Discussion paper

Searching for business websites

Arnout van Delden
Dick Windmeijer
Olav ten Bosch

December 2019

Content

Summary

Enterprise websites are a promising source of information for official business statistics. In that context it is important to know the linkage between business website addresses (URLs) and a business population frame: the general business register (GBR). More specifically, we are interested in the 'domain', which is a part of the URL. Within their GBR, Statistics Netherlands has already obtained domains for about one third of the legal units from the Chamber of Commerce (COC). Legal units have provided those domains when they registered at the COC.

As a first step to update the currently available domains we have linked an external data set, obtained from the company DataProvider (DP) to the GBR. The data were linked by exact linkage using the domains and legal unit identification numbers. As the next step, we have developed a URL finding methodology, which is based on a supervised machine learning approach. We have limited ourselves to legal units that are one-to-one linked to enterprises and to enterprises with 10 or more employees.

We first created a labelled set of legal units, split up into 'website+' legal units from the COC, 'website+' legal units from DP and 'website-' legal units. 'website+' legal units are legal units with a known domain and 'website-' legal units are legal units known to have no website. For the latter, one of the sources were respondents to the ICT survey. Next, we used contact information of legal units in the GBR, such as their legal name and address, and automatically searched for URLs using Google API. This resulted in a set of candidate domains for each legal unit. Next, a machine learning model was trained, using the labelled set, to predict the probability that a candidate domain corresponds with the correct domain. We then select the domain with the highest probability. If this probability is above a particular threshold we consider the candidate domain to be correct and otherwise we consider it to be incorrect.

The URL retrieval model resulted in an average F1 score of 0.80 over two label categories, when the candidate domain with the highest probability, the top-one domain, was selected. We conclude the paper by discussing various possibilities to improve the URL retrieval model.

Keywords

Machine learning linkage, URL retrieval, domain, General Business Register

1. Introduction

Internet is a promising source of information for official business statistics. For instance business websites could be used to extract up to date contact information of businesses. Further one might use the information from website texts to assess the economic activity of businesses (Berardi et al., 2015). Statistics Netherlands is interested to use website text information to classify the population of businesses by characteristics that not systematically collected in administrative data. Examples of such classifications are whether a business is innovative or not (Van der Doef et al., 2018), whether it concerns a family businesses or not (Bosch et al., 2016) and whether a businesses has a webshop or not (Oostrom et al., 2016).

The interest by Statistics Netherlands to use website text information to derive new classification variables in business statistics was the motivation for the current discussion paper. For many applications of the use of website information in official business statistics, it is important to link website addresses (URLs) to a frame that contains the population of statistical business units. This frame is further referred to as the General Business Register (GBR). The base unit type, from which the statistical units are a composite, is the legal unit. Legal units are units that register at the Chamber of commerce (COC) when they start as a business. Therefore, in the current paper, we are interested to derive the link between a legal unit and a URL. Websites also regularly display a legal number.

One important statistical unit is the enterprise. The enterprise is the statistical unit of the short term business statistics, which is an indicator of the economic business cycle and it is the statistical unit of the Information and Communication Technologies (ICT) survey, which is a survey on IT use by businesses. In the present paper we will use this ICT survey as one of the sources to identify units without a URL since this survey asks enterprises whether they have a website or not. Unfortunately, the URL itself was not asked for in the year that we used their data. The ICT survey is limited to enterprises with ten or more employees, we will therefore (also) limit ourselves to legal units that are related to an enterprise with ten or more employees.

Additionally, we have two more limitations to the scope of this study. First of all, as a starting point for developing a methodology linking URLs to a set of statistical units, we limit ourselves to one of the most simple situations: the case where we seek URLs of legal units that are one-to-one related to an enterprise. In practice, many of the smaller enterprises have a one-to-one relationship with a legal unit, so this starting point is relevant for the practice of official statistics. Second, we limit ourselves to one-to-one linkages between a single URL to a legal unit. In fact, some of the legal units may be related to multiple websites. For instance, different websites may concern different products or different establishments (local units) of the same legal unit. Also the opposite may be true: one website may relate to multiple legal units.

Currently, for about one-third of the legal units in the GBR that are related to an enterprises a URL is recorded. This URL has been obtained from the COC, that registers the URLs of businesses. Part of the URLs in the GBR are outdated, since businesses usually do not provide updated information to the COC.

As a first approach to update the currently available URLs and to add new URLs we have linked an external data set containing domains and contact information to the GBR. These external data were obtained from the company DataProvider, and are further referred to as DP data. That approach has been taken before by Oostrom et al. (2016). As a subsequent step, we selected all legal units without a URL for which we aim to find a URL. Those legal units are referred to as the target legal units. For each of those legal units, we use the contact information to search for URLs through a search engine. In the present paper, we refer to the latter approach as URL finding. An advantage of URL finding is that it can be applied to populations that differ from the ones found in the GBR and that it can be used to find URLs which are additional to URLs found in external data sets. Another advantage is that when the external data suffer from undercoverage compared to the GBR or when the data contain errors in the identification variables, resulting in erroneously missing linkages, URL finding can be applied to the remaining units.

Using contact information to search for the website of a target legal unit will often result in multiple search results. Each combination of a URL search result with its target legal unit leads to a candidate pair for linkage. Subsequently, one aims to select the true linkage among the candidate pairs. This true linkage is also referred to as a match. One approach to select the true linkage (a match) among a set of candidate pairs is probabilistic linkage, see Felligi and Sunter (1969), Hertzog et al. (2007) and the literature overview in Ariel et al. (2014). Probabilistic linkage particularly useful when there is not necessarily a full similarity between the identification variables in the two sources that are to be linked. In our situation there is often no full similarity between the contact information of target legal unit and the contact information of candidate websites in the search results.

Probabilistic linkage is an unsupervised method in the sense its model parameters can be estimated without having a sample of pairs that are labelled to be true or false links. A disadvantage however is that it relies on a rather strong assumption. The availability of a set of labelled examples can be used to improve the accuracy of the selection which candidate pairs are a true link (Tuoto, 2016). Given the presence of these labels, one can use a supervised learning method to estimate the probabilities that candidate pairs are a true link or not. Examples of using supervised machine learning for linkage of data sets are Cochinwala et al. (2001) and Christen (2008). In the current paper, we also use a supervised machine learning approach. We make use of a set of already known URLs. Such an approach has been used before by Barcaroli et al. (2018) as part of an European Project (ESSnet) on big data. The objectives of the current paper are to develop a URL finding methodology and a first version of tooling to apply it. The methodology is generic and can also be used by other NSIs.

The remainder of this paper is organised as follows. In Section 2 we describe the general approach used in the present paper. In section 3 the data used in the current paper is described. In the Sections 4-6 we describe the design, training, selection and testing of the machine learning model. Finally, in section 7 we discuss the main findings, points for improvement and give directions for future research.

2. General approach

For the process of searching URLs for legal units for which we currently do not know their URL, we use a supervised machine learning approach. More specifically, we are interested to find the so called 'domain' part of the URL, which is further explained in section 4. In the current section we limit ourselves to the most important steps, drawn in Figure 1. The exact procedure and their results are presented in subsequent sections. For the remainder of this paper we distinguish among two groups of legal units: a) legal units that (in reality) do not have a website, b) and legal units that (in reality) do have a website. These two groups are referred to as the 'website-' and 'website+' legal units respectively.

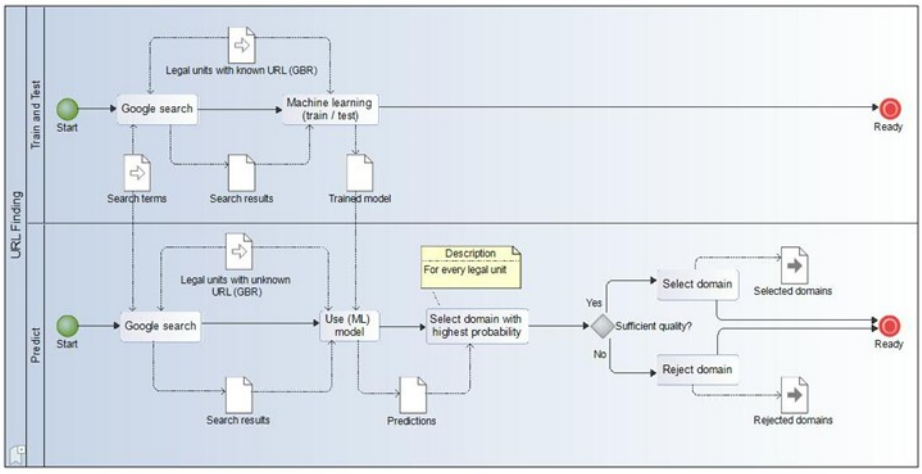

Figure 1. Steps in URL finding.

We distinguish two phases: a 'train and test phase' and a 'prediction phase', shown by the two lanes in Figure 1. We start by creating a sample of legal units of the population which are to be used in the labelled set. We sample from the population of and 'website+' legal units and we identify a set of 'website-' legal units, see section 3. Next, we use contact information, such as the legal name and the address and automatically search for URLs using Google API. We select the domain part of the retrieved URLs and keep the set of unique domains (see section 4). Those domain are also referred to as candidate domains. Then we derive and order a number of features to be used in machine learning models (see section 5). Next, we train different machine learning models to predict which of the candidate domains of a

legal unit corresponds to the correct domain. The best performing model will be used in the prediction phase (see section 6). In the prediction phase we search domains for legal units for which we do not yet know their website, using the same approach as described in section 4 and 5. The applied model returns an estimated probability to be the correct domain for each candidate domain. We then select the candidate domain with the highest probability. If this probability is above a particular threshold we then consider the domain to be correct, otherwise we consider it to be incorrect. As a possible refinement for the future, one might define two thresholds and when the estimated probability is in between those two thresholds one needs to manually validate whether the selected domain is a match or not.

3. Prepare a labelled set

The first step 'prepare a labelled set' consists of the sub steps:

- add known URLs to the population frame, section 3.1
- select the target population, section 3.2
- sample from the 'website+' legal units and derive 'website-' legal units. section 3.3

3.1 Add known URLs to the population frame

In the present paper, we have used the GBR of 1 May 2018. It contains 2.8 million Dutch legal units and 2.0 million of them are related to an enterprise, see Table 1. The remaining 0.8 million legal units are not related to an enterprise, which usually concerns legal units of which the corresponding enterprise has ceased its activities. Sometime it concerns legal units that belong to a foreign enterprise, but are present in the GBR because they appear in one of our tax systems.

For about 0.62 million of the 2.0 million legal units, there is a URL registered from the COC (see the second column of Table 1), which is slightly less than one third of the total number of legal units. Not all of those URLs are unique: for some of the enterprises that consist of multiple legal units, the same URL is registered for all underlying legal units. For other enterprises with multiple legal units, URLs are uniquely linked to specific legal units underlying the enterprise. It is unclear yet whether this has simply been reported this way to the COC or whether there is another reason for this difference.

As explained in the introduction, we limit ourselves to legal units that are one-to-one related to an enterprise (in other words enterprises which are composed of one legal unit). Furthermore, we link one URL per legal unit. The GBR contained 1.68 million legal units that are one-to-one relate to an enterprise, of which 561 thousand had a URL from the COC and 118 thousand had a URL obtained from DP data using exact

linkage, that was not yet available from the COC; see the penultimate column in Table 1. Furthermore, we limit ourselves to enterprises with 10 or more employees. We obtained 62 thousand legal units that were one-to-one related to an enterprise with 10 or more employees, see the final column of Table 1. Of those 62 thousand legal units, there were more than 35 thousand legal units with an URL from the COC.

Table 1. The total number of legal units and those from which we know that they have a website ('website+') subdivided by source of the URL. The final column refers to the target population.

	Legal units	Legal units related to an enterprise	Legal units 1: 1 related to an enterprise	Legal units 1:1 related to an enterprise ≥ 10 EMP
Total	2 832 410	2 007 309	1 685 054	62 235
'Website+' legal units	NA	801 784	697 354	43 387
'Website+' legal units, from COC	752 544	627 892	560 819	35 154
'Website+' legal units, from DP, not from COC: exact linkage on LU ID-number	172 729	147 971	118 273	7 895
'Website+' legal units, from DP, not from COC: probabilistic linkage	NA	25 921	18 262	438

NA = not available, EMP = employees

3.1.1 Link DP data to the GBR

In order to have more URLs, we linked URLs from DP. DP scrapes websites concerning domains from a large number of countries on a monthly basis. We obtained data from April 2018. From those websites, DP provided identification variables such as the legal unit identification number, the business name, email address, phone number and so on. Unfortunately, there was a considerable amount of missingness for those variables in the DP data, see Figure 2. DP selected URLs that were likely to belong to Dutch businesses by checking whether the values of the identification variables matched with those known from registered Dutch business lists. They restricted the URLs to relevant top-level domains such as '.nl', '.com' and '.eu'.

The DP data consisted of 3.2 million unique URLs. Those URLs were linked to the legal units in the GBR with that are related to an enterprise, in three steps (Heemann, 2018). In the first step, URLs of the DP data were linked, by exact linkage on URL, to the 628 thousand legal units with a URL in the GBR that were obtained from the COC. This resulted in 307 thousand linked unique URLs. Second, we linked the URLs in the DP data by using the legal unit identification number (if present). The legal unit identification number was available for slightly more than 25 per cent of the DP URLs (Figure 2). In this way, we obtained 148 thousand legal units with a URL from DP that

did not yet have a URL from the COC, see the fourth row × second column in Table 1. Thirdly, URLs of the DP data were linked to the GBR, by using a combination of probabilistic linkage with machine learning. This way an additional 26 thousand legal units with a URL were obtained, see ultimate row × second column in Table 1. In total, we obtained 802 thousand legal units with a URL (see the second row × second column in Table 1), which corresponds to approximately 40 per cent of the total number of legal units with an enterprise.

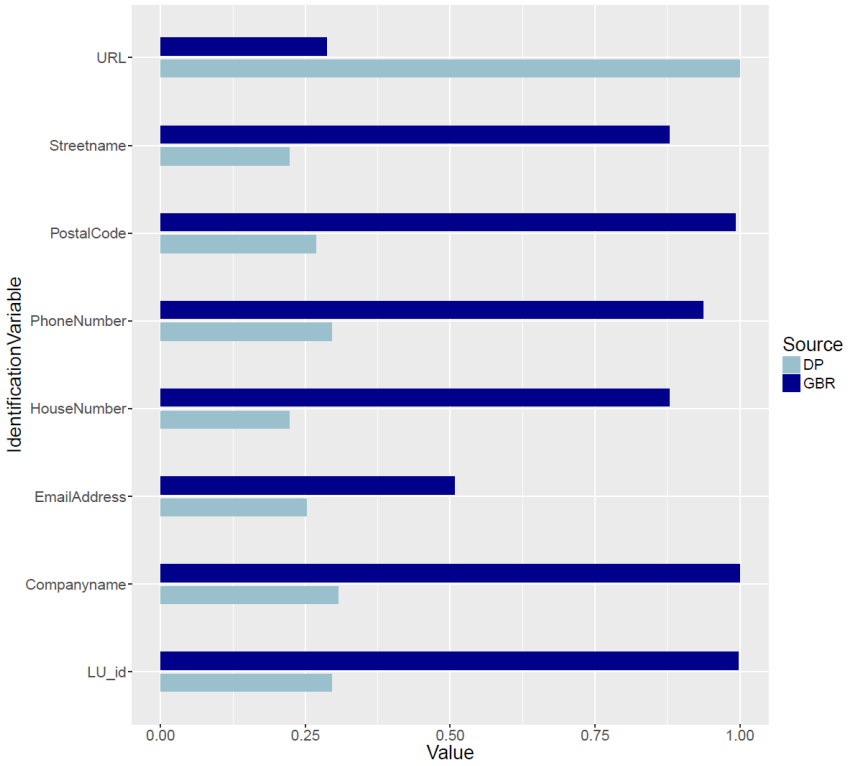

Figure 2. The fraction of cases with available data for eight identifying variables that are both present in the DP data and in the GBR.

From the total of 148 thousand legal units related to an enterprise with a URL from DP that was not already obtained from COC, there were 118 thousand with one-to-one related to an enterprise and 7.9 thousand of them referred to an enterprise of 10 or more employees (see the third row of Table 1). Similarly, for the probabilistically linked URLs the corresponding numbers were 26 thousand, 18 thousand and 438 legal units (see the fourth row of Table 1). In total there were slightly more than 43 thousand 'website+' legal units in the target population (second row × final column of Table 1).

3.1.2 Link the ICT survey to the GBR

In the present study, we selected as 'website-' legal units those legal units that fulfilled two criteria: they do not belong to the 'website+' legal units and the corresponding enterprise in the ICT survey responded to have no URL, based on the ICT survey of 2017. This ICT survey had a total sample size of 10 732 enterprises with

8 909 respondents, of which 845 responded to have no URL. We linked those 845 respondents to the legal units with a one-to-one relationship to an enterprise and obtained 279 legal units that fulfilled the two criteria.

3.2 Division over the NACE codes

The division of the number of legal units in the target population by economic sector is given in Table 2. This is further partitioned into the 'website+' and 'website -' legal units and by source of the URLs. The table shows that the total number of URLs from COC in the target population is larger than those from DP. DP added 7.9 thousand to the 35 thousand URLs had been obtained from COC already.

Table 2. Number of legal units in the target population by economic sector (letter of the NACE code) divided over 'website+' and 'website -' legal units.

Economic sector	Total	'website+' legal units			No URL in ICT survey	'website-' legal units
		URL from COC	URL from DP	URL from DP not from COC		according to both criteria
A	1963	601	473	364	1	0
B	69	40	29	8	0	0
C	8060	4840	3354	1199	124	33
D	81	27	22	13	6	4
E	297	173	119	32	5	2
F	4760	2607	1715	899	56	15
G	13463	7190	5030	1528	149	48
H	3410	1568	1112	526	81	40
I	4709	2132	1803	764	39	18
J	3033	2051	1292	198	66	16
K	1113	570	393	114	14	6
L	747	455	325	79	10	3
M	5791	3686	2785	619	105	30
N	4862	2626	1752	662	152	56
O	555	517	506	11	0	0
P	1937	1453	1298	117	1	0
Q	4599	2995	2474	449	33	7
R	1423	872	716	148	0	0
S	1363	751	643	165	3	1
T	0	0	0	0	0	0
U	0	0	0	0	0	0
Total	62235	35154	25841	7895	845	279

The proportion of 'website+' legal units varied considerably by economic sector. The highest proportions of legal units with a website were found in sector O 'public

administration and defense; compulsory social security' and in sector P 'education'. The smallest proportions were found in sector A 'agriculture, forestry and fishing' and D 'electricity, gas, steam and air conditioning supply'.

The proportion of 'website-' legal units was very small. One of the reasons is that the ICT survey concerns a sample of the population. The largest absolute numbers of 'website-' legal units were found in the economic sectors G 'wholesale and retail trade; repair of motor vehicles and motorcycles', H 'transportation and storage' and N 'administrative and support service activities'.

Note that there is a large group of legal units that are neither attributed to the 'website+' legal units nor to the 'website-' legal units. This concerns legal units of which we do not know yet whether they have a website or not.

3.3 Select units from the target population for the labelled set

From the target population data, summarised in Table 1, we sampled units for the labelled set as follows. From the 'website+' legal units we randomly selected 1501 units with a URL from COC and 1499 units with a URL from DP. Furthermore, we selected all 279 'website-' legal units.

The units for the labelled set were split randomly such that 70% was put in the training set and 30% in the test set. Let the pair (x, y) denote the number of units in the trainingsset and in the test set respectively. This pair was (1037, 464) for the 'website+' legal units from COC, (1061, 438) for the 'website+' legal units from DP, and (197, 82) for the 'website-' legal units.

A part of the process of sampling the legal units is that we checked the validity of the URLs of the 'website+' legal, by visiting the websites (automated). We found that for 191 of the 3000 legal units the websites resulted in http error codes 4xx (Client error) or 5xx (Server error). For all legal units with such an error, we kept the original URL in the labelled set. We did not drop those units, since we are not sure why we obtained that error, it might be just a temporary technical error. Furthermore, some of the URLs were redirected to another address. For those units we kept both the original as well as the redirect URL, since both URLs may be retrieved when one searches for URLs of legal units.

4. URL search

The next step 'URL search' consists of the sub steps:

- select contact information, section 4.1
- search URLs on internet, section 4.2
- analyse the usefulness of the different queries. section 4.3

4.1 Select contact information

The GBR information from the labelled set contains a large number of variables, of which we used a limited number in the present study. From the enterprise, we used the main economic activity code and the number of employees. That code was used in the machine learning model and the number of employees was used to select the target population. From the legal unit we used its identification number for the linkage with the DP data.
Furthermore, from the legal unit we used the following contact information:

- legal name
- trade name

Additionally, we used address information from an establishment, that is a local unit, underlying the legal units. A legal unit consists of one or more establishments. We selected the address of one establishment per legal unit, using the following approach. First we selected the main establishment. A business determines itself, which establishment they consider to be their main establishment. For those legal units for which no unique establishment was obtained yet, we selected the so called contact person. The contact person is the establishment where the top of the ownership of an enterprise or enterprise group is located. From the selected establishments we used the variables:

- street name
- house number
- municipality
- postal code
- phone number
- post office box number.

4.2 Search URLs on internet

We formulated six search queries containing contact information of the legal unit that were automatically applied via the Google custom search engine API, see Table 3. In query number 5 we add 'inanchor: contact', this means that one searches for an anchor text that contains the word 'contact'. An anchor text is a text behind which a hyperlink is hidden. We will explain the meaning of the component '-site …' within the search queries at the end of this section.

After applying the search queries for all of the sampled legal units we store the first 10 search results per query. Note that sometimes fewer than 10 search results are returned, so 10 is the maximum number that is stored. In total we store up to 60 search results per legal unit. Each search result consists of the texts within four so-called search locations, namely the title (purple text in Figure 3), the URL (green text in Figure 3), the snippet (black text of Figure 3) and PageMap. PageMap are invisible blocks of JSON that summarise the webpage and contains metadata about the webpage.

Table 3. Search query types.

Query type	Description
0	(legal name or trade name) + 'contact' + '-site:..'
1	(legal name or trade name) + street name + 'contact' + '-site:..'
2	(legal name or trade name) + postal code + 'contact'+ '-site:..'
3	Street name + house number + municipality+ '-site:..'
4	(legal name or trade name) + '-site:..'
5	(legal name or trade name) + 'inanchor:contact'+ '-site:..'

Figure 3. Example of two search results

When applying our approach for the first time we found that the URL of our search results often concerned so-called business directory websites. Business directory websites are websites that contains listings of businesses such as the yellow page websites and phonebook websites. Since we aim to focus on (specific) business websites, we consider links to those sites as mismatches. In each search query applied we excluded a list of directory websites. This list was constructed from preliminary results and consisted of domains that were found more than 60 times in the collection of search results for the first 1000 legal units within our sample. So if a domain occurred more than 60 times we were sure that they referred to at least two different legal units. Finally, after we searched for all legal units in the labelled set, we again removed the search results referring to the same domain more than 60 times. If the same domain was found for two or more legal units, the search results linking to these domains were also removed.

https://	video.	google.	co.	uk/	videoplay	? Docid =	724
1	2	3	4	5	6	7	8

Figure 4. Terminology on structure and components of a URL. 1= protocol, 2 = subdomain, 3: domain, 4: second-level domain; 5: top-level domain, 6: path, 7: parameter, 8: parameter value. Many websites use 'www' as their subdomain to indicate 'world wide web' but this is not mandatory.

Finally, from each URL we derived the 'domain' part, see Figure 4 for the different parts of an URL. We will consider a searched URL for a legal unit to be correct when its domain is identical to the domain of the URL in the labelled set for the corresponding legal unit.

4.3 Analysis of usefulness of different queries

For a limited number of the legal units for our labelled set, we did not retrieve any search results, see Table 4. For the legal units with a URL from the COC and those with a URL from the DP data, no search results were obtained due to a technical error. This technical error was either due to the Google API or due to a network error. For the five legal units with no URL according to the ICT survey there was no technical error. For the remainder of this paper, we excluded all legal units for which no search results were obtained. We did so, because even in case of the legal units with no URL according to the ICT survey, we are not entirely sure that they really have no website or that our search queries were incorrectly unable to find a website.

Table 4. Retrieval of search results per subpopulation.

Subpopulation	Number of legal units per subpopulation	Legal units for which search results were obtained	Legal units for which no search results were obtained
URL from COC	1501	1409	92
URL from DP	1499	1408	91
No URL according to ICT	279	274	5

We analysed the effectiveness of the different queries in finding different domains as follows. First we counted, for each query type, the number of search results where the retrieved domains corresponded to the known domain, summed over the total of 3000 legal units with a known domain in the labelled set. Recall that each search query returns at most 10 search results, and each search result may return the same domain multiple times. Different search results that share the same domain will normally have different URLs, for instance they can refer to different pages of the same domain.

The result of this analysis is given in Figure 5. The top row of Figure 5, '0 correct domains in result set' stands for the situation where we do retrieve search results, but all retrieved domains are incorrect. The second row stands for the number of cases within the search results per query where exactly one time the correct domain is retrieved, the third row for the number of cases where the correct domain is retrieved two times within the search results and so on. The sum over each column stands for the number of legal units for which search results were obtained. This sum was largest for query type 0 (2751) and smallest for query type 1 (2272). In all cases this sum was smaller than the total number of legal units for which at least some search results were obtained, namely 1409 + 1408 = 2817, see Table 4. Thus, for query type 0, for 2817 - 2751 = 66 legal units no search results were found with query type 0, where at least one search result was found for that legal unit with one of the other query types. Furthermore, for query type 1, for 2817 - 2272 = 445 legal units no search results were found for query type 1 whereas at least one search result was found for that legal unit with one of the other query types.

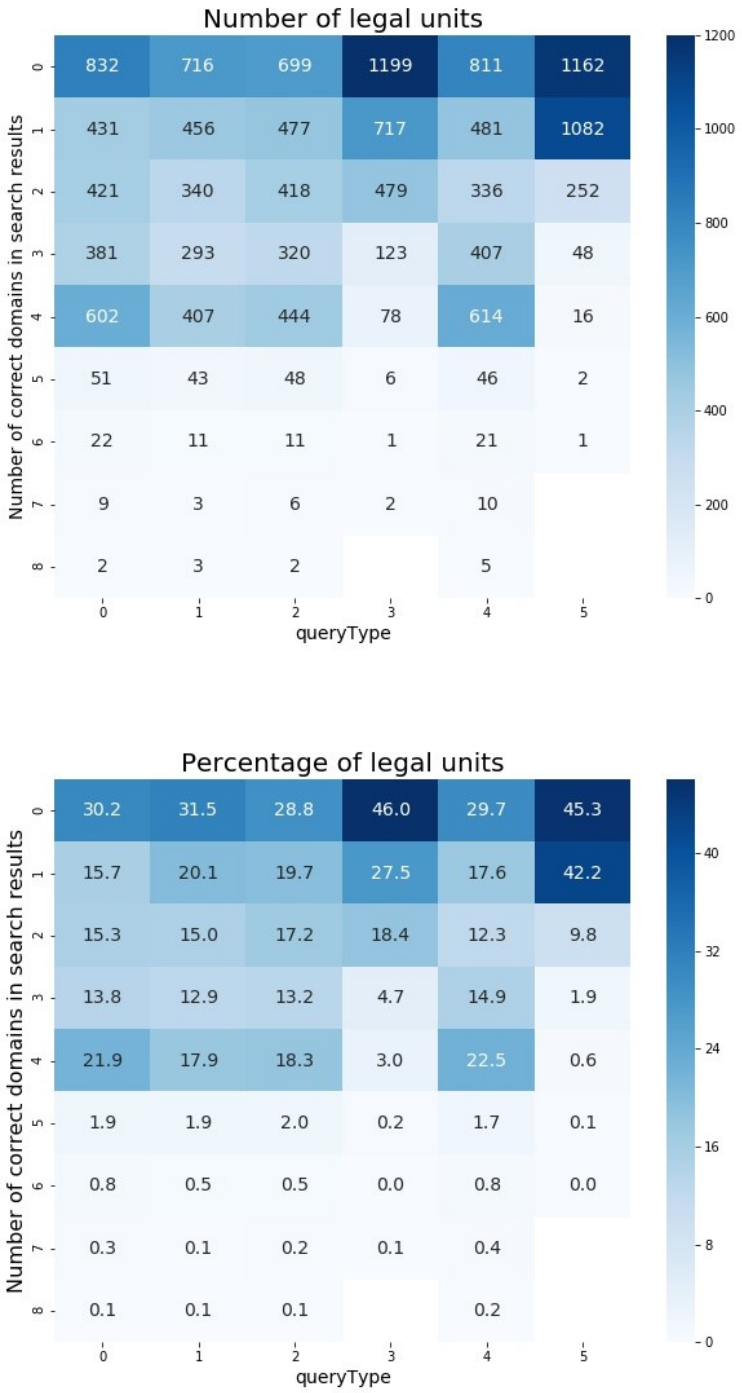

Figure 5. Number (top panel) and percentage (bottom panel) of search results where the returned domain corresponds to the known domain, for 3000 'website+' legal units.

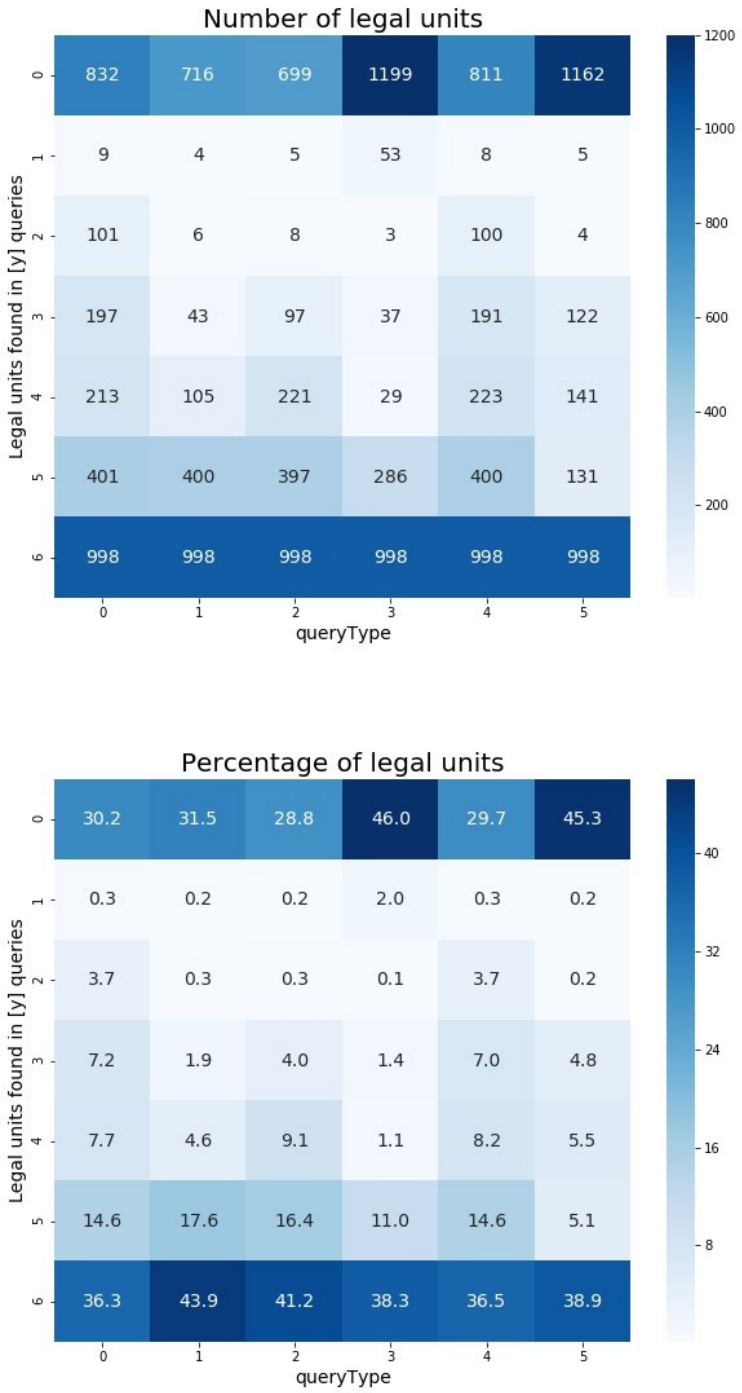

Figure 6 Number (top panel) and percentage (bottom panel) of legal units for which the correct domain is found in {0, 1, …6 } queries versus query type.

In less than 5 per cent of the cases the correct domain was found five times or more for a given same query type. For query type 3 and 5, the distribution of the number of correct domains per query types was far more skewed than for the other query types. For query type 3 and 5 in about 45% of the cases where a search result was obtained the correct domain was not found. Furthermore, in 27 % (type 3) and 42% (type 5) of the cases the correct domain was found only once within a query type. The percentage of cases with a correct domain sharply dropped for larger numbers of correct domains for query type 3 and 5.

Next, we analysed to what extent the different query types were supplementary to each other in terms of their effectiveness to find the correct domain. We counted the number of queries in which the correct domain of the legal unit was found, and plotted this against the corresponding query types in which those correct domains were found, see Figure 6. For instance, the second row in Figure 6 shows the number of legal units for which the correct domain was only found in one query type. In most of the cases (53 times) this concerned query type 3. In just a few cases, this concerned the other query types. Thus, although query type 3 by itself, was not so effective in terms of the total number of correct domains that were found (Figure 5), the domains that were found were supplementary to the other query types. The third row in Figure 6 shows the number of legal units for which the correct domain was found in two query types. For most of the cases this concerned query type 0 and 4. This suggests that there was a large overlap in the domains that were returned from those two query types. Also, Figure 5 shows that the total number of correctly found domains was nearly the same for query type 0 and 4. For future work, we may decide to no longer include query type 4.

5. Investigate the feature set

We will now refer to the retrieved domains per target legal unit as the candidate domains for that legal unit, that is they are candidates to be the correct domain. We will use two types of features for the machine learning model that estimates the probability that a candidate domain is the correct domain of the targeted legal unit:

- features that express the level of agreement between a contact variable of the legal unit and text in a specific location of the search result. Thus, to what extent does the address of the legal unit agrees with the text found in the search results? We refer these to this type of features as 'agreement features'.
- features that quantify how well the google search engine is able to find domains for the legal unit. We refer to this type of features as 'search engine features'.

The step to investigate the feature set of the model consists of the following substeps:

- derive agreement features, section 5.1
- derive search engine features, section 5.2
- derive the labels of the labelled set, section 5.3

- determine the relative importance of the features, section 5.4
- explore the effect of the features on model performance. section 5.5

5.1 Derive agreement features

In the current section we describe how we derived the agreement features. An overview of the agreement features is given in Table 5.

Table 5. Overview of the agreement features (see text).

Identifying variable	Search location	Label (xxx = 'min', 'max', 'mean')
Legal name	title	eqTitleLegalName_xxx
	snippet	eqSnippetLegalName_ xxx
Trade name	title	eqTitleTradeName_ xxx
	snippet	eqSnippetTradeName_ xxx
Locality	title	eqTitleLocality_ xxx
	snippet	eqSnippetLocality_ xxx
	PageMap	eqPagemapLocality_ xxx
Address	title	eqTitleAddress_ xxx
	snippet	eqSnippetAddress_ xxx
	PageMap	eqPagemapAddress_ xxx
Postal code	title	eqTitlePostalcode_ xxx
	snippet	eqSnippetPostalcode_ xxx
	PageMap	eqPagemapPostalcode_ xxx

The first step to derive the agreement features was to process the identifying variables of the GBR and the texts of the search results. The street names which are stored in the GBR are abbreviated. For instance 'laan' (English: avenue) is abbreviated to 'ln', 'straat' (English: street) to 'str', and 'plein' (English: square) to 'pln'. We used a set of rules to restore the full street names. Furthermore, the legal and trade names contained abbreviations like "bv", "nv", "incorp." and so on. We used a set of rules to drop those parts. Next, the texts of the identifying variables were tokenised into words. The texts of the search results were also tokenised into words and punctuation marks were dropped.

After the tokenisation, for each word of the identifying variables of the GBR we quantified the agreement with each word in the search results by using the Jaro-Winkler similarity (see below). For each identifying variable we selected the maximum similarity for each of the applicable search locations (title, URL, snippet and PageMap) and used that maximum as the value of the feature. For an address or a name which may consist of two or more words, we first computed the maximum similarity for each word separately and then we took the maximum over all corresponding words. We realise that in some cases this word-by-word approach may have overestimated the true similarity; this is a point to be improved in future.

The Jaro-Winkler similarity is based on an improvement by Winkler (1990) of a method proposed by Jaro (1989). This similarity measure has especially been designed as a step in the linkage of data sets based on names, addresses and so on. This measure accounts for often occurring errors (typos, transpositions of characters that are close together) when entering names and addresses and so on.

Since we are ultimately interested in finding the domains of legal units, we summarised the above derived Jaro-Winkler similarities at domain level. Recall that within the set of (at most 60) search results, a domain can be found multiple times. The same search query as well as different search queries can yield URLs which share the same domain. Let $s_{j\ell m}$ stand for an estimated Jaro-Winkler similarity of identifying variable j, at search location ℓ and search result m. Further, let $\mathcal{M}_k$ be the collection of search results (including duplicate if any) that share the same domain k. For a given identifying variable j and search location j we can have different values $s_{j\ell m}$ for different search results m that share the same domain k ($m \in \mathcal{M}_k$). For instance, Table 7 shows a fictional example of three search results for a legal unit with the legal name 'Bert's Barbershop' that share the same domain. The three search results lead to three different titles. For each title, the Jaro-Winkler similarity between each pair {title word, legal name word} is computed and its maximum is taken to be $s_{j\ell m}$, see the pen ultimate column of Table 6. The final column shows which word pair had the highest Jaro-Winkler similarity. Surprisingly, we found that the Jaro-Winkler similarity between words that share just a few characters to be rather high. We obtained Jaro-Winkler similarities of 0.55, 1.00 and 0.60. We summarised these different values for the same domain k by taking the minimum, the maximum and the average value over $s_{j\ell m}$, with $m \in \mathcal{M}_k$; we denote them by $s_{j\ell k}^{(\min)}$, $s_{j\ell k}^{(\max)}$ and $s_{j\ell k}^{(\text{avg})}$ respectively. In our example, $s_{j\ell k}^{(\min)} = 0.55$, $s_{j\ell k}^{(\max)} = 1.00$ and $s_{j\ell k}^{(\text{avg})} = 0.72$. These minimum, the maximum and the average values are the agreement features that were used in the machine learning model.

Table 6. Fictional example of the Jaro-Winkler similarity for three search results for a unit with legal name 'Bert's Barbershop' that share the same domain (see text).

result (m)	domain	domain (k)	title	Jaro-Winkler ($s_{j\ell m}$)	Word pair with highest JW score
1	bb.com/main	bb	Welcome to our shop in Sesamestreet	0.55	{bert, sesamestreet}
2	bb.com/contact	bb	Contact: Bert's Barbershop, 26th Sesamestreet	1.00	{bert, bert}
3	bb.com/prices	bb	Special prices cutting and shaving	0.60	{barbershop, prices}

5.2 Derive search engine features

We derived two search engine features. The first search engine feature captures the rank position of a domain k within the retrieved search results for a given target legal unit. Let the rank of a search result m, irrespective of the underlying domain, be denoted by r_m. A search result which is ranked on top is given a value of 10 ($r_m = 10$), the next one is given a value of 9 and so on, up to a value of 1. Recall that for each legal unit six different queries were run, so each value {10, 9, 8, …, 1} can occur six times but the actual number of search results per query varies and can be smaller than 10. Therefore we wanted to normalise the rank position, such that the results are comparable over the different legal units. We normalised the rank position as follows. Let the set of domains for a given legal unit be denoted by $\mathcal{K}$. The normalised rank position of domain k, denoted by ρ_k, for a given legal unit was computed as:

$$\rho_k = \sum_{m \in \mathcal{M}_k} r_m \Big/ \sum_{\ell \in \mathcal{K}} \sum_{m \in \mathcal{M}_\ell} r_m \qquad (1)$$

The second search engine feature summarises the frequency that a domain k is found within the search results for a given legal unit relative to the average frequency of finding a domain for that legal unit. Let $n_k = |\mathcal{M}_k|$ be the number of times that domain k is found for a given legal unit. We then compute the difference between n_k and the average number of times that a domain is found for that legal unit. This average number, $\bar{n}_k$, is given by $\bar{n}_k = \frac{1}{|\mathcal{K}|}\sum_{k \in \mathcal{K}} n_k$. So a positive difference implies that domain k is found more frequently than the average, and a negative difference implies that domain k is found less frequently. Because the distribution of n_k will differ for the different legal units, we normalised this difference, by dividing it by the standard deviation of n_k. Let σ_{n_k} denote this standard deviation, which is estimated by $\hat{\sigma}_{n_k} = \sqrt{\frac{1}{|\mathcal{K}|-1}\sum_{k \in \mathcal{K}}(n_k - \bar{n}_k)^2}$. The normalised frequency for domain k for a given legal unit, denoted by z_k was computed as:

$$z_k = (n_k - \bar{n}_k)/\hat{\sigma}_{n_k} \qquad (2)$$

Note that, the agreement features and the search engine features are (approximately) on the same scale. The agreement features and the normalised rank position have values in the range [0, 1]. The normalised frequency has a mean of 0 and most of its values will be between -2 and 2. This way we avoided that differences in scale caused certain features to dominate in importance for the model.

5.3 Derive the labels of the labelled set

In section 3.3 we described the set of legal units for the labelled set. Now we describe how their labels were derived. We used a label with two categories namely 'True' and 'False'. From the search results per legal unit we obtained a set of candidate domains. For the 'website+' legal units the label is 'True' if a candidate

domain equals the correct domain (including the domain of the redirected website if any) of that legal unit and the label is 'False' otherwise. For the 'website-' legal units any retrieved candidate domain has the label 'False'.

Recall from section 4.3 that throughout the analysis in the current paper, legal units without any retrieved search results for all six queries were left out of any further analysis. This was done because we cannot be sure whether those legal units in reality have no website or whether we were unable to find their website. Therefore, we did not apply label 'True' to the 'website-' legal units.

5.4 Determine the relative importance of the features

We analysed the relative importance of the different features for predicting whether a candidate domain was correct or not. All of the computed measures were based on the values of the features (see section 5.1 and 0) in the collection of all candidate domains for all legal units in the trainingsset. Part of the measures also needed the values of the labels in the trainingsset.

The first method we used was to analyse the similarity among the features, using Spearmans rank correlation. The values within each feature were ordered from small to large and the first value (the smallest one) was given rank number 1, the second one rank number 2 and so on. Spearmans rank correlation between two features is then given as the Pearson correlation coefficient of those rank numbers. A distance matrix was computed between each of the features, with 1 minus Spearmans rank correlation as the elements of the matrix. Next, this distance matrix was used in a so called hierarchical clustering method (bottom up). In this clustering method the starting point is that each individual feature forms a separate cluster. Next, the two clusters with the smallest distance are joined into a new (combined) cluster. Then, the distance matrix is recomputed such that distance between a cluster with the new (combined) cluster is the average of the two underlying distances. Thereafter, the two clusters with the smallest distance are joined. This procedure is repeated until there is just one single cluster left. The result of this collapsing procedure is plotted in a so called dendrogram, where the height of a join corresponds with the distance at which the clusters were joined.

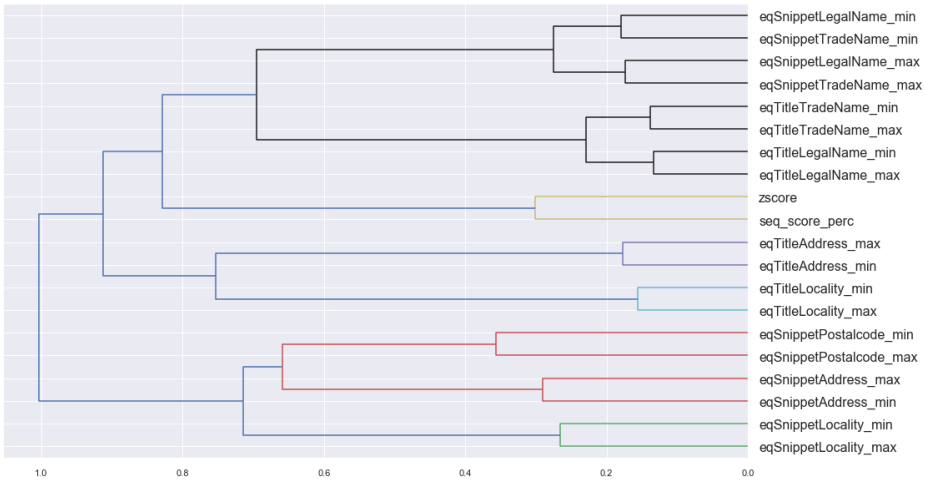

Figure 7. Dendrogram for hierarchical clustering based on Spearman rank correlation among the top 20 most important features. The labels of the agreement features are given in Table 1; the label 'seq_score_prec' equals the normalised rank position and 'zscore' equals the normalised frequency.

We found that Spearman rank correlations of the feature $s_{j\ell k}^{(\min)}$, $s_{j\ell k}^{(\max)}$ and $s_{j\ell k}^{(\text{avg})}$ were often very close together (not shown). We therefore decided to keep only the minimum and the maximum variant for further analysis since they were most far apart.

The dendrogram of the hierarchical clustering of the top-20 (as given in Figure 8) most important features showed first of all that the 'min' and 'max' variant of an agreement feature variable were often relatively similar, as could be expected (see Figure 7). Next, the legal name and trade name were relatively similar for the title and the snippet. Thereafter, the two search engine features were most similar.

The second method we used was to analyse some measure of association between the features and the labels. We used the following three methods:

- Random forest feature importance
- Pearson correlation
- Information gain

Random forest feature importance A random forest model is based on a set decision trees. A decision tree splits the output values that it tries to predict on a impurity measure (Gini or entropy, see Hastie et al, 2009). We used entropy as impurity measure. The smaller the impurity the more units belong to the same label category within a node. Thus, our case, a small impurity, implies that a large proportion of units belong to either label 'True' or label 'False'. The random forest feature importance is defined as the total decrease in node impurity (weighted by the probability of reaching that node) averaged over all trees in the forest.

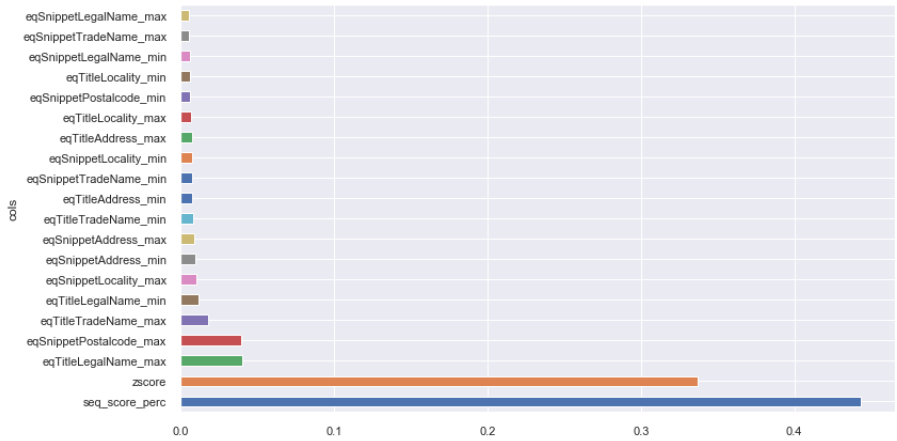

Figure 8. Random forest feature importance results (top 20).

Results of the random forest feature importance (see Figure 8) show that the feature ρ_k ('seq_score_perc') has the largest feature importance, followed by z_k ('zscore'). The feature importance of the other feature were much smaller. Of those, the two features with the largest feature importance were the Jaro-Winkler similarity for the legal name in the title (maximum) and the Jaro-Winkler similarity on the postal code in the snippet (maximum).

Pearson correlation Let u_k be the general notation for a feature at domain k that is used in the text minings model and let v_k denote the label for candidate domain k. Pearson's correlation coefficient was computed between u_k and v_k. Pearson's correlation coefficient, denoted by $\mathrm{Cor}(u_k, v_k)$, is given by $\mathrm{Cor}(u_k, v_k) = \mathrm{Cov}(u_k, v_k)/\sqrt{\mathrm{Var}(u_k)\mathrm{Var}(v_k)}$, where Cov stands for the covariance and Var for the variance.

Information gain. The information gain, also referred to as the Kullback–Leibler divergence, the relative entropy and the expected mutual information, is a measure of how one probability distribution is different from a second, reference probability distribution. Its exact definition can be found in Cover and Thomas (1991). An information gain of 0 implies that feature u_k is not explanatory for the label v_k, otherwise the information gain is larger than 0.

The scores for the association between the features and the labels for the Feature importance, the Pearson correlation and the Information gain are given in Table 7. It shows that the features can be divided into three groups. The first group, with the highest scores, are search engine features: ρ_k and z_k. The second group concerns the features that were derived from PageMap. Those features have the lowest scores. The third group concerns the remaining features, derived from the title or the snippet. Within each of those three groups, the relative ordering of the other features depended on the score function. Furthermore, concerning the third group, the 'maximum' variant had often a higher ranking than the 'minimum' variant, exceptions being the snippet trade name, snippet legal name, snippet locality and title locality.

Table 7. Association between the features and the labels for three measures.

Feature	Feature importance	Pearson correlation	Information gain
seq_score_perc	0.414	0.713	0.264
zscore	0.336	0.733	0.226
eqTitleLegalName_max	0.041	0.273	0.091
eqSnippetPostalcode_max	0.040	0.312	0.086
eqTitleTradeName_max	0.016	0.250	0.086
eqSnippetAddress_max	0.015	0.228	0.070
eqSnippetAddress_min	0.013	0.171	0.054
eqTitleLegalName_min	0.012	0.089	0.048
eqSnippetLocality_max	0.011	0.241	0.071
eqSnippetLocality_min	0.011	0.142	0.038
eqTitleTradeName_min	0.010	0.076	0.045
eqSnippetPostalcode_min	0.009	0.174	0.038
eqTitleAddress_max	0.009	0.103	0.035
eqSnippetTradeName_min	0.008	0.091	0.033
eqSnippetLegalName_min	0.008	0.075	0.032
eqTitleLocality_min	0.008	0.102	0.031
eqSnippetTradeName_max	0.008	0.177	0.058
eqTitleAddress_min	0.008	0.121	0.039
eqTitleLocality_max	0.008	0.126	0.035
eqSnippetLegalName_max	0.007	0.206	0.067
eqTitlePostalcode_max	0.005	0.083	0.009
eqTitlePostalcode_min	0.005	0.129	0.018
eqPagemapLocality_max	0.000	0.002	0.000
eqPagemapLocality_min	0.000	0.023	0.001
eqPagemapAddress_max	0.000	0.001	0.000
eqPagemapPostalcode_max	0.000	0.006	0.000
eqPagemapAddress_min	0.000	0.021	0.000
eqPagemapPostalcode_min	0.000	0.022	0.000

5.5 Explore the effect of the features on model performance

We analysed the effect of different numbers of features on model performance. We computed the impact of adding one feature to a random forest model on the performance of the model, as follows:

For score y = { 'Feature importance', 'Pearson correlation', Information gain'}:
For $x = 1, 2$ to Q: (where Q stands for the total number of features)
Select the top x features according to the ordering score y. For this set of features select the hyperparameters of the machine learning model using a five-fold cross-validation on the training set, using Matthews Correlation Coefficient as the metric to be

optimised. Next, we computed the F1 score of this model on the test set, while selecting the URL with the largest probability to be correct.

The metrics Matthews Correlation Coefficient and F1 score are explained in section 8. We computed the F1 score for the test set for two situations. In the first situation we used the labels of all candidate domains in the retrieved search results per legal unit. In the second situation we selected the top-one domain per target legal unit, namely the candidate domain with the highest estimated probability that the domain is correct. We refer to the first situation as the prediction level 'all candidate domains' and to the second as the prediction level 'top-one domain'.

As expected, the F1 score of a random forest model increased with the number of features included in the model. This holds both for the prediction level 'all candidate domains' (Figure 9) as well as for the 'top-one domain' (Figure 10). The actual F1 scores varied slightly for the three different score functions but the main pattern remained the same. For the 'all candidate domains' level the F1 score increased from 0.957 for a single feature to about 0.967 for about six features and then remained stable, so the increase was relatively small. For the 'top-one domain' level the score increased from about 0.74 for a single feature to slightly above 0.79 for about 5 to 6 features and slightly decreased thereafter. Based on these results, we decided to keep all features in the model because we expected that keeping the agreement features in the model makes the results more generalisable in future. One can argue that the additional value of the 'PageMap'-features is very small, i.e. the final six features, and that those features can be dropped from the model in future.

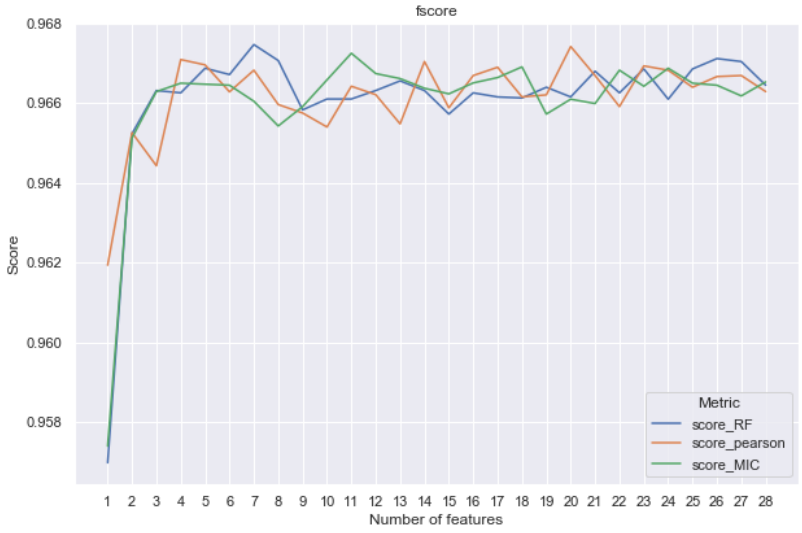

Figure 9. Effect of the number of features on the F1 score of a random forest model at the prediction level 'all candidate domains'. The metrics refer to Random forest feature importance ('score_RF'), Pearson correlation ('score_pearson') and Information gain ('score_MIC').

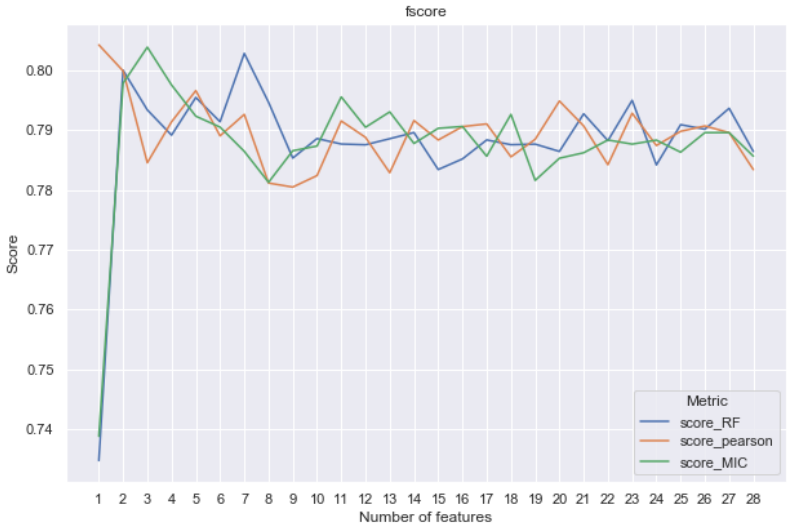

Figure 10. Effect of the number of features on the F1 score of a random forest model at the prediction level 'top-one domain'. See also the title of Figure 9.

6. Model selection and testing

Given the selected set of features, as determined in the previous step, we finally perform two steps:

- select a machine learning model, section 6.1
- analyse the quality of the model predictions. section 6.2

6.1 Select a machine learning model

We compared the performance of the Gaussian Naïve Bayes classifier (NB), a Support Vector machine (SVM) and a Random Forest (RF) model. The performance measures F1, precision and recall are explained in section 8. The macro average (macro avg) of a performance score is the average of the scores per category, with an equal weight for each category. The micro average (micro avg) value of a performance score is obtained by using the number of individual cases per category, see section 8.

The hyperparameters of each model were determined by a five-fold cross validation, using Matthews Correlation Coefficient as the performance measure to be optimised. The hyper parameter settings of the three models are given in Table 8. When the values are displayed as '[ ]' then a range of values was used and their ultimate values was determined by a grid search in the cross-validation procedure.

To understand the performance of the models it is important to recall that all results are scored with respect to the candidate domains in the retrieved search results. In the confusion matrix we distinguish the label category from the predicted category.

The label category denotes whether the candidate domain equals the correct domain or not. The predicted category equals 'True' when the *predicted probability*[1] that the candidate domain is the correct one is ≥ 0.5, and it is 'False' otherwise. See also Table 13 in section 8 for an example of how the domains are scored.

Table 8. Parameter settings of the machine learning models. The selected hyperparameters within a range, is given in **bold**.

Model	Hyperparameters	Range Values
NB	Smoothing:	$1.0\ 10^{-9}$
	class prior:	learned from the data
SVM	Kernel:	radial
	C:	[1,2,3,**5**,7,10]
	Gamma:	[0.01, 0.02 , 0.03, 0.05, 0.10, 0.3, **0.5**]
RF	Number of trees in the forest	100
	Criterion for quality of split	[gini, **entropy**]
	Maximum depth of the tree	[2,5,10,20,**None**]
	Minimum number of samples required to split an internal node	[**10**, 20, 40, 50]
	Minimum number of samples required to be at a leaf node	[**1**, 5, 10, 20, 30]
	Number of features to consider when looking for best split	[**Q**, $\sqrt{Q}$, $\log_2(Q)$, $0.5\ Q$][1]
	Minimum weighted fraction of the sum total of weights required to be at a leaf node	0
	Maximum number of leaf nodes	No limitation
	Minimum impurity decrease at a split	0
	Whether bootstrap samples are used to grow the tree	True
	Out of bag samples are used to estimate generalisation accuracy	False
	Verbosity for fitting and predicting	0
	Class weights	None

[1] Q stands for the total number of features.

The three models clearly showed a better performance for the category 'False' than for the category 'True' for the prediction level 'all candidate domains', see Table 9. For all models, the F1 score for the category 'False' was above 0.97 whereas the F1 score for the category 'True' was 0.71 (NB), 0.78 (RF) and 0.77 (SVM). The most relevant results are the model performances at the prediction level 'top-one domain', shown in Table 10, since we aim to find one domain per legal unit. In contrast to the

[1] The probability of the SVM model was estimated using Platt scaling.

predictions for 'all candidate domains', the performance for the 'top-one domain' of the category 'True' was better than for the category 'False', the actual value depending on the model. The model performances for SVM and RF were nearly the same, whereas the performance for NB was clearly worse, especially for the category 'False'. For the latter category, NB yielded an F1 score of 0.62, RF of 0.73 and SVM of 0.77. For the category 'True' differences between the models were much smaller, with F1 values of 0.82 (NB), 0.83 (RF) and 0.84 (SVM). Because the performance of SVM was slightly better than that of RF, we decided to use the SVM model as our final URL retrieval model.

Table 9. Performance of the models at the prediction level 'all candidate domains'.

Model	Label	F1	Precision	Recall	Support
NB	False	0.971	0.987	0.956	7740
	True	0.714	0.615	0.850	641
RF	False	0.982	0.980	0.984	7740
	True	0.780	0.802	0.760	641
SVM	False	0.982	0.977	0.986	7740
	True	0.766	0.813	0.724	641

Table 10. Performance of the models at the prediction level 'top-one domain'.

Model	Label	F1	Precision	Recall	Support
NB	False	0.615	0.848	0.483	381
	True	0.820	0.727	0.941	558
	micro avg	0.755	0.755	0.755	939
	macro avg	0.718	0.788	0.712	939
RF	False	0.729	0.747	0.711	370
	True	0.830	0.818	0.844	569
	micro avg	0.791	0.791	0.791	939
	macro avg	0.779	0.782	0.777	939
SVM	False	0.771	0.772	0.770	392
	True	0.837	0.836	0.837	547
	micro avg	0.809	0.809	0.809	939
	macro avg	0.804	0.804	0.804	939

6.2 Analysing the quality of fitted model

We inspected the quality of the selected model using three types of analysis: we computed the model performance per subpopulation, we computed a learning curve and we analysed the distribution of the predicted probabilities for the correct and incorrect domains. Each analysis will be explained further below.

6.2.1 Model performance per subpopulation

We computed the model performance for the selected model on the complete test set and on three subpopulations within the test set:

- 'website+' legal units, with a domain obtained from the COC

- 'website+' legal units, with a domain obtained from DP
- 'website-' legal units

The results for the classifier performance by subpopulation at the prediction level 'all candidate domains' showed that the performance of the category 'False' was nearly the same for all three subpopulations, see Table 11. Note that the precision for the 'website -' legal units for the category 'False' is 1.0 by design, because we did not include cases with category 'True' in the current study. For the category 'True', the F1, precision and recall were better for legal units with a domain obtained from DP than for legal units with a domain obtained from the COC.

Table 11. Performance of the SVM model at the prediction level 'all candidate domains', for three subpopulations.

Subpopulation	Label	F1	Precision	Recall	Support
All	False	0.98	0.98	0.99	7740
	True	0.77	0.81	0.72	641
	micro avg	0.97	0.97	0.97	8381
	macro avg	0.87	0.89	0.86	8381
Website +, COC	False	0.98	0.98	0.99	3748
	True	0.75	0.80	0.70	309
	micro avg	0.96	0.96	0.96	4057
	macro avg	0.86	0.89	0.84	4057
Website +, DP	False	0.98	0.97	0.99	3264
	True	0.81	0.90	0.74	332
	micro avg	0.97	0.97	0.97	3596
	macro avg	0.90	0.94	0.87	3596
Website -	False	0.98	1.00	0.97	728
	True	0.00	0.00	0.00	0.00
	micro avg	0.97	0.97	0.97	728
	macro avg	0.49	0.50	0.48	728

For the 'top-one domain' level, see Table 12, we found that the recall and the F1 of the category 'False' was better for 'website+' legal units with a domain from DP than for those with a domain from COC. For the 'website-' legal units the recall had a lower score than the 'website+' legal units. The performance of the category 'True' was better for the 'website+' legal units with a domain from DP (F1 of 0.89) than those with a domain from COC (F1 of 0.82). The overall performance for finding URLs was quite good.

Table 12. Performance of the SVM model at the 'top-one domain' level, for different subpopulations.

Subpopulation	Type	F1	Precision	Recall	Support
All	False	0.77	0.77	0.77	392
	True	0.84	0.84	0.84	547
	micro avg	0.81	0.81	0.81	939
	macro avg	0.80	0.80	0.80	939
Website +, COC	False	0.75	0.74	0.75	183
	True	0.82	0.82	0.82	262
	micro avg	0.79	0.79	0.79	445
	macro avg	0.78	0.78	0.78	445
Website +, DP	False	0.77	0.72	0.84	129
	True	0.89	0.92	0.85	285
	micro avg	0.85	0.85	0.85	414
	macro avg	0.83	0.82	0.84	414
Website -	False	0.83	1.00	0.71	80
	True	0.00	0.00	0.00	0
	micro avg	0.71	0.71	0.71	80
	macro avg	0.42	0.50	0.36	80

6.2.2 Learning curve

We checked whether the model results could have been further improved by including more learning examples. We computed a learning curve at the prediction level 'all candidate domains' for the F1 score, based on a five-fold cross validation within the set of training examples.

Figure 11. Learning curve at the prediction level 'all candidate domains' using a five-fold cross validation on the trainingsset. The bandwidth shows the variation over each of the five cross-validation results.

The learning curve shows that the F1 score for the legal units that were held out in each cross-validation score stabilised around 4000 training instances (red line in

Figure 11). Furthermore, the curve shows that the F1 score for the legal units in the trainingsset stabilised 12 000 training instances. Overall it strongly suggests that the performance of the model will hardly increase by adding more training instances.

6.2.3 Probability distribution of correct and incorrect domains

For each legal unit in the test set we sorted the retrieved domains by their predicted probability, from high to low. The domain with the highest predicted probability was given rank 1, the second highest was given rank 2 and so on. Next, we made a frequency distribution of the rank numbers of the correct domains (within the set of retrieved candidate domains) as well as for the incorrect domains. Additionally, we made a frequency distribution of the probabilities of the correct domains (within the set of retrieved domains) as well as for the incorrect domains.

The correct domain corresponded in the majority of the cases with the predicted top-one domain, i.e. the domain with the highest predicted probability, see Figure 12. In a small part however, it corresponded with the domain with rank 2 (the second highest probability). Lower ranks also occurred although not very often. The distribution of the ranks for the incorrect domains showed that all ranks up to about 25 occurred. The frequency of ranks peaked at rank 2 and gradually decreased thereafter. It decreased from 2 onwards simply because the number of legal units with two search results is larger than with three search results and so on.

Surprisingly, the distribution of the predicted probability for the correct domains showed two peaks: one peak at high probabilities, of say $P(\hat{v}_k = \text{'True'}) > 0.9$, and one peak at small probabilities of $P(\hat{v}_k = \text{'True'}) < 0.1$ (see Figure 13). The first peak corresponds with true positives and the second peak with false negatives. We checked a number of units for which the correct domain had a small probability. Those units were found to have small values for their search engine features (ρ_k and z_k) but rather high values for their agreement features (ρ_{jk}). So the set of units in Figure 13 that show a peak at low probabilities involve units with a domain that is obtained at a relatively low position in the set of search results but for which their identifying variables in the GBR agree reasonably well with those of the search results.

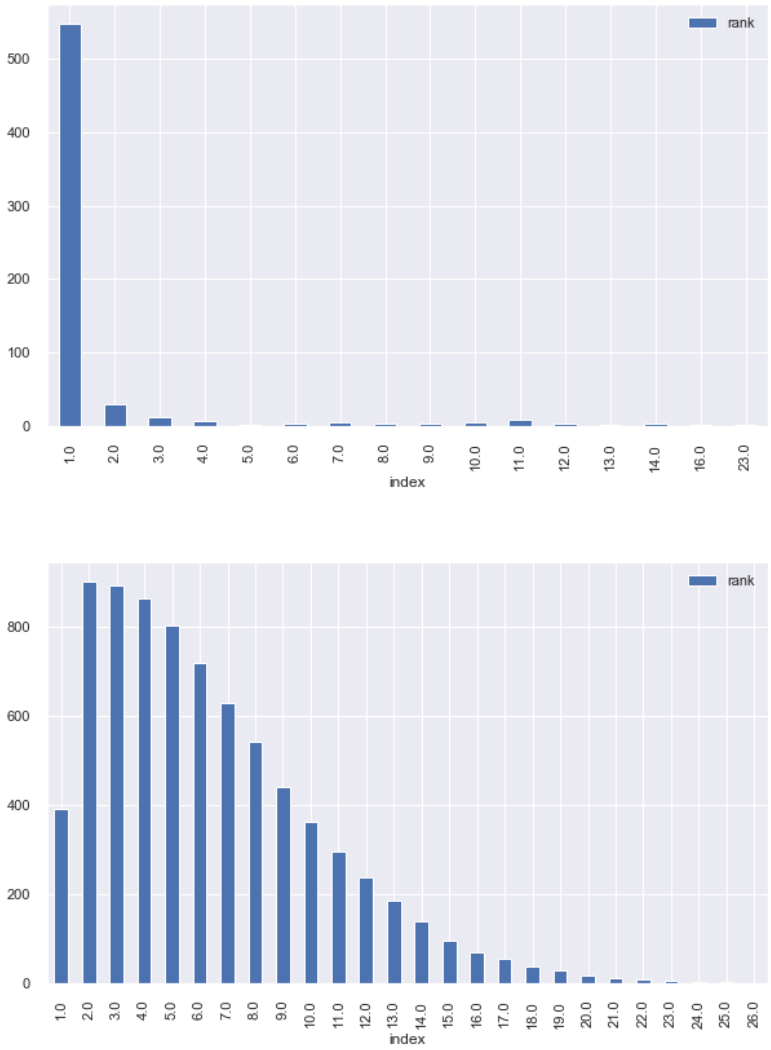

Figure 12. Frequency distribution for the rank numbers (horizontal axis) of the correct domains (upper panel) and the incorrect domains (lower panel) in the test set.

The predicted probability for the incorrect domain showed a very large peak at small probabilities (see Figure 13). Still a limited number of cases are found with probabilities of 0.5 and higher, the false positives. Recall that this probability distribution is computed at the prediction level 'all candidate domains', so by design the number of the correct domains is equals the number of legal units with a domain whereas the number of incorrectly found domains is much larger.

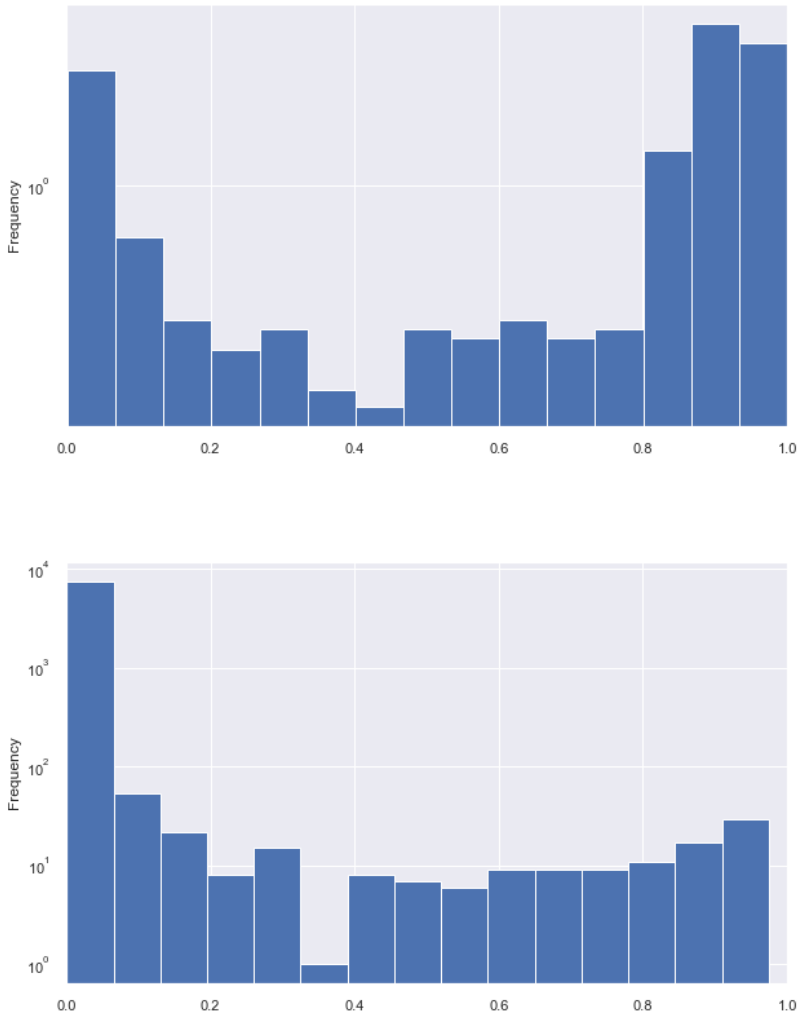

Figure 13. Frequency distribution (vertical axis, log-scale) for the estimated probability (horizontal axis) for the correct domains (upper panel) and for the incorrect domains (lower panel).

7. Discussion

We have successfully developed a method to search for URLs, more specifically: domains, of legal units. Our method can be applied to amend a GBR that contains already a proportion of legal units with a domain, but it can also be applied to lists of population units for which one aims to know the domain. An example of the latter can be found in Meertens et al. (2018) that aims to find domains of European webshops. Our approach is very similar to that of ISTAT (Barcaroli et al. 2018) who trained a machine learning model to find the URLs of enterprises that were part of the ICT survey population. A difference is that ISTAT visited the websites and scraped

contact information from the sites. That approach is more costly and more time consuming than our approach where we directly use the search results themselves.

Our method can be applied in practice, by running the (Python) code that we developed. We are currently refactoring the code in order to make it available through GitHub, see https://github.com/SNStatComp/urlfinding. It will consist of different modules. The minimum set consists of a URL search module, a feature derivation module and a module to apply the trained model. Possible future extensions could be a module to retrain the model, a module to train other algorithms, a feature selection module and a search query analysis module.

The basic method that we use automatically enters contact information of legal units from the COC via a Google API: trade and legal name, address and so on. Although the COC data are open source, we should be aware that Google analyses search queries and might reconstruct the enumeration of legal units in our business register. Of course, we want to prevent that anyone can reveal our business register. We are aware of three precautions that one can take to greatly reduce the risk for this to happen. A first precaution is to search only for a *selection* of our register and, within a 'search session', take this selection to be random rather than systematic. Secondly, during a 'search session' one can randomise the units as well as the query types over different search engines. Within the set of search engines, one can add search engines where one can search anonymously. A third precaution is to mix noise into the search queries: one can add names, addresses and so on to the search queries that do not belong to the target population where one is interested in and one could add query types that are meaningless. We leave it for future research how these precautions can be used best in practice.

An analysis of our search queries showed that some of the query types were complementary to each other, meaning that for a given legal unit those complementary query types resulted in different domains that were returned. We also found that some of the query types resulted in a considerable overlap in the returned domains. In future, we might investigate whether we can find nearly the same set of domains with fewer query types.

The analysis of the relevance of our features for the machine learning models showed that the two most important features for predicting the correct domain were the two search engine features. These features might well depend on the kind of search engine that is used and on the version of the search engine. These features are sensitive to the algorithm behind the search engine. It will therefore be necessary to regularly retrain the model to ensure the quality of the URL retrieval outcomes.

We see a number of options to improve the current agreement variables. First of all, we used a tokeniser that split postal codes of the form '0000 AA' into the tokens '0000' and 'AA'. That does not lead to an optimal comparison with the postal codes in the GBR. An improvement would be to use a regular expression to extract a postal code from the text. Similarly, we could also use regular expressions to extract email addresses and a phone numbers. Another improvement could be to visit the

retrieved websites, as was done by Barcaroli et al. (2018), scrape its content and try to extract a legal unit identification number and other identification variables from the website. These identification variables can then be added to the features of the machine learning model. A further improvement might be to analyse in more detail which features are important to achieve a stable model performance. For instance, we now have retained two variants of each agreement features namely 'min' and 'max'. Maybe it is sufficient to retain only the 'max' variant, since that variant often had a higher score than the 'min' variant. A final improvement concerns the computation of the level of agreement between the contact variables which was based on the Jaro-Winkler similarity. The fictional example in Table 6 showed that unrelated word pairs can have a surprisingly high Jaro-Winkler similarity. In future research we might experiment with other word pair distance measures such as the Levenshtein distance.

We have obtained the precision, recall and F1 score on the total test set for COC, DP and the ICT survey part of the tests sets separately. We did not evaluate the overall performance of our URL finding within the GBR. We can do this in future by taking a random sample from the legal units within the GBR - for the population that we have developed the model for - and determining whether the legal unit has a website and if so which domain is correct. This can be done manually. Instead we did evaluate the model performance for three subpopulations: 1) the 'website+' legal units with a domain from the COC, 2) the 'website+' legal units with a domain from DP but not from COC and 3) the 'website-' legal units. What has not been evaluated yet is the model performance for legal units without a known domain from COC or DP. We have no reason to believe that the performance for legal units in the GBR with no known domain from COC nor from DP is different from subpopulation 2) or 3), but it would be better to assess this explicitly in future.

We experienced two fundamental difficulties with training a machine learning model for URL finding that needs to be addressed. The first fundamental difficulty was: how to treat legal units for which we did not retrieve any website? In the current paper, we used the approach that we simply cannot say anything about those units based on the (absence) of search results. This absence may indicate that the legal unit does not have a website, but it may also mean that we simply incorrectly did not find the website of the legal unit. In the near future, we aim to split the prediction of domains in two steps. In the first step, a machine learning model predicts whether a legal unit has a website or not. For the group for which the model predicts that it has a website, a second machine learning model aims to predict which domain is the correct one. In this new situation, we can separately make training examples for having a website (yes/no) with features to predict this. We could for instance create a feature that counts the number of search queries without search results.

The second fundamental difficulty was that we trained the model for all retrieved domains, but ultimately we are only interested at the performance of the model for the top-one domain. Once we have selected the domain with the largest probability we have one result per legal unit, and we can compute the performance at legal unit level. The performance for the category 'False' (the retrieved domain is incorrect)

was clearly better at the prediction level 'all candidate domains' than at the prediction level 'top-one domain' because at the former level there are multiple incorrect domains per legal unit (and only one correct domain per legal unit). A practical solution is that we return the top-two domains for each legal unit which enlarges the probability that the correct domain is found. We might then visit the top-two domains and scrape the contact information from the website. For instance, one can extract a legal unit identification number and a value added tax number from the website. This can subsequently be used to compute additional features and might be used to train an additional machine learning model for predicting the correct domain.

An interesting finding was that the probability distribution for the correct domains had two peaks: one at high and one at low probabilities. The peak at a low probability appears to refer to units which have poorer scores on the ranking in the search results (Google gives them a low rank) whereas its identifying variables correspond quite well with those of the GBR. The low ranking of these enterprises is likely to be caused by the ranking system that Google uses to rank the webpages, PageRank and other algorithms. For instance it may well be that a webpage of a business with a more commonly occurring street name is ranked lower than the webpage of a business with a more rarely occurring street name. A possible improvement of our model is to train a second machine learning model, but now without the search engine features (ρ_k, z_k). Next, we could predict the probability that the domain is the correct by both models and take the maximum probability over the two models.

In summary, there are a few measures that we can take to improve the presented approach to find domains. One is to first predict whether a legal unit has a website or not and then to predict which domain. A second improvement is to scrape a limited set of most promising candidate domains and extract contact information. Also, using regular expressions might help to find email addresses, phone numbers, legal unit identification numbers or value added tax numbers. A third potential improvement is to train an additional machine learning model that only includes the identifying variables. One could then take the maximum probability over a model that has both agreement features and search engine features, and a model which only includes agreement features.

Apart from the improvement of the current model, there are a few points at which the scope of the current URL retrieval method can be broadened. First of all, we limited the current approach to legal units with a one-to-one relationship to enterprises and it was restricted to enterprises with ten or more employees. One extension is to test whether the method can also be used for smaller enterprises. Another extension is to include legal units with a many-to-one relationship with enterprises. Furthermore, we try to link domains to legal units by using contact-type of information. A third extension is to include geographical information in the search process, see Holness (2018).

Acknowlegdements

The authors thank Piet Daas, Bas Haverkort, and Guido van den Heuvel for their valuable comments to an earlier version of this paper. The authors thank DataProvider for giving access to their scraped website data.

8. Appendix: evaluation measures

In the current paper, we selected the candidate domain with the highest predicted probability. This domain is predicted to be correct if its predicted probability is larger than 0.5. We are interested to count number of cases in the predicted versus the labelled categories for the candidate domains. It is important to understand how this was scored; this is shown in Table 13. Let us assume that the correct domain of a targeted legal unit is 'twinkle' and that we have retrieved a candidate domain 'twinkle'. Since this candidate domain corresponds to the correct domain, the label category is 'True'. If the predicted probability for this domain is larger than 0.5 then the predicted category is 'True' otherwise it is 'False'. Now, assume that we have another candidate domain, namely 'twilight'. For this domain the label is 'False' since it is not the correct domain. Now, like before, if the predicted probability for this domain by the model is larger than 0.5 then the predicted category is 'True' otherwise it is 'False'.

Table 13. Example how the label and the predicted categories are scored for a legal unit with 'twinkle' as the correct domain.

Retrieved candidate domain	Predicted probability that the predicted domain is correct	Label category for the candidate domain	Predicted category for the candidate domain
'twinkle'	0.6	True	True
'twinkle'	0.4	True	False
'twilight'	0.8	False	True
'twilight'	0.4	False	False

The confusion matrix with the predicted versus the label categories is given in Table 14. The symbol N_{10} stands for the number of cases with label category 'True' (subscript 1) and predicted category 'False' (subscript 0). Subscript '•' stands for the total of 'True' and 'False'.

Table 14. Confusion matrix with the predicted versus the label categories.

	Predicted category (h)		
Label category (g)	True	False	Total
True	N_{11}	N_{10}	$N_{1\bullet}$
False	N_{01}	N_{00}	$N_{0\bullet}$
Total	$N_{\bullet 1}$	$N_{\bullet 0}$	$N_{\bullet\bullet}$

The recall for label category g is given as:

$$\text{Recall}(g) = N_{gg}/N_{g\bullet} \tag{3}$$

The precision for predicted category h is given as:

$$\text{Precision}(h) = N_{hh}/N_{\bullet h} \tag{4}$$

The F1 score for label category k is the harmonic mean of the precision for category k and recall of category k and is given by:

$$\begin{aligned} &\text{F1}(k) \\ &= 2 \cdot \text{Precision}(k) \cdot \text{Recall}(k)/\big(\text{Precision}(k) + \text{Recall}\,(k)\big) \end{aligned} \tag{5}$$

The macro-average over the two classes of the score functions recall, precision and F1, is given by their unweighted average. For instance, for the recall the macro-average is given by $\text{Macro Recall} = \{\text{Recall}\,(g = \text{True}) + \text{Recall}\,(g = \text{False})\}/2$.

The micro-average score for recall, precision is computed by directly using the counted number of both classes in the numerator and denominator of the formula. For instance, for recall, the micro-average is given by

$$\text{Micro avg Recall} = (N_{11} + N_{01})/(N_{1\bullet} + N_{0\bullet}) = (N_{11} + N_{01})/N_{\bullet\bullet} \tag{6}$$

The micro-average F1 is the harmonic mean of the micro average recall and the micro average precision.

A disadvantage of the F1 score as given in (5) is its sensitivity to an imbalance in the number of units per category. In our situation, category 'False' dominates for the training of the model which is done for all retrieved domains. We therefore used an alternative score function to train the model, namely Matthews Correlation Coefficient (see e.g. Powers, 2011). In Matthews Correlation Coefficient, errors in both categories are equally weighted.

Matthews Correlation Coefficient, denoted by MCC, is defined as:

$$\text{MCC} = (P_{11} \cdot\ P_{00} - P_{01}P_{10})/\sqrt{(P_{1\bullet} \cdot P_{\bullet 1} \cdot P_{0\bullet} \cdot P_{\bullet 0})} \tag{7}$$

where $P_{gh} = N_{gh}/N_{\bullet\bullet}$ stand for the relative cell frequencies.

It can be shown (see below) that the absolute value of the Matthews Correlation Coefficient is equivalent to:

$$|\mathrm{MCC}| = \sqrt{\chi^2/N_{\bullet\bullet}} \tag{8}$$

where χ^2 is given by

$$\chi^2 = N_{\bullet\bullet} \sum_{h,g} \left(P_{gh} - P_{g\bullet}P_{\bullet h}\right)^2 / P_{g\bullet}P_{\bullet h} \tag{9}$$

where P_{gh} stands for the observed cell proportion and $P_{g\bullet}P_{\bullet h}$ is the expected cell proportion when the predicted cell proportions are independent of the true cell proportions. A χ^2 of 0 implies that the machine learning model is as good as throwing a coin, whereas a χ^2 of 1 implies that the label categories 'True' and 'False' are perfectly predicted. Matthews Correlation Coefficient is also referred to as the phi-correlation coefficient (Jurman et al. 2012).

To show that (7) and (8) are equivalent, we first rewrite $\chi^2/N_{\bullet\bullet}$ as follows:

$$\begin{aligned}\frac{\chi^2}{N_{\bullet\bullet}} &= \frac{(P_{11} - P_{1\bullet}P_{\bullet 1})^2}{P_{1\bullet}P_{\bullet 1}} + \frac{(P_{00} - P_{0\bullet}P_{\bullet 0})^2}{P_{0\bullet}P_{\bullet 0}} + \frac{(P_{10} - P_{1\bullet}P_{\bullet 0})^2}{P_{1\bullet}P_{\bullet 0}} \\ &\quad + \frac{(P_{01} - P_{0\bullet}P_{\bullet 1})^2}{P_{0\bullet}P_{\bullet 1}} \\ &= \frac{(P_{11} - P_{1\bullet}P_{\bullet 1})^2}{P_{1\bullet}P_{\bullet 1}P_{0\bullet}P_{\bullet 0}} \{P_{0\bullet}P_{\bullet 0} + P_{1\bullet}P_{\bullet 1} + P_{\bullet 1}P_{0\bullet} + P_{1\bullet}P_{\bullet 0}\} \\ &= \frac{(P_{11} - P_{1\bullet}P_{\bullet 1})^2}{P_{1\bullet}P_{\bullet 1}P_{0\bullet}P_{\bullet 0}}\end{aligned} \tag{10}$$

where in the second line we used that all four terms of in the denominator are of equal size. In the third line we used that the sum over all expected cell proportions is 1.

Given the outcome of (10) we now only have be show that $P_{11} \cdot P_{00} - P_{01}P_{10} = \sqrt{(P_{11} - P_{1\bullet}P_{\bullet 1})^2} = P_{11} - P_{1\bullet}P_{\bullet 1}$:

$$\begin{aligned}P_{11}\,P_{00} - P_{01}P_{10} &= P_{11} \cdot \left(P_{00} - \frac{P_{01}P_{10}}{P_{11}}\right) \\ &= P_{11}\left(P_{00} + P_{11} + P_{10} + P_{01}\right. \\ &\quad \left. - \left\{P_{11} + P_{10} + P_{01} + \frac{P_{01}P_{10}}{P_{11}}\right\}\right) \\ &= P_{11} \cdot \left(1 - \left\{P_{11} + P_{10} + P_{01} + \frac{P_{01}P_{10}}{P_{11}}\right\}\right) \\ &= P_{11} - (P_{11}^2 + P_{11}P_{10} + P_{11}P_{01} + P_{01}P_{10}) \\ &= P_{11} - (P_{11} + P_{10})(P_{11} + P_{01}) \\ &= P_{11} - P_{1\bullet}P_{\bullet 1}\end{aligned} \tag{11}$$

9. References

Arasu, A., Götz, M. and Kaushik, R. (2010). On Active Learning of Record Matching Packages. In: Proceedings of the 2010 ACM SIGMOD *International Conference on Management of data*, 783-794. Indianapolis.

Ariel, A., Bakker, B.F.M., de Groot, M., van Grootheest, G., van der Laan, J., Smit, J. and Verkerk, B. (2014). Record Linkage in Health Data: a simulation study. Statistics Netherlands. Available at http://www.biolink-nl.eu/public/2014%20Record%20linkage%20simulation.pdf (accessed March 2018)

Barcaroli, G., Scannapieco, M. and Summa, D. (2018) On the Use of Internet as a Data Source for Official Statistics: a Strategy for Identifying Enterprises on the Web. *Rivista Italiana di Economia Demografia e Statistica 70(4)*. Available at https://www.istat.it › files › 2018/06 › a4_RIEDS-2016 (Accessed November 2019).

Berardi, G., Esuli, A., Fagni, T. and Sebastiani, F. (2015). Classifying websites by industry sector: A study in feature design. *Proceedings of the 30th ACM Symposium on Applied Computing*, 1053-1059

Bosch, O. ten, Windmeijer, D. and Le, M. Q. (2016). Verslag internetrobots familiebedrijven. CBS memo. (in Dutch).

Christen, P. (2008). Automatic record linkage using seeded nearest neighbour and support vector machine classification. *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 151–159. Las Vegas USA, 24-27 August 2008.

Christen, P. (2012). Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. Data-Centric Systems and Applications. Berlin Heidelberg: Springer-Verlag.

Cochinwala, M., Kurien, V., Lalk, G. and Shasha, D. (2001). Efficient data reconciliation. *Information Sciences 137*(1-4), 1-15.

Cover, T.M. and Thomas, J.A. (1991). Elements of Information Theory. New York, USA: Wiley interscience.

Doef, S. van der, Daas, P.J.H. and Windmeijer, H.J.M. (2018). Identifying innovative companies from their website. Abstract presented at the Conference Big data meets survey design (BigSurv18). Barcelona, Spain, 25-27 October 2018

Felligi, I. and Sunter, A. (1969). A theory of record linkage. *Journal of the American Statistical Association 64* (328), 1183–1210.

Hastie, T., Tibshirani, R. and Friedman, J. (2009). The elements of statistical learning. Springer Series in Statistics (2nd edition), New York: Springer.

Hertzog, T., Scheuren, F. and Winkler, W. E. (2007). Data quality and record linkage techniques. New York, USA: Springer Verlag.

Heemann, D. (2018). Records linkage in official statistics. Internship report of Leiden University, The Netherlands. (available upon request)

Holness, P. (2018). Automated Methods of Data Collection: Retail locations and shopping centres in Vancouver. *Paper presented at the International Methodology Symposium, 6–9 November 2018 Ottawa, Canada*. A summary is available at https://www150.statcan.gc.ca/n1/pub/12-206-x/2018001/01-eng.htm

Jaro, M. A. (1989). "Advances in record linkage methodology as applied to the 1985 census of Tampa Florida". *Journal of the American Statistical Association 84* (406), 414–420.

Jurman, G., Riccadonna, S. and Furlanello, C. (2012). A Comparison of MCC and CEN Error Measures in Multi-Class Prediction. *PLoS ONE 7*(8): e41882. doi:10.1371/journal.pone.0041882

Meertens, Q.A., Diks, C.G.H., Herik, H.J. van den and Takes, F.W. (2018). A Data-Driven Supply-Side Approach for Measuring Cross-Border Internet Purchases. Available at arXiv:1805.06930v1 [stat.AP] 17 May 2018.

Oostrom, L., Walker, A.N., Staats, B., Slootbeek-Van Laar, M., Ortega Azurduy S. and Rooijakkers B. (2016). Measuring the internet economy in The Netherlands: a big data analysis. CBS Discussion paper 2016-14. Available at https://www.cbs.nl/-/media/_pdf/2016/40/measuringthe-internet-economy.pdf (accessed march 2018).

Powers, D.M.W. (2011). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies 2*, 37-63

Sarawagi, S. and Bhamidipaty, A. (2002). Interactive Deduplication using Active Learning. Proceedings of the 8th ACM SIGKDD international conference on Knowledge discovery and data mining, 269-278. Edmonton, Canada, 23 - 26 July 2002.

Tuoto, T. (2016). New proposal for linkage error estimation. *Statistical Journal of the IAOS* 32, 413–420.

Winkler, W. E. *(1990).* "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage". *Proceedings of the Section on Survey Research Methods. American Statistical Association, 354–359.*

Explanation of symbols

Empty cell	Figure not applicable
.	Figure is unknown, insufficiently reliable or confidential
*	Provisional figure
**	Revised provisional figure
2017–2018	2017 to 2018 inclusive
2017/2018	Average for 2017 to 2018 inclusive
2017/'18	Crop year, financial year, school year, etc., beginning in 2017 and ending in 2018
2013/'14–2017/'18	Crop year, financial year, etc., 2015/'16 to 2017/'18 inclusive

Due to rounding, some totals may not correspond to the sum of the separate figures.

Colophon

Publisher
Centraal Bureau voor de Statistiek
Henri Faasdreef 312, 2492 JP Den Haag
www.cbs.nl

Prepress
Statistics Netherlands, CCN Creation and visualisation

Design
Edenspiekermann

Information
Telephone +31 88 570 70 70, fax +31 70 337 59 94
Via contactform: www.cbsl.nl/information