90012	Production of live theatrical presentations, concerts and opera or dance productions and other stage productions	522	0.08
	93299	Other recreation (no marina)	3602	0.17
	90020	Services for performing arts	5102	0.20
	90030	Writing and other creative arts	18895	0.21
	90011	Performance of stage art	10291	0.22
high scores	93291	Marinas	251	0.81
	91021	Museums	280	0.78
	86912	Practice of physiotherapists	5206	0.77
	96022	Beauty care, pedicures and manicures	13903	0.74
	93121	Field football	135	0.73

In 2023, we first tried to improve these F1 scores by adding new features from administrative data sets. We used legal form, complexity of unit (available in the SBR); we used relative distribution of quarterly turnover within a year as well as turnover per employee (available in Value Added Tax declarations combined with survey data); we used a building use function (available in a Dutch registry on buildings) and we used categories of establishments and textual descriptions of establishments (available in DataLand a data set constructed by Dutch municipalities with purpose to register the values of buildings). The best in terms of model performance was the ‘textual descriptions of establishments’. However, when we added these texts to the additional website texts model performance was lower than without these texts. Also, when only using these text the performance was lower compared to the use of the website texts. Therefore, we decided not to use them.

Instead, we improved the processing of the texts, for instance we removed URLs within the texts and we dropped a number of URLs that were highly likely to be written in English rather than in Dutch (using LangDetect). Furthermore, we decided to drop codes that were too hard to predict, e.g. 90012, 93299 and we combined a few codes that the model could not distinguish. In the final selection we had 24 5-digit NACE codes: 11 outside and 13 inside section R. Now we achieved an average F1 score of 0.735, which was a clear improvement compared to the results of Figure 4.3.2.

In 2024, we tested how well we could detect misclassification in the 11 NACE codes in section R using the GEM algorithm, which was run on 24 5-digit NACE codes. We limited ourselves to misclassified units of which the true code lied within the 24 5-digit NACE codes: the subtle errors. Remember that we have manually checked units from which we know the true codes. Those manually checked units were used to evaluate the model outcomes. The results can be found in Table 4.3.2.

The first setting that we tried, denoted by “all data”, concerned the population of all LUs with URLs from which we scraped website texts for those 24 codes. That resulted in a TPR of 0.284 and a TNR of 0.703 (see Table 4.3.2) The true proportion of errors was 0.19. Based on the earlier results (Figure 4.3.1) we had expected higher TPR and TNR values. The issue with the “all data” setting was that the population size of the 24 codes varied enormously and ML models are known to be sensitive to data imbalance. We therefore decided to limit the maximum population size to 1000 (setting “max 1000”). With that setting we obtained much better results: a TPR of 0.459 and a TNR of 0.941. Finally, we further improved the balancedness of the data by adding units to codes with small population sized. We added all units that were available for which we had textual activity description form the CoC (setting “max 1000 supl”). Unfortunately, the results were slightly less good, probably because the CoC text are usually very short and they were less suitable for NACE prediction.

Table 4.3.2: Test to detect misclassifications in 11 NACE codes in section R for three settings, limited to the subtle errors.

	Full set	Test set			
setting	Estimated prop. errors	True prop. errors	Estimated prop. errors	TPR	TNR
All data	0.064	0.190	0.294	0.284	0.703
Max 1000	0.067	0.190	0.143	0.459	0.941
Max 1000 supl	0.065	0.199	0.138	0.433	0.935

In conclusion, we developed a method to predict the probability that a registered NACE code of a legal unit in the SBR is incorrect, using website texts and background variable of legal units. The method, tested on a real case-based simulated data is promising since we found 58-80 per cent of the erroneous codes while we hardly unjustly claimed units to be misclassified that were in fact correct. We have also applied the method to a real case scenario. Unfortunately, in this section it was challenging to achieve good NACE predictions because website texts do not give enough information to predict some activities while other NACE codes are very hard to distinguish based on only the website information. For a selection of NACE codes we could find up to 45 per cent of the erroneous codes (subtle errors) while 94 per cent of the units that were correct were also identified as being correct. In future we plan to also test the obvious errors (errors with true code outside the now included 24 codes). We also want to test a number of potential improvements, some of which concern improvement of the initial estimate that are used to start the GEM algorithm.

4.4 NACE prediction at Statistics Netherlands

One of the important steps when using text mining for classifiers is the choice of the features. We were interested to test whether it improves the model performance when the selected features are really content-wise related to the classes that they predict, rather than just entering all features (after some standardization steps). Content-wise related features may help in official statistics to trust the results of the text mining outcome, and to avoid spurious correlations.

Table 4.4.1: Different features sets that were compared for NACE prediction.

Feature set	Description
all	This benchmark uses the top x most predictive words. The value of x is mentioned in Table 1 and 2.
YAKE	yet another keyword extractor (Campos et al., 2020): this is a second benchmark where keywords are extracted from a text, this keywords selection does not depend on the classes to be predicted. The keywords are extracts from the original text, they do not have to in the top x words.
D-words	descriptive words: these are knowledge-based words specific to NACE codes that manual editors currently use at CBS when they check business' activity descriptions.
C-words	concept words: these are abstractions of D-words. For instance the D-words station wagon, vehicle, four-wheel drive are examples of the C-word car. First D-words are selected in the text and next they are replaced by their corresponding C-words. In the past, CBS made knowledge graph that contained both C- and D-words for each of the NACE classes.
IGFSS	improved global feature selection scheme (Uysal, 2016): a method to select words associated with a class to be predicted. We used approximately 10 words per class that were most related to a class. In practice nearly all words were positively related to a class.
IGFSS+	As IGFSS but with additional cleaning of the data set before applying the feature selection scheme by Uysal.
All + { other}	Create two separate 'term frequency inverse document frequency' (tf-idf) matrices for the two feature sets and combining these feature sets, as was done in Li et al. (2019).

In this study we compared a large number of feature selection methods, see Table 4.4.1. The aim of this study was to compare the performance of predicting NACE codes of content-related features as opposed to more neutral feature sets, where the features were used in the form an TF-IDF matrix.

We used two different data sets. The general approach for both data sets was that we linked URLs from DataProvider(DP) to LUs in the SBR. Next, websites were scraped. If URLs were not reached or if they were uninformative (" this domain is unavailable"), they were dropped. Next, language detection software was used and only Dutch texts were kept. A small number of URLs occurred more than once for different legal units. In that case, the URLs linked to a legal unit with the highest number of employees were kept. Words were extracted from scraped content, lowercased, lemmatised and stop words were removed. We randomly split the data set into an 80% train set and 20% test set (6 472 URLs). We used a Naive Bayes model and a Support Vector Machine (SVM) whose hyper parameters were tuned using cross-validation on the train set.

Dataset 1. We started with about 400 000 LUs belonging a set of 109 related 5-digit NACE codes (see Toledo (2021), on tourism and recreation, expanded with the car industry (since some NACE codes in tourism and recreation are easily confused with the car industry). CBS originally linked 35 733 URLs from DP. The cleaned data set contained 24 893 URLs of 106 5-digit NACE codes, corresponding to 54 4-digit NACE codes. We ended with 52 4-digit NACE codes, because 4-digit NACE codes with less than 25 cases were dropped. We used the registered NACE codes for our test and we are aware that part of those codes will be incorrect, but we expect that this does not really disturb the test of the different feature datasets.

Set 2. This data set concerned 64 5-digit NACE codes in section R on tourism and recreation. Within this data set, 16 050 LUs were manually checked on their NACE code and corrected if needed. We added 18 codes outside section R that are often the true codes of units that are misclassified in section R. Next, we linked DP-URLs to all LUs (not only the checked ones) of 64 + 18 5-digit codes , scraped and cleaned the content. We first tested how well we could predict the NACE codes and selected NACE codes with at least 110 units. The reason behind this minimum of 110 was that we wanted to use 100 units per code in our gold set of the GEM algorithm predicting misclassifications and then 10 units are left over to predict in the noisy set (see section 4.3). We ended up with 41 codes of sufficient size: 23 within R and 18 outside R. Thereafter, we dropped some NACE codes that we could not predict accurately (see Table 4.3.1 in section 4.3) and we merged some NACE codes that the model could not distinguish. We then ended up with a final set of 24 5-digit NACE codes: 11 outside and 13 inside section R. Then, we selected 100 units for each of those 24 codes for which it is likely that the NACE code is correct, by using a combination of up to 10 manually checked units and the remainder are codes for which three ML models predicted a code that agreed with the registered one.

Table 4.4.2: Data set 1. F1 scores on test set predicted as averaged over 52 4-digit NACE codes, for different feature sets (see Table 1)

Algorithm				D-	C-			all+D-	all+C-	
		all	YAKE	words	words	IGFSS	all+YAKE	words	words	all+IGFSS
	number features	5000	2687	1223	488	502	7687	6223	5488	5502
SVM	macro avg	0.54	0.46	0.53	0.51	0.48	0.53	0.57	0.57	0.56
	weighted avg	0.70	0.65	0.69	0.69	0.65	0.70	0.72	0.72	0.72
	st.dev	0.218	0.236	0.213	0.228	0.237	0.225	0.215	0.211	0.216
NB	macro avg	0.25	0.24	0.30	0.32	0.35	0.32	0.35	0.35	0.37
	weighted avg	0.58	0.55	0.60	0.61	0.59	0.62	0.65	0.64	0.64
	st.dev	0.314	0.298	0.319	0.310	0.299	0.299	0.311	0.309	0.305

Table 4.4.3: Data set 2. F1 scores on test set predicted as averaged over 24 5-digit NACE codes, for different feature sets (see Table 1)

Algorithm		all		D-	IGFSS	all+YAKE	all + D-	all +	all +
			YAKE	words			words	IGFSS	IGFSS+
	number features	500	210	500	237	710	1000	737	742
SVM	macro avg	0,756	0.646	0.861	0.730	0.752	0.861	0.802	0.807
	weighted avg	0.755	0.649	0.862	0.732	0.753	0.862	0.802	0.806
	st.dev								
NB	macro avg	0.757	0.605	0.829	0.716	0.746	0.836	0.788	0.808
	weighted avg	0.756	0.608	0.830	0.716	0.745	0.837	0.786	0.807
	st.dev								

The results of both data sets were in line with each other, see Tables 4.4.2 and 4.4.3. First of all we found that for a given feature setting, performance of the SVM model was better than for NB. Further, we found that the ordering of the feature sets according to their performance were very similar for the weighted and the macro average. Note that for data set 2 the results of the macro and weighted average are nearly identical because nearly all classes consisted of 100 LUs.

Furthermore, we found that the more the words were related to the content of the classes the better the scores: D-words performed better than IGFSS, which performed better than YAKE. The performance of the setting “all words” was close to that of the D-words. When two TF-IDF matrixes were combined, the performance was often better than the performance of the individual matrices: all + D-words had the overall best performance. We decided therefore to use the features with two TF-IDF matrices, all + D-words, in the GEM algorithm, in section 4.3.

4.5 NACE prediction at Statistics Austria

Statistics Austria investigated classifying the NACE code of enterprises in the Statistical Business Register (SBR) using text data collected from enterprise websites. The results presented in the sub-chapter focus on predicting 2 digits NACE code of legal units from the SBR.

Data acquisition and pre-processing

The data collected for this classification task was mostly collected for enterprises which are part of the sampling population of the annual survey on the usage of information and communication technologies (ICT) in Austrian enterprises for the years 2019 until 2021. Additionally, for the survey year 2021 websites of enterprises with number of employed persons ranging from 5 to 9 were also included. Statistics Austria used their internal URL linking procedure to link websites to legal units from the SBR and in order to scrape the text from these websites. Part of this linking procedure contains directly linking a website to a legal unit.

For training and testing classification models only deterministically linked websites where used resulting in scraped text from over 128 000 website-enterprise pairs (e_i, u_i) with about 88600 websites linked to roughly 62500 enterprises.

Before applying a classification model, the text gathered from the website is pre-processed:

- Transform each word with the German morphological lexicon available on https://www.openthesaurus.de/about/download\footnote{Accessed 02.09.2024}.
- Remove all digits and punctuations.
- Remove characters not part of the German dictionary.
- Remove German stop words.

For this purpose a “word” is defined as a consecutive sequence of characters containing no spaces.

Feature selection

With the pre-processing steps applied the collected text still contains over two million different words, raising the need for a feature selection method. Initial tests using general descriptions and example for NACE codes showed that roughly halve of the words in these descriptions do not show up in the text collected from websites. During this work a more data-driven feature selection method was tested, see Uysal (2016). This strategy combines both a global and local feature selection score to create a balanced set of features used for classification models. For global feature selection score, the (GI), (DFS), and (IG), were tested. For the local feature selection score, the (OR) was used. The selection strategy was applied once to all available data before applying a train and test split. Selecting features after each train and test split would be methodologically more sound. The computational intensity of this selection method is however quite high and, performing this step once was a convenient choice for testing this method.

Up to 200 and 500 words, denoted in the following sub-sections as W_{200} and W_{500}, for each 2-digit NACE code were selected with this method.

Classifier

For the classification a Neural Network model and the XGBoost algorithm were tested. The Neural Network was applied using the R package Keras and the TensorFlow software, see Allaire and Chollet (2019) and Abadi et al. (2015). The XGBoost algorithm, see Chen and Guestrin (2016), was applied through the R package xgboost, see Chen et al. (2023).

The motivation for testing a Neural Network model was based on the efficient implementation of modern Neural Network software, which is designed to handle thousands of features. Additionally, the use of pre-trained word embeddings could potentially improve prediction quality without requiring large amounts of data.

For the Neural Network two different architectures were used, shown in Figures 4.5.1 and 4.5.2. The first architecture uses one-hot encoded words as input, weighted by the term frequency-inverse document frequency (TF-IDF) transformation, and consists of feed-forward layers. The feature set used contained W_{200}. The second model specification builds on the first and utilises additional input features found within W_{500}. These features are transformed using pre-trained word embeddings, followed by multiple convolutional filters. The outputs from the feed-forward and convolutional layers are concatenated in a penultimate layer and then fed into a final softmax layer. This model used pre-trained embeddings from fastText, see Joulin et al. (2016). These embeddings are trained on Wikipedia and Common Crawl, an open repository of web-crawled data. Prior to applying these word embeddings, the dimensionality reduction algorithm proposed by Raunak (2017) was applied to reduce the dimensionality from 300 to 50. This was done primarily to reduce training time, and initial tests indicated that the reduced dimensionality resulted in minimal performance loss regarding prediction quality.

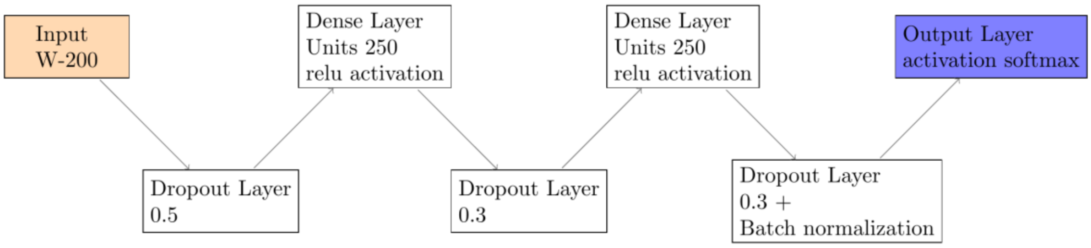

Figure 4.5.1: Neural network architecture wide

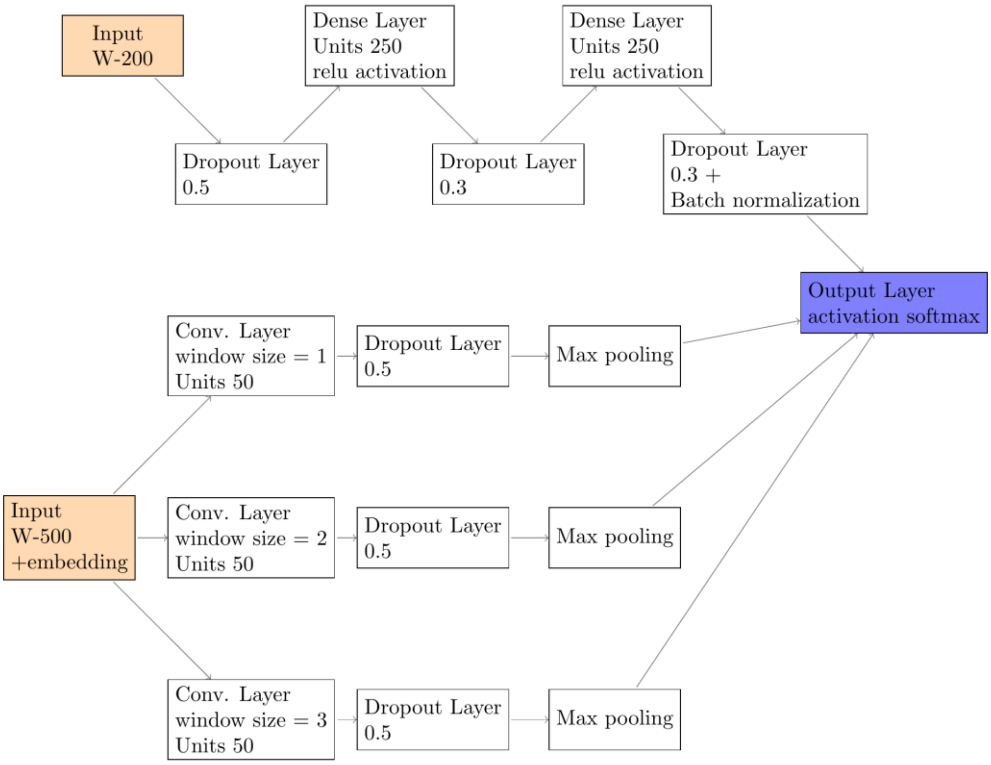

Figure 4.5.2: Neural network architecture wide and deep

Figures 4.5.1 and 4.5.2 show the choice of hyperparameters, which were determined with an additional tuning step.

Applying the XGBoost algorithm is more straightforward than the use of Neural Networks. In addition, XGBoost can handle sparse matrices, which significantly reduces memory consumption during model training. Table 4.5.1 shows the hyperparameters chosen for the XGBoost algorithm. These parameters were not derived using hyperparameter tuning but were instead taken from another classification task where they proved to be useful.

Table 4.5.1: Hyperparameters chosen for the XGBoost algorithm

Parameter	**Value**
nrounds	1000
max_depth	7
eta	0.01
subsample	0.5
colsample_bytree	1
eval_metric	mlogloss
objective	multi:softprob

Results

For evaluating the models 5-fold cross validation with a 80-20 split was applied four times, totaling a number of 20 prediction runs. In addition, two additional settings were tested. The first uses the prediction on the 1st NACE level as predictor and the second aims to limit the number of noise by selecting text only from certain sub-pages on a website.

Using predictions on a higher level hierarchy

Respecting the hierarchical structure of the NACE-code may potentially improve the prediction quality of the model. To follow up on this idea a model was trained to predict for each enterprise and URL pair (e_i, u_i) the NACE level 1 code. The model used for the prediction was the Neural Network model shown in Figure 4.5.2 with a total of 500 words per NACE level 1 code, using the feature selection by Uysal (2016). NACE level 1 predictions where generated for each enterprise and URL pair (e_i, u_i) 4 times using cross validation and the average of the resulting predicted probability scores were used as additional inputs for predicting the NACE level 2 codes. For the following results this scenario will be denoted as “Hierarchy”.

Selecting certain subpages of a website

As the data collected might hold a lot of noise, it could be beneficial to pre-select certain parts of the available data prior to the feature selection. For this setting only text from the landing page and certain sub-pages was used for classifying the NACE code. The sub-pages were defined by having one of the following words either in its URL or in the text between the hyperreference tags on the landing page:

```
enterprise, company, unternehmen, home, welcome, ueber, über uns, über, geschichte,
about us, uber uns, about, unsere, willkommen, produkt, product, artikel, article,
organisation, dienstleistung, angebot, leistung, offer
```

For the following results this scenario will be denoted as “SelectedLinks”.

Figure 4.5.3 displays the distribution of accuracy, F1-Score, and top-5 accuracy (horizontal panels) for each method (colours) across all cross-validation runs (x-axis). Different data inputs are shows on the y-axis. The methods tested include the Neural Network (NNet), the XGBoost model (XGBoost), the use of predicted NACE level 1 probability scores (Hierarchy), and using only part of the available text data (Selected Links). The results indicate no significant differences between the feature selection scores DFS, Gini and Information Gain (vertical panels). Furthermore, the use of word embeddings and convolutional filters showed hardly any improvements despite utilizing potentially much more information compared to using only a one-hot encoding matrix. Including NACE level 1 codes as model input (Hierarchy) improves the F1, accuracy, and top-5 accuracy measures. The results for XGBoost are similar to those achieved with the Neural Network, and selecting only a limited amount of text data does not enhance prediction performance. For top-5 accuracy, there even appears to be a negative effect.

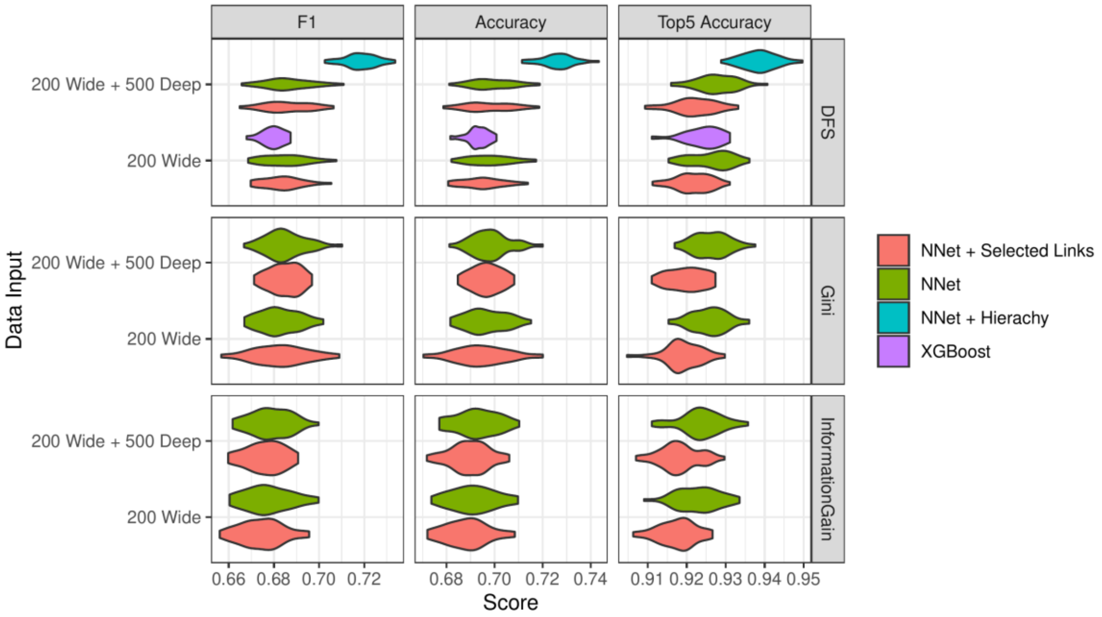

Figure 4.5.3: Overall results using the Neural Network models (NNet), the XGBoost model (XGBoost), using predicted NACE level 1 codes (Hierarchy) and only a subset of available text data (Selected Links). The vertical panels indicate different performance measures and the horizontal panels show different feature selection scores.

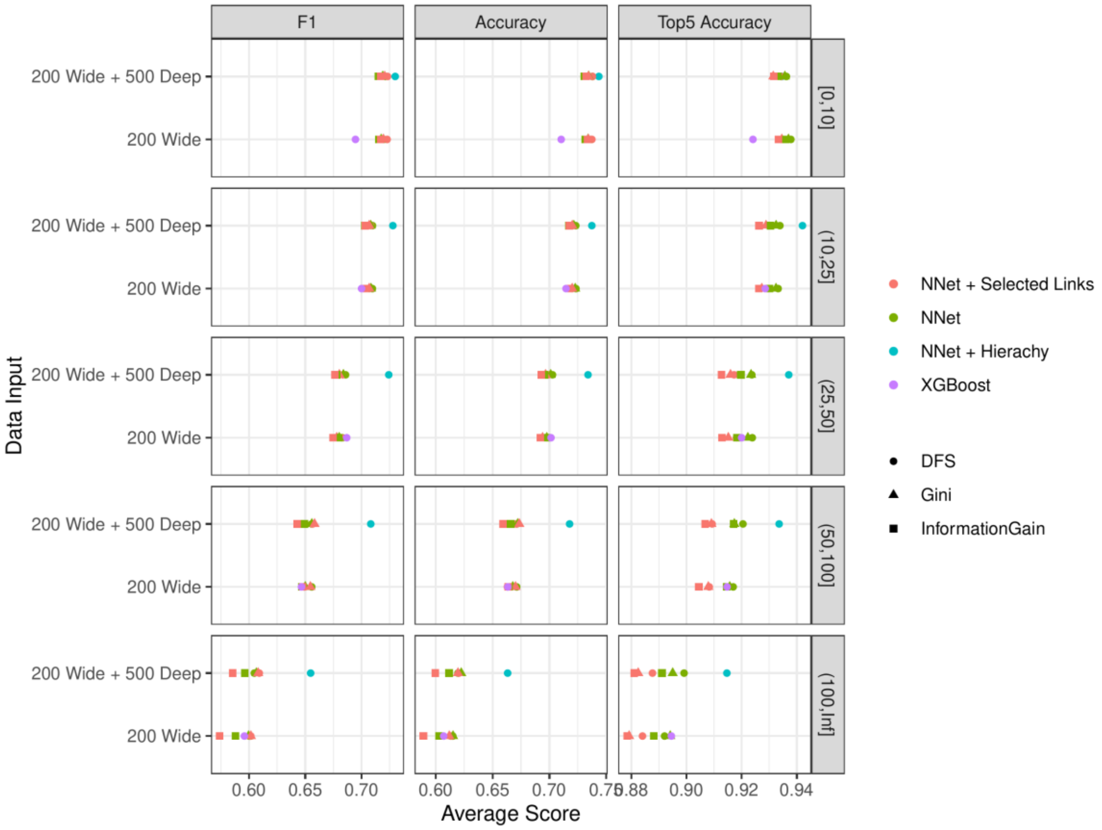

Figure 4.5.4: Average scores per model (color) and feature selection score used (point shape). The horizontal panels distinguish the results by the number of employed people for each enterprise.

Figure 4.5.4 shows the average scores based on the number of employees, grouped into the classes [0,10], (10,25], (25,50], (50,100], and 100+. The figure clearly demonstrates that predicting the 2-digit NACE code becomes increasingly difficult as the size of the company, in terms of employees, grows. Incorporating NACE level 1 predictions (Hierarchy) again shows consistent improvements over the other strategies and models. Notably, the XGBoost model performed significantly worse than the other models for the smallest size class, which could be attributed to the hyperparameters not being fine-tuned for this specific problem.

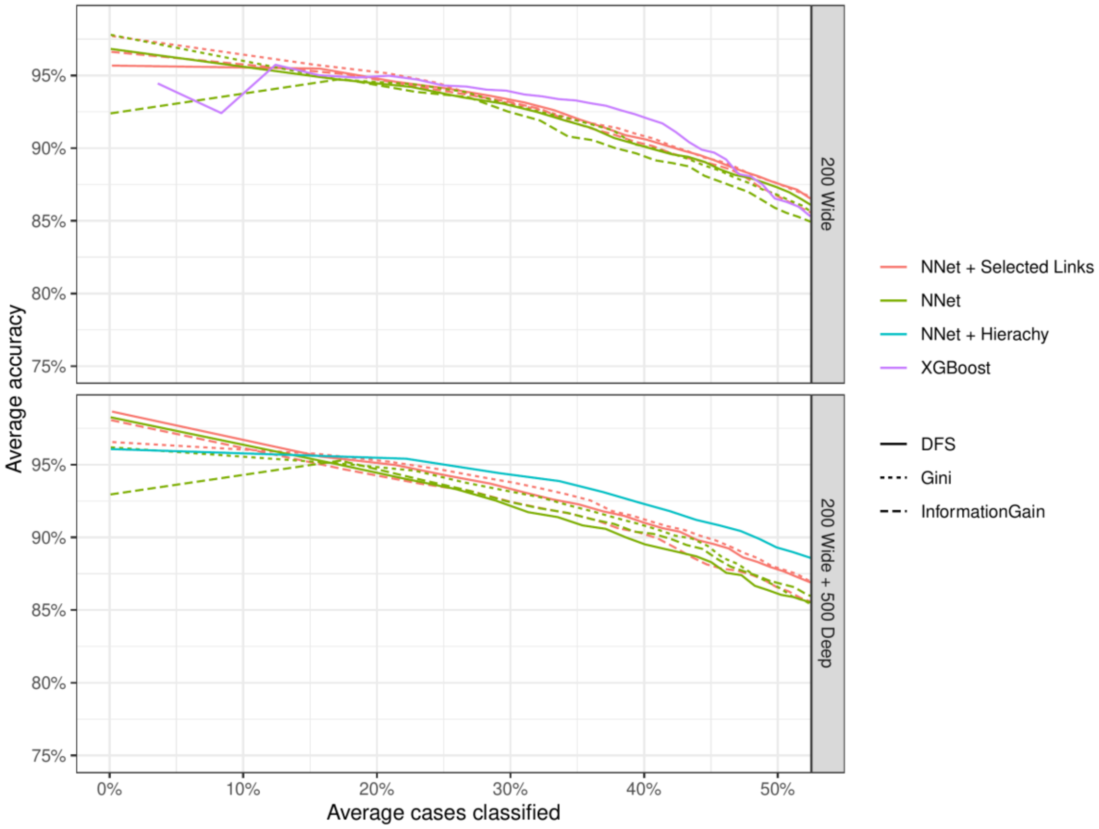

Figure 4.5.5: Average accuracy given average classified cases for the different model specifications (colors) and feature selection score (linetype). The horizontal panel display the inputs used.

Figure 4.5.5 shows the average accuracy achieved (y-Axis) when classifying cases where the most probable prediction yields a prediction probability greater or equal x (x-Axis). For example for the methods using predicted NACE level 1 codes (teal line) directly classifying the 25% of cases with the highest prediction probability results in a prediction accuracy slightly below 95%. From Figure 4.5.5 it is clear to see that none of the models can achieve a high accuracy when directly predicting cases where the model produces the highest predicted probabilities. Using the predictions to support the manual editing of NACE codes seems to be more effective at this stage.

Hierarchical performance measures

In order to respect the hierarchical nature of the NACE code classification, metrics which respect the hierarchy of the classification were additionally used.
Given an enterprise e_i with the true NACE codes on levels 1 to 4 being $c_i = \{c_{1;i}, c_{2;i}, c_{3;i}, c_{4;i}\}$ and $c_i^* = \{c_{1;i}^*, c_{2;i}^*, c_{3;i}^*, c_{4;i}^*\}$ being the predicted NACE codes from a classification model, the category distance between true and predicted NACE codes for level $L = 1, \dots, 4$ can be defined as:

$$Dis(c_i, c_i^*, L) = 2 \cdot \left(\sum_{l=1}^{L} \mathbb{1}_{c_{l;i}^* \neq c_{l;i}} \right)$$

Thinking of the hierarchical classification as a graph of depth L the $Dis(c_i, c_i^*, L)$ is the shortest path between $c_{L;i}^*$ and $c_{L;i}$.

For a given class $\tilde{c}$ the contribution of enterprise e_i and predicted NACE c_i^* being either a false positive (FP) or false negative (FN) predicted is defined as follows:

$$Conb(e_i, \tilde{c}) = \begin{cases} \min\left(1, \max\left(-1{,}1 - \dfrac{Dis(c_i, \tilde{c})}{L}\right)\right) & \text{,if } e_i \text{ is } FP \\ \min\left(1, \max\left(-1{,}1 - \dfrac{Dis(c_i^*, \tilde{c})}{L}\right)\right) & \text{,if } e_i \text{ is } FN \end{cases}$$

Precision ($PR^{CD}(\tilde{c})$) and recall ($RE^{CD}(\tilde{c})$) of a given class $\tilde{c}$ incorporating class distance are then given by

$$PR^{CD}(\tilde{c}) = \frac{\max\left(0, TP(\tilde{c}) + FpConb(\tilde{c}) + FnConb(\tilde{c})\right)}{TP(\tilde{c}) + FP(\tilde{c}) + FnConb(\tilde{c})},$$

$$RE^{CD}(\tilde{c}) = \frac{\max\left(0, TP(\tilde{c}) + FpConb(\tilde{c}) + FnConb(\tilde{c})\right)}{TP(\tilde{c}) + FN(\tilde{c}) + FpConb(\tilde{c})},$$

$$AC^{CD}(\tilde{c}) = \frac{TP(\tilde{c}) + TN(\tilde{c}) + FpConb(\tilde{c}) + FnConb(\tilde{c})}{TP(\tilde{c}) + FP(\tilde{c}) + TN(\tilde{c}) + FN(\tilde{c})} \quad ,$$

where $FPConb(\tilde{c})$ and $FnConb(\tilde{c})$ are defined as

$$FpConb(\tilde{c}) := \sum_{e_i \in FP} C\ onb(e_i, \tilde{c})$$

$$FnConb(\tilde{c}) := \sum_{e_i \in FN} C\ onb(e_i, \tilde{c}) \quad .$$

The class distance based F1-Score can then be computed as

$$F1^{CD}(\tilde{c}) = 2 \frac{PR^{CD}(\tilde{c}) \cdot RE^{CD}(\tilde{c})}{PR^{CD}(\tilde{c}) + RE^{CD}(\tilde{c})}$$

Including the notion of class distance the overall accuracy of a classifier AC^{CD} is proposed as

$$AC^{CD} = \frac{\sum_{\tilde{c}} T\ P(\tilde{c}) + FpConb(\tilde{c})}{\sum_{\tilde{c}} T\ P(\tilde{c}) + FN(\tilde{c})}$$

Figure 4.5.6 shows, in the same fashion as figure 3, the Accuracy and micro F1 score based on the above presented class distance based performance measures. The methods are ranked similar to what was observed in Figure 4.5.3. This might be caused by predicting only up the NACE level 2 codes and not more detailed.

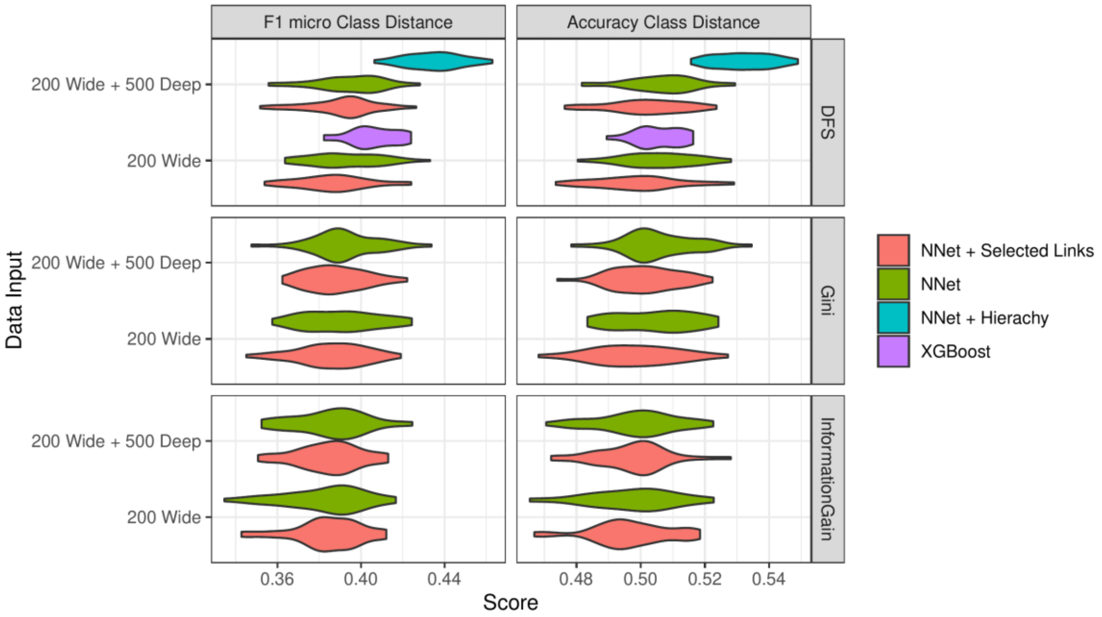

Figure 4.5.6: Overall results using the Neural Network models (NNet), the XGBoost model (XGBoost), using predicted NACE level 1 codes (Hierarchy) and only a subset of available text data (Selected Links). The vertical panels indicate different hierarchical performance measures and the horizontal measures show different feature selection scores.

4.6 NACE prediction at Statistics Sweden

Project Overview and Initial Challenges

At Statistics Sweden early in the project unstructured business descriptions were extracted from business websites. These descriptions, along with the KB-BERT[3] LLM model and other machine learning techniques, were used to classify business NACE codes on 2 through four-digit levels. None of the models were able to produce any results above 60% accuracy, and it was concluded that the approach required further refinement to meet the accuracy standards necessary for production use.

Shift in Focus

To address this issue, the focus shifted to using business descriptions from the Digital Annual Reports (DiÅr), which at the time covered approximately 50% of Swedish enterprises, including both private and public sectors. The data from the Digital Annual Reports not only offered better coverage but also contained more structured and readable business descriptions, making them more suitable for language modelling.

Analysis and Key Observations

We continued to analyse this data source for NACE code classification, focusing on the three-digit levels 16.1, 25.6, 56.1 and 71.1 based on input from domain experts. Using language modelling and both supervised and unsupervised machine learning techniques, some important observations were made:

[3] Swedish case on Hugging Face: https://huggingface.co/KB/bert-base-swedish-cased

- Business activity descriptions alone seldom capture the hierarchical nature of the NACE code taxonomy.
- The degree to which business activities can be accurately described and distinguished using natural language varies among the Swedish NACE code descriptions.

These factors likely affect the quality of the self-reported business descriptions and NACE classifications, and thus the quality of the training data we use for supervised classification models.

Additional Applications and Benefits

In addition to the primary goal of making the NACE 2.1 migration more efficient, we identified several additional applications that could enhance the quality of the Swedish Business Register (SSBR) and improve our classification models:

- Identifying and correcting misclassified businesses in the SSBR.
- Identifying and correcting poorly written business activity descriptions in the SSBR.
- Improving Swedish NACE descriptions.

Ongoing Collaboration

The work continued in close cooperation with the internal NACE 2.1 migration project team. This project is currently ongoing, and our input will be continuously used to establish the implementation approach at Statistics Sweden.

Lessons learned and looking forward

During the project, various experiences highlighted the necessity for an internal position paper. This document underscores the importance of a clear strategy to adapt to the demands of collaborative and exploratory settings like the WIN-project. Such environments require specific development processes and environments. Traditional processes and policies governing Official Statistics production may not efficiently utilize resources and could impede progress towards the project’s goals.

4.7 Contact information discovery at Statistics Hesse

The extraction of characteristics from data obtained from the Internet offers substantial potential for enhancing official statistics. In particular, enterprise websites could act as valuable source of enterprise-specific information that could enrich the Statistical Business Register meaningfully.

Statistics Hesse has worked on two case studies with the goal to extract contact data from enterprise websites. The first case study focuses on gathering e-mail-addresses; the second case study aims to extract the names of business executives/managers. Both studies use imprint pages from enterprise websites, as enterprises must comply with the German tele media act and supply certain enterprise information on the imprint of their website, such as addresses, commercial register numbers, electronic contact information and the names of the business executives/managers. Both case studies evaluate the potential of using imprint pages of enterprise websites as reliable source of information and assess the quality of the data.

E-mail addresses

Case Study Goal and Methodology

The process of extracting e-mail-addresses from enterprise websites follows a deterministic approach, making use of regular expressions. Additionally, deterministic filtering rules are applied to distinguish between relevant and irrelevant e-mail-addresses.

Pipeline Structure

The pipeline for extracting e-mail-addresses follows four main steps:

Step 1: Creating a dataset with webscraped data from enterprise websites
Webscraped textdata needs to be stored in a database (e.g. SQL database), to make the scraped text available for further analysis.

Step 2: Preprocessing the webscraped textdata
The main reason for preprocessing is to convert e-mail-addresses into the standardized e-mail-address-format. The RFC 53224 and RFC 9525 define format, allowed characters, and special symbols for e-mail-addresses. Figure 4.7.1 exemplifies the structure:

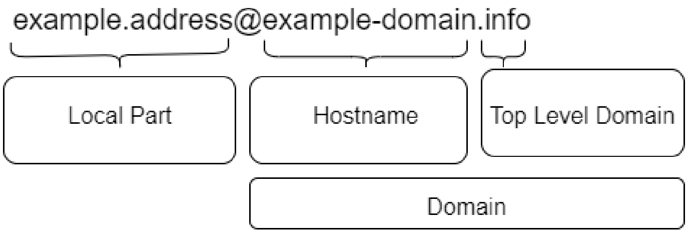

Figure 4.7.1 Structure of e-mail-addresses

Upper- and lowercase Latin letters (A-z), as well as digits 0-9 can be used for the local part and the hostname. While the local part can use a variety of characters (!, #, $, % , & , ' , * , + , - , / , = , ? , ^ , _ , ` , { , | , } , ~, .), the hostname can only consist of '-' and/or '.'.

The Top Level Domain denotes the section after the last dot. For the purpose of extracting e-mail-addresses belonging to enterprises and listed on the enterprise website, two types of top level domains can be distinguished: country code (defined by ISO 31666, e.g. '.de', '.fr') and generic top level domains (e.g. '.com', '.info', '.net', '.org').

A first step in pre-processing is to convert language specific special letters, such as umlauts in German ('ä', 'ö', 'ü' to 'ae', 'oe', 'ue'). As normalization of letters is language specific, this step needs to be adjusted for each country.

On some websites, e-mail-addresses are "protected" by spelling variations in the html code to avoid spam and phishing crawlers (e.g. 'example.address[at]example-domain[dot]info'; 'info @ example-address.de'). However, on the website itself, the e-mail-addresses are visible to the viewer as regular e-mail-addresses (e.g. 'example.address@example-domain.info'; 'info@example-adress.de'). Normalizing these irregularities is easily possible: [at] or (at) can be changed to @, [dot] or (dot) can be converted into a punctuation mark, spaces can be stripped.

[4] https://www.rfc-editor.org/rfc/rfc5322

[5] https://www.rfc-editor.org/rfc/rfc952

[6] https://www.iso.org/obp/ui/#search

Funded by
the European Union

Step 3: Applying regular expressions to find e-mail-addresses
Extracting e-mail-addresses via deterministic rules from text data is relatively easy as there is a recurring pattern according to the rules specified in RFC 5322 and RFC 952 (see step 2). This pattern can be expressed as a regular expression:

```
"[.!'$%&'*+-/=?^_`{|}~A-z0-9]+@[-.A-z0-9]+\\.[A-z]+"
```

A first feasibility study carried out at Statistics Hesse on websites of retail trade enterprises showed that e-mail-addresses could be found on the majority of websites:

Table 4.7.1: Results of e-mail-address-extraction for company websites (in retail trade)

	Enterprises	Domains	E-mail-addresses found
Enterprises with URL	1300	1423	1048
Enterprises without URL	918		
Sum	2218	1423	1048

However, there are also pitfalls in e-mail-extraction. Most notably extracting false positives and/or extracting irrelevant e-mail-addresses.

a) False positives

In the case of false positives, the regular expression matches a string sequence, but the string-sequence is either not an e-mail-address at all (e.g. ‘header_1920x1920@2x.jpg’) or is an obviously incorrect e-mail-address (e.g. //info@teatime-bensheim.de).

These false positives can be eliminated by defining rules that determine which patterns are to be excluded.

b) Extracting irrelevant e-mail-addresses

Relevant e-mail-addresses are defined as e-mail-addresses that belong to the enterprise, whereas irrelevant e-mail-addresses most likely are not owned by the enterprise, despite being listed on the enterprise’s website. Separating irrelevant from relevant e-mail-addresses can be done by matching the local part and/or host name of the e-mail-address found with the domain-URL of the enterprise’s website and/or company name.
Another obstacle is differentiating between multiple relevant e-mail-addresses listed on the enterprise website, e.g. a functional e-mail-address (info@info_stystems.com), as well as addresses belonging to employees (e.g. mister_example@info_systems.com).

Step 4: Categorising found e-mail-addresses into defined categories
To identify relevant e-mail-addresses, the solution is a categorisation system that ranks e-mail-addresses according to their relevance. Table 4.7.2 shows the categorization scheme, as well as the criteria:

Table 4.7.2: Categorization scheme of e-mail-addresses

Criteria	Functional e-mail-address	High probability	Medium probability	Low probability
Local part matches dictionary[7] and hostname matches the domain or enterprise-name	X			
Local part or hostname matches domain and the enterprise name		X		
Local part or hostname matches domain and the enterprise name and the e-mail-address was found on an imprint page		X		
Local part or hostname matches the domain or enterprise name			X	
Local part matches exclusion dictionary[8]				X
E-Mail-Address listed on the enterprise website				X

Both functional and high-probability e-mail-addresses are most likely to be the relevant email address for the enterprise.

Names of Business Executives / Managers

Case Study Goal and Methodology

The primary goal of this case study is to extract the names of business executives/ managers from imprint websites of enterprises. To achieve this, a Named Entity Recognition (NER) approach was combined with a dictionaries approach. In the last step, the results were validated.

Pipeline Structure

The pipeline for extracting information on business executives/managers was divided into three main parts:

Step 1: Preparation
A dataset containing imprint websites of enterprises was used. This stage involved removing superfluous

[7] The dictionary contains frequently used words for functional e-mail-addresses, such as ‘info’, ‘service’, ‘shop’, ‘support’. The dictionary is country-specific and should be created separately for each language.

[8] The exclusion dictionary contains words signalling irrelevant or false e-mail-addresses that appear regularly on or in the html-code of websites, e.g. ‘example’, ‘test’, ‘job’. Again, the exclusion dictionary is country-specific and should be created separately for each language.

content from imprint pages and extracting a so called “word window”. A “word window” displays a defined number of words around certain key phrases – such as “Business Executive:”, “Director”, “Manager:”, “Authorized to represent” – as the names of business executives/managers' most likely appear in this context. This process ensures that names that appear by chance on the website, but do not belong to the management, are not extracted. The result is a dataset of “word windows” for each imprint website - only if the keywords were found.

Step 2: Named Entity Recognition (NER)
Utilizing a NER algorithm (‘spaCy’), this step focuses on extracting names of business executives/managers from the prepared dataset. This process also included separating first and last names. However, ‘spaCy’ wasn't always successful in identifying names listed in the 'word windows', leading to the additional incorporation of another approach.

Step 3: Dictionary Method
To augment the NER process, a dictionary approach was introduced. This approach utilized the ‘babynames’-package for R, containing names of babies born in the US between 1880 and 2017. While the list includes a high number of first names, a few popular German first names are missing. Therefore, a dictionary of German first names was additionally included and used to match as many first names as possible. The dictionary was then applied to the word windows to identify first names. To determine last names, the consecutive string was also extracted up to a line break. This method proved effective in most cases for obtaining full names.

Results

The case study involved 1476 imprint websites of enterprises, with the approach being able to extract names of business executives/managers from 1046 websites. This means the approaches did not find any names for around 30% of websites. However, this is not necessarily due to the fact that no names are listed on the websites. In most cases, the names of the business executives/managers are listed, but not in the expected position on the website, which in turn causes the word window to fail. In other cases, the given name, which the word window captured correctly, was so unusual that neither the NER algorithm nor the dictionary were able to identify. Overall, from 1046 found names on websites, 915 names were identified by the NER approach, 131 names were matched by the dictionary approach.

For the evaluation, 399 names were manually validated against a dataset of correct names. Results show that 313 names were extracted correctly from the enterprise website and are the correct names of the current business executives/managers. Therefore, the approach has an error rate of around 22%.

4.8 Search for contact information and new establishments at Statistics Finland

Google Maps Places API

Google Maps Places API is a service that allows access to Google Maps location through standard URL requests. There are multiple ways of calling the API and the results are returned as JSON or XML response. Limit for free use is 28 500 map loads per month.

We tested the API throughout Finland to find contact information and new business establishments. We used text search with business name and municipality as parameters. A first test was done for two companies known to have multiple establishments and not keen to answer surveys. The results revealed some wrong addresses in Business Register and new establishments for the companies. Not all new establishments were found, however, as Places API gets updated by the users and not all business owners immediately update their establishment addresses to Google Maps. Maps can be used in pointing the establishment location on business web pages without updating the actual data to Maps.

We then proceeded in trying to find addresses for 1 000 business id's with multiple establishments and less than 5 person years that have rarely taken part in the inquiry on establishment structure and personnel. Initially we tried using postal codes as a parameter instead of municipality, but the number of requests we would have had to make would have been too much. However, keeping in mind the number of free requests that can be made per month and how long making a request takes, the API would be very useful in detecting addresses for single establishment businesses. If more than one address were returned it would be obvious that the Business Register was not up to date.

There's also a possibility to get a lot more information on the business by using Googles place ID and Place Details -search. Place ID is available through for example Text Search, so after doing a basic search for business addresses, another search could be done for the business websites. We checked this for one business and the result was an URL that pointed to a subpage for the establishment addresses.

We managed to run roughly 30 business names and all municipalities in Finland per business before Google sent a potential violation of Google's Acceptable Use Policy -notice, after which the run was stopped. We responded to Google and after a couple of days and some back and forth, we were given permission to continue with the precaution of fixing the issue that led to the notice. Taking a look at the Terms of Service uncovered the following:

e. Prohibitions on Content

Unless expressly permitted by the content owner or by applicable law, you will not, and will not permit your end users or others acting on your behalf to, do the following with content returned from the APIs:

1. **Scrape, build databases, or otherwise create permanent copies of such content**, or keep cached copies longer than permitted by the cache header;
2. Copy, translate, modify, create a derivative work of, sell, lease, lend, convey, **distribute**, **publicly display**, or sublicense to any third party;
3. Misrepresent the source or ownership; or
4. Remove, obscure, or alter any copyright, trademark, or other proprietary rights notices; or falsify or delete any author attributions, legal notices, or other labels of the origin or source of material.

The most restricting parts are bolded, and as permanent copy includes any format of storing the data including Excel, csv, json or txt, there is no way of using the data in Business Register or even in surveys as reference material. Systematic data retrieval is also what we were doing, meaning scraping, which is also prohibited. Therefore, the Terms of Service fairly comprehensively ban the use of Places API and the results received in any way that could be usable in enhancing a Business Register.

The express permission or applicable law are only ways to get access to and use the data. Hence, we contacted the sales department to see if there is an Enterprise or a paid version we could use. Sales, however, told us that they did not give legal advice and that the Terms applied to all contracts, leaving us with more questions than actual answers. As to the applicable law, our own legal department takes a very careful stance on using the Places API.

Security issues were also raised inside Statistics Finland as to how Google handles the information given in the requests. The questions that should be answered before Places API use would be allowed are: 1) does Google store data during requests, 2) what data does Google store, 3) what does Google do with the stored data, and 4) where are the data stored and requests handled. As the Terms of Service does not allow us to use the API in a productive way, answering these questions has not yet been relevant. There is an ample amount of information on security and data of all Google services on their website.

Bing Maps and Open Street Map

We also took a quick look at the Bing Maps API Terms of Use and some documentation, which weren’t as restricting as Google's Places API Terms of Service, but there are for example prohibitions to building databases.

> **Section 5. General Restrictions.**
> When using the Services, you may not, nor may you permit End Users to:
> • **(k)** Redistribute, resell, or sublicense access to any Microsoft service or Content;
> • **(n)** Copy, store, archive, or create a database of Content, except that you may store geocodes locally for use solely with your Applications;
> Additional restrictions may apply to use of particular Content or functionalities, as set forth in the Documentation from time to time.

Bing Maps does not seem to have a direct counterpart to Places API. Manual search of a multilocation business turned out similar data as Google Maps.

Open Street Map (OSM) may be also useful. A quick manual search for multilocation business turned out 4/5 locations, and a bulk load for areas such as countries and cities are available. Unfortunately, more advanced API search did not produce the same results as manual search and currently we do not consider OSM as a potential source of information. The map services are dependent on individuals and businesses to update the platforms which of course creates a level of uncertainty regarding the accuracy and timeliness of the data.

4.9 Wrap-up

In this chapter we explored in depth the concept of enhancing the business register from web data, with NACE as an important target.

With respect to the NACE prediction the results presented by CBS and Statistics Austria both paint a similar picture. In both cases a fully automated classification of NACE codes using data from enterprise websites seems infeasible at the time. Interesting to note is that the choice of classification model does seem to play a minor role with respect to the prediction quality. A more important aspect seems to be the quality of the input data. From some NACE codes the website texts are too diverse and not specific enough to result in a good NACE prediction. The challenge here is to find better input data. A related aspect is that the quality of the pre-processing is important as well as the quality of the collected training and test data. Aspects of the latter are that one needs a sufficient amount of units per NACE code with correct labels and that the training set is not too unbalanced.

An important aspect of the NACE is its hierarchical structure where at deeper levels the distinction between categories become more subtle and detailed. This also makes accurately predicting NACE codes at lower levels increasingly difficult. A difficulty that also extends to human classification, given the complexity of businesses and their economic activities.

This leads to reason that even though fully automated NACE classification is not within reach, integrating the NACE classification into the manual NACE editing process, for instance in form of a recommendation system, can still hold value for improving the quality of NACE coding overall.

Contact information discovery as performed by Statistics Hesse is a field where other statistical institutes can profit. Although country legislation such as imprint rules may vary in practice, the concept of extracting meaningful information from contact pages and other administrative pages might really improve SBR quality.

5 Cross-cutting subjects

5.1 Literature review on webscraping for statistics

At the start of the WIN project a literature study was performed. The study focused on papers on the most common topics described in this report: URL finding methodology and tooling, and on the use of business websites to predict economic activity. The aim of the study was to build on existing knowledge and experiences as much as possible and learning from what already has been done.

About 20 papers were identified: two papers on URL finding, fifteen papers on predicting economic activity using text mining, one paper on scraping enterprise websites and one overview paper on feature engineering in text mining. Most of the papers have been summarised in terms of:

- General aspects: type of data sources, statistic of interest, training and / or prediction methods, quality measures
- Methods: description, assumptions, strong points and limitations
- Application: agency, topic, datasets used, and results

The table below gives a concise overview of the papers reviewed and some findings from the summaries.

Title	**1st Author yr**	**Some findings**
Classifying Websites by Industry Sector: A Study in Feature Design	Berardi 2015	Prediction of economic activity; Binary SVM classifier, one-versus-rest. Assign to three nodes (rather than one).
On the use of internet as a data source for official statistics: a strategy for identifying enterprises on the web	Barcaroli 2016	Linking websites to statistical units, search via Bing; Multiple ML models tested; Java program available
Classifying Firms with Text Mining	Caterini 2018	Forecasting reactivations of enterprises applying ML (LDA, Naïve Bayes, Random Forest)
A Heuristic Approach for Website Classification with Mixed Feature Extraction	Du 2018	Website encoder using Word2Vec, Text CNN Feature Extractor, Bidirectional GRU Feature Extractor, Fully Connected Classifier
Exploring a knowledge-based approach to predicting NACE codes of enterprises based on web page texts	Kühnemann 2020	NACE prediction; Classical Machine learning (SVM, Multinomial Naïve Bayes), Hierarchical classification, Dictionary-based
Automated Industry Classification with DeepLearning	Wood 2017	
Classifying businesses by economic activity using webbased text mining	Roelands 2017	Naïve Bayes, logistics regression, SVM classifier, random forest k-nearest neighbour. Knowledge-based features versus automatic feature selection. TF versus TF_IDF approach. Single label versus multilabel approach
Searching for business websites	van Delden 2019	Search for URLs; Naïve Bayes, SVM classifier, random forest model trained on a training set coded with correct / incorrect URL; Python program available
Feature Engineering for Text Classification	Scott	Text classification methods; relatively old paper, year unknown

Using Public Data to Generate Industrial Classification Codes	Cuffe 2019	The use of Google Places API for access to business information; 2-digit NAICS classification for approximately 120,000 single-unit employer establishments; Doc2Vec; Random Forest
An Automated Industry Coding Application for New U.S. Business Establishments	Kearney 2005	
Using Machine Learning to Assign North American Industry Classification System Codes to Establishments Based on Business Description Write-Ins	Dumbacher 2019	NAICS (North American Industry Classification System) prediction using bag-of-words
Creating an Automated Industry and Occupation Coding Process for the American Community Survey	Thompson 2012	
Automation of NOGA coding (NOGAuto)	FSO (Swiss) 2021	Targets NACE and so-called NOGA classification of businesses; more a preliminary abstract than a report on results
Web Mining of Firm Websites: A Framework for Web Scraping and a Pilot Study for Germany	Kinne 2019	
On the Automated Classification of Web Sites	Pierre 2001	Targets NAICS classification from domain registry data, website texts and annual reports; uses classical KNN
Improving Text Classification by Shrinkage in a Hierarchy of Classes	McCallum 1998	
A Study of Approaches to Hypertext Categorization	Yang 2002	Classifying company websites into industry classification using Naïve Bayes, kNN, and First Order Inductive Learner
Creating an Automated Industry and occupation Coding Process for the American Community Survey	Thompson 2012	

All in all, we can say that many papers have been written before on linking web data to SBR units and classification approaches using web data over time. They can be a useful and inspirational source for further work, but they also have to be seen in the context of evolution of the web, which continuously grows in size, complexity, and the state of play of text mining and ML methods, which also develop rapidly.

The full results of the literature study and can be found on the wiki (accessible to WIN members): https://webgate.ec.europa.eu/fpfis/wikis/display/WIN/UC+5+Literature

5.2 Software perspective

In an ideal situation any software developed in the scope of any activity in this project at any stage of maturity would be open source from the beginning, aligning to the principle „open by default" of the ESS principles on open source software. In practice, however, things are different. The work in this project builds on earlier initiatives which have been implemented in various closed or open source software solutions. The partners in this project used the software at hand to reach the results described in this document. Also, because of the tight integration to national Business Registers, the work was not executed on the WIH but on national premises.

All in all, the software landscape of the URL finding initiatives is as far as we know:

- Statistics Netherlands has an open source URL finding implementation: https://github.com/SNStatComp/urlfinding
However, this was implemented some years ago and Statistics Netherlands now uses a 3rd party data approach, which makes the maintenance of this software not a focus point.
- Statistics Hesse has a mature in-house R-based URL finding software solution, aligned with the URL finding methodology described in this document, which is not open source, due to complexity and restrictions caused by certain non-open source parts.
- Statistics Austria implemented its URL finding into R and Selenium, which is successfully used internally but at this moment is not available as open source to others.
- From an earlier ESSnet project there is a Python based solution for URL finding still available and periodically updated by ISTAT: https://github.com/SummaIstat/GUrlSearcher

The software landscape for the „enhancing the business register phase" is more diverse. Due to the experimentation with different methods, the strong relationship with the setup of the national SBR and the use of quickly evolving machine learning components, no generic reusable open source software has been developed up to now. This could be something to work on in a consecutive project.

All in all, we think that for the URL finding activities it would certainly be possible to build on the existing solutions to develop an one-for-all NSIs URL finding solution, building on one or more of the most common search engines. This would require additional work, which should also be weighed against the possible use of data from the work-in-progress EU search engine project. More on that in section 5.4.

5.3 Quality perspective

Within the WIN project the WP3 use cases worked together with WP4 task quality. The final deliverable of WP4 task quality contains a quality perspective on all the work in the WIN project, including the use cases of WP3.

The WP3 deliverable contains an extensive section discovering data sources / landscaping , including subsections for exploring all websites as well as to finding representative websites and reflects the problem as a Multi-Criteria Decision Making problem. It also describes a specific selection model for new use cases such as the one described here.

The rest of the document describes the quality perspective in different phases of the statistical process chain in the input and throughput phases and concludes with guidelines for a centralised web data infrastructure.

For more detailed information on the quality perspective, we refer to the deliverable from WP4.

5.4 European perspective

Two topics, which were not studied in this project but came up on the run, we think deserve to be mentioned here:

- The relationship between National Statistical Business registers and the European Business Register (EGR): https://ec.europa.eu/eurostat/web/statistical-business-registers
Working on advanced methods for SBR enhancement it is valuable to keep in mind that certain units might be present in multiple administrative sources, which makes working together and

sharing (web) data and web retrieval methods among organisations important. This has not been studied in this project. We mention it here for possible further exploration.

- The European search engine project: https://openwebsearch.eu/
  This project builds up an index of the web, which might be of great interest to statistical organisations for enhancing their SBR.
  It could potentially be used, for example for finding SBR units from this index, possibly replacing search engine-based URL finding methods. Depending on the frequency scraping, the index could be a safe replacement for web data discovery, which is often done via (paid) commercial search engines. A clear example of this is URL finding as explained in this document. In addition the texts in the index might be useful for NACE detection or other text-driven classification / coding tasks. Also, the index could also be an instrument for creating new official statistics on aspects of the Internet (growth/size/variety), website characteristics, techniques used or other phenomena on the internet, however this is not the subject of this use case.

6 Discussion and future work

This deliverable provides a comprehensive overview of the methods and approaches employed by participating countries to enhance their statistical business registers (SBRs) using web data. These registers are a central production component for official statistics. Improvements in such register pave the way to control representational errors in all business statistics and contribute to overall quality, which underlines the relevance of this work. The primary focus is on two key areas: URL finding and business register enrichment, with a particular emphasis on NACE.

With respect to ***URL finding*** the approach as demonstrated by Statistics Austria, Statistics Hesse, and Statistics Sweden (and in earlier projects by Statistics Netherlands and Statistics Italy) has emerged as a proven and effective method for discovering websites relevant to statistical units. While there are variations in terms of search engines, queries, and machine learning techniques used, the work described here provides a solid foundation for URL finding. In addition, Statistics Finland has explored the potential domain registry data to complement the web data portfolio.

Statistics Netherlands found that linking third party data to the business register can be valuable, if the third party data contains unique and/ or non-unique identifying variables, with not too many missings. Experiences by Statistics Hesse and Statistics Netherlands are that the third party data cannot yet be linked sufficiently well to replace other means of URL finding. Although most linkages are one-to-one, there are also a considerable number of many-to-one, one-to-many and many-to-many linkages. For those more complex linkages decisions have to be taken which URLs to choose, which is likely to depend on the variable that one aims to improve. For the very large, complex and influential units such improvements always need to be checked / done manually. However for the smaller and middle-large units, we should aim for an automated procedure

Regarding ***business register improvements***, and in particular targeted to NACE, the deliverable highlights the importance of distinguishing between NACE prediction, determination, misclassification detection, and support for NACE transitions. Statistics Netherlands showed how NACE misclassifications can be detected and new ones predicted. Both Statistics Austria and Sweden showed how NACE prediction methods can be put into practice. In fact, all those NACE contributions focus on identifying the correct *main* economic activity of a specific *website*. In future research it would be useful to study if we could also automatically predict or verify *secondary* economic activities. Additionally, there can be a difference between the (main) economic activity mentioned on the website and the main economic activity of the related legal or statistical unit. That problem cannot easily be solved automatically.

With respect to other variables than NACE, Statistics Hesse showed very mature work on discovering and improving contact information in the business register from web data and Statistics Finland explored detecting new establishments using map data. Most of these methods are a mix of automatic predictions and manual checks.

Concerning ***future work,*** we believe that the results presented can be applied to any other statistical organisation, the best way to do this depends on the business register setup in that organisation, the richness and openness of the web in the respective country and other specific circumstances. Without doubt, it is essential that natural language processing techniques must become a routinely statistical production tool in statistical offices in order to exploit the richness of data sources as described in this study. Moreover, in our vision future international work should concentrate on implementing proven methods that have the highest chance of applicability in multiple contexts. The creation of a mature open source software implementation for URL finding building on the experiences in this project might be such an endeavour, as is a possible exploration of the use of the openwebsearch project results. Common practices and implementations of NACE detection (or prediction etc.) and improving administrative information are also worthful to further invest in.

After working on implementations, it would be good to put effort into improvements and extensions of the methods. Some examples of extensions are: accounting for the dynamics in URLs (keep the URL lists up to date), make a distinction between the main and secondary URLs of businesses, and use of other online sources than web data for business register enhancement.

The diverse experiences and approaches presented in this study show how the choice of complementary data sources might influence the result. The search for high-quality textual data and the integration with numeric data and linking strategies with trustful data already under control of the NSI might be promising directions of research. This links to the relatively new idea of selective / statistical scraping (see the methodology deliverable of this project from WP4) where as much as possible web data is explored from a context already known to the statistical office.

With respect to the detection of indicators from web data, this document had a focus on NACE. Nevertheless it’s good to stress that many of the same methods might be applicable to derive other indicators, such as for example:

- Patents
- Government programs and subsidies
- Businesses structures (also from press releases)
- Financial information
- Association membership
- Green policies
- Relationship with SDGs

All in all, we conclude that business register enhancement from web data is a topic that already proved to be valuable in multiple statistical agencies. Moreover, as a business register is an essential statistical instrument for many economic statistics, improvements directly pays off in terms of quality. It is definitely worth continuing to invest in.

7 Acknowledgements

This report underwent a thorough review by David Salgado from INE, Statistics Spain. We express our gratitude for his positive feedback, constructive comments, and the raising of some strategic issues about the continuation of this work. We used the review to enhance the quality of this document.

We further want to thank all colleagues in all participating institutes for valuable discussions supporting the work which could not have been done without the knowledge of many specialists on business registers, administrative sources, classifications, text mining, machine learning and many other topics.

We finally thank the colleagues from Eurostat for the many fruitful sessions where we discussed how we can bring the challenging field of web data for official statistics further from a national and international perspective.

8 References

- Abadi, Martín, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, et al. 2015. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems." http://tensorflow.org/
- Allaire, JJ, and François Chollet. 2019. *Keras: R Interface to 'Keras'*. https://CRAN.R-project.org/package=keras
- Bosch, O. ten, Delden, A. van, Wolf, N. de (2023). Business register improvements: a balance between search, scrape and 3rd party data, NTTS conference, March 2023, Brussels
- Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324
- Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., and Jatowt, A. (2020). YAKE! Keyword extraction from single documents using multiple local features. Information Sciences 509 (2020) 257–289.
- Chen, Tianqi, and Carlos Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. KDD '16. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785
- Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. 2023. *Xgboost: Extreme Gradient Boosting*. https://CRAN.R-project.org/package=xgboost
- Daas, P.J.H., van der Doef, S. (2020) Detecting Innovative Companies via their Website. Statistical Journal of IAOS 36(4), pp. 1239-1251, https://doi.org/10.3233/SJI-200627
- Daas, P., Hassink, W., Klijs, B. (2023). On the Validity of Using Webpage Texts to Identify the Target Population of a Survey: An Application to Detect Online Platforms. IZA - Institute of Labor Economics discussion paper 15941, February. https://docs.iza.org/dp15941.pdf
- Gussenbauer, J., Toledo, E., Delden, A. van, Windmeijer, H.J.M. and Kowarik, A. (2022). Using webdata to derive the economic activity of businesses. Paper presented at the European Conference on Quality in Official Statistics, Vilnius, Lithuania, 8-10 June, 2022
- Harrison, John. 2020. *RSelenium: R Bindings for 'Selenium WebDriver'*. https://CRAN.R-project.org/package=RSelenium
- Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. "Bag of Tricks for Efficient Text Classification." *arXiv Preprint arXiv:1607.01759*
- Kühnemann, H. Et al (2022). Report: URL finding methodology, Draft (ver. 5.0), 2022-01-31, ESSnet WIN, WP2 & WP3, UC5, https://cros.ec.europa.eu/book-page/web-intelligence-network-reports
- Kühnemann, H (2023, March 7-9) URL Finding Methodology (Conference Presentation) Conference on New Techniques and Technologies (NTTS) for Statistics, Brussels, Belgium
- Raunak, Vikas. 2017. "Simple and Effective Dimensionality Reduction for Word Embeddings." https://arxiv.org/abs/1708.03629
- Schalken, N., Delden, A. van, Scholtus S. and Windmeijer, D.W.J. (2023). Automatic detection of NACE misclassifications with multiple sources. Paper presented at the European Establishment Statistics Workshop 2023; Lisbon, 19-22 September, 2023. Available at https://sites.google.com/enbes.org/home/home/news-and-events/eesw23/eesw23-programme

- Sozzi, A. 2017. Measuring Sustainability Reporting using Web Scraping and Natural Language Processing, Msc-dissertation, Office for National Statistics (ONS), https://github.com/AlessandraSozzi/MSc-dissertation
- Sun, Aixin, and Ee-Peng Lim. 2001. “Hierarchical Text Classification and Evaluation.” In *Proceedings 2001 IEEE International Conference on Data Mining*, 521–28. IEEE.
- Li, X. et al. (2019) Improving rare disease classification using imperfect knowledge graph. BMC medical informatics and decision making 19, 1–10.
- Oosterveen, V., Delden, A. van, and Scholtus, S. (2021). Notice the noise: detecting misclassifications in register data. Paper for NTTS 2021, 9-11 maart Brussel. Available at https://coms.events/NTTS2021/data/abstracts/en/abstract_0068.html (Accessed Dec. 2022).
- Uysal, A. K. (2016). “An Improved Global Feature Selection Scheme for Text classification.” Expert Syst. Appl. 43 (C): 82–92. Available at https://doi.org/10.1016/j.eswa.2015.08.050
- Wright, Marvin, and Andreas Ziegler. 2017. “Ranger: A Fast Implementation of Random Forests for High Dimensional Data in c++ and r.” *Journal of Statistical Software, Articles* 77 (1): 1–17. https://doi.org/10.18637/jss.v077.i01