Building Domain-specific Corpora from the Web

van Noord, Rik; Garcia-Romero, Cristian; Esplà-Gomis, Miquel; Pla Sempere, Leopoldo; Toral, Antonio

Published in:
Proceedings of the BUCC Workshop within LREC 2022

Proceedings of the 15th Workshop on Building and Using Comparable Corpora (BUCC 2022) @LREC2022 , pages 23–32
Marseille, June 25, 2022

Building Domain-specific Corpora from the Web: the Case of European Digital Service Infrastructures

Rik van Noord[1], Cristian García-Romero[2], Miquel Esplà-Gomis[2]
Leopoldo Pla[2], Antonio Toral[1]
[1]University of Groningen, [2]Universitat d'Alacant
rikvannoord@gmail.com, cgarcia@dlsi.ua.es
mespla@dlsi.ua.es, lpla@dlsi.ua.es, a.toral.ruiz@rug.nl

Abstract

An important goal of the MaCoCu project is to improve EU-specific NLP systems that concern their Digital Service Infrastructures (DSIs). In this paper we aim at boosting the creation of such domain-specific NLP systems. To do so, we explore the feasibility of building an automatic classifier that allows to identify which segments in a generic (potentially parallel) corpus are relevant for a particular DSI. We create an evaluation data set by crawling DSI-specific web domains and then compare different strategies to build our DSI classifier for text in three languages: English, Spanish and Dutch. We use pre-trained (multilingual) language models to perform the classification, with zero-shot classification for Spanish and Dutch. The results are promising, as we are able to classify DSIs with between 70 and 80% accuracy, even without in-language training data. A manual annotation of the data revealed that we can also find DSI-specific data on crawled texts from general web domains with reasonable accuracy. We publicly release all data, predictions and code, as to allow future investigations in whether exploiting this DSI-specific data actually leads to improved performance on particular applications, such as machine translation.

Keywords: Digital Service Infrastructures, Text Classification, Web Crawling

1. Introduction

The Connecting Europe Facility (CEF)[1] was set up by the European Commission to promote growth, jobs and competitiveness through targeted infrastructure investment at the European level. A key component is the e-Translation platform[2] of the European Language Resource Coordination program, which provides automated translation to facilitate multilingual communication and exchange of documents between public administrations and citizens of the EU and CEF-affiliated countries. A main application of this platform is on their services called *Digital Service Infrastructures* (henceforth DSIs, see Table 1 for an overview). For these services to function adequately, it is of vital importance that the automatic translations of texts and documents are of high quality.

Among DSIs, it is easy to identify clearly different textual domains, such as information technologies, health systems, legal processes, etc. On the other hand, they are also complex, compartmentalized and often highly specific, making it challenging, for example, to train a single machine translation (MT) model that would perform well across all DSIs. It would clearly be beneficial to use domain-specific MT systems for different areas and domains, rather than using a single generic MT system for all of them. We therefore work under the hypothesis that the MT used within the scope of each DSI can be improved by carefully selecting relevant training data per individual DSI, rather than simply using generic training data. Common methods to exploit such data include pre-training on generic data and fine-tuning on domain-specific data (Luong and Manning, 2015; Freitag and Al-Onaizan, 2016), instance weighting (Wang et al., 2017) and pivot-based domain adaptation (Li et al., 2018; Ben-David et al., 2020). In order to obtain this domain-specific data, we would require an automatic system that can classify sentences into whether they fit in a DSI or not. To the best of our knowledge, no such system exists yet. Therefore, in this paper, we aim at building such a DSI classifier as a first step in potentially creating DSI-specific MT systems. Given the multilingual nature of Europe and the DSIs, we will attempt to build a classifier that can handle multiple languages. To achieve this, we first crawl DSI-specific websites, whose content will then be used to train our automatic classifier.

Our ultimate goal, as part of the MaCoCu project[3], is to apply this classifier to generic web-crawled corpora in official EU (or related) languages, such as ParaCrawl (Bañón et al., 2020a).[4] We will not release hard categories per sentence or document, but rather release the softmax probability distribution of our best model over the DSI categories.[5] Users can then simply select their own threshold in selecting instances per DSI. Since most of these corpora are parallel with English, only having an English parser could suffice, but we would also want to be able to classify non-parallel corpora for non-English languages. Therefore, we will also train

[1] https://ec.europa.eu/inea/en/connecting-europe-facility

[2] https://webgate.ec.europa.eu/etranslation/public/welcome.html

[3] https://macocu.eu/

[4] https://www.paracrawl.eu/

[5] Though note that not all DSIs are necessarily completely disjoint classes (see Section 2).

		English		Spanish		Dutch	
DSI	**Domain**	Crawled	Clean	Crawled	Clean	Crawled	Clean
BRIS	Business, Market	0	0	0	0	0	0
Cybersecurity	ICT	1,390,239	209,053	176,886	40,425	5,237	759
EESSI	Social security	267,086	49,345	30,181	2,398	5,979	739
E-health	Health, Medicine	63,582	13,891	75	31	0	0
E-justice	Justice, Law	6,942,090	262,933	2,277,413	146,968	1,356,537	151,752
E-procurement	Public procurement	23,133	3,557	0	0	0	0
Europeana	Culture	965,220	14,327	76,037	1,566	0	0
ODR	Consumers' rights	4,669,948	163,365	3,849,469	104,251	101,704	20,842
Open Data Portal	Multiple domains	33,792,223	75,394	254	19	703	228
Safer Internet	ICT	134,439	24,767	142	39	125	9

Table 1: The number of crawled and cleaned sentences per DSI, per language.

a multilingual model on the English data, that is able to perform zero-shot classification. We will test our method on Spanish and Dutch, aside from English, as these languages are MaCoCu objectives. Though we look in particular at DSIs, we believe this paper can be beneficial to all researchers that are interested in classifying web-crawled data for specific textual domains.
A description of the crawling of DSI-specific data is provided in Section 2, after which we evaluate the performance of our DSI classifiers in Section 3. Our English and Spanish classifiers perform quite well, with Dutch lagging a bit behind. We obtain the best DSI classification performance by fine-tuning a pretrained language model, with DEBERTA for English and XLM-R for Spanish and Dutch. We then apply the best English model on two corpora of unseen ParaCrawl sentences in Section 4 and analyse its performance by manually annotating a subset of the data.

2. Data

DSIs The targeted DSIs (listed in Table 1, taken from the MaCoCu project) range from rather general (E-health, Cybersecurity) to highly specific, such as Electronic Exchange of Social Security Information (EESSI) and the Business Registers Interconnection System (BRIS). Even looking at just the DSIs themselves, and the corresponding textual domains, shows that this task will be challenging. First, there is considerable overlap between the domain of some DSIs, namely for Cybersecurity & Safer Internet and for E-justice & Online Dispute Resolution (ODR).[6] Moreover, there are also DSIs that are very general and hard to define exactly in terms of domain (e.g. Europeana, Open Data Portal).

ELRC-Share There already exists a database with corpora that are tagged with certain DSIs: ELRC-Share (Lösch et al., 2018). However, on a closer inspection we found that it did not match our exact needs. First, many of the corpora are tagged with all DSIs, but do not actually assign a DSI per sentence, document or any subset of the corpus. The tags only seem to indicate that the corpus *could* be useful when working with DSI data. Second, the DSI tags often seem questionable or plain wrong. For example, there are a number of corpora tagged with *Europeana* that contain just general texts (news, Wikipedia) and are not specific to the *Culture* domain (see Table 1). Third, the correctly tagged corpora usually contain little data or are highly specific, likely making it difficult to train a general classifier on it. Fourth, even if there is data available, it is mainly for English, with very sparse resources for other languages. For these reasons, we decided to crawl our own DSI-specific data. We will outline this process below.

2.1. Crawling DSI-specific web domains

First, we create a methodology to select the DSI-specific web domains we will crawl. For some DSIs there was only a single domain publicly available (e.g. *Europeana*). In the case of the DSIs that do not have a specific portal, we manually checked the publicly available information about projects related to these DSIs.[7] We also used Google results to obtain more web domains. Finally, we selected the official website of the European Commission[8], since it contains data relevant for some DSIs, though in the end we only found data for *EESSI* (`ec.europa.eu/social`). Note that for certain DSIs, the whole service consists of more than what can be found on a website, for example software packages for *Cybersecurity*. The full list of domains crawled per DSI can be found in Appendix C.
Once we selected all the web domains for the DSIs with services available on a website, we used Bitextor[9] in order to crawl them and process the result-

[6] However, throughout the paper, we do treat them as separate categories.

[7] `https://ec.europa.eu/inea/en/connecting-europe-facility/cef-telecom/projects-by-dsi`

[8] `https://ec.europa.eu`

[9] `https://github.com/bitextor/bitextor`

	English			Spanish		Dutch	
DSI	Train	Dev.	Test	Dev.	Test	Dev.	Test
Cybersecurity	207,053	1,000	1,000	1,000	1,000	379	380
EESSI	47,345	1,000	1,000	1,000	1,000	369	370
E-health	11,891	1,000	1,000	0	0	0	0
E-justice	260,933	1,000	1,000	1,000	1,000	1,000	1,000
Europeana	12,327	1,000	1,000	783	783	0	0
Online Dispute Resolution	161,365	1,000	1,000	1,000	1,000	1,000	1,000
Open Data Portal	73,394	1,000	1,000	0	0	114	114
Safer Internet	22,767	1,000	1,000	0	0	0	0
Other	797,075	8,000	8,000	4,783	4,783	2,862	2,864
Total	1,594,150	16,000	16,000	9,566	9,566	5,724	5,728

Table 2: Label division for the sentence-level train, development and test sets for the three languages of interest.

ing data.[10] Bitextor is a tool to harvest bitexts from multilingual websites, but in this case we have just used the first part of the pipeline, which is a monolingual process. For crawling, we use `wget` and store the data downloaded in the Web ARChive (WARC) file format..[11] Then, WARC files are processed using `warc2preprocess`, which involves:

1. Applying the Fix Text For You library (FTFY) (Speer, 2019) to fix common text problems such as *mojibake* (that is, garbled text that is the result of text being decoded using an unintended character encoding).
2. Detecting the language of the documents with CLD2[12] and discarding those which are not in one of the targeted languages.
3. Removing boilerplates (that is, text which is the same from page to page, usually menu items or footer elements) using Boilerpipe (Kohlschütter et al., 2010).
4. Parsing HTML using the HTML tokenizer implemented in the Python code of `warc2preprocess` in Bitextor, which takes into account the structure of the HTML elements for a more accurate paragraph and segment delimitation when extracting plain text.

We apply a number of cleaning steps to the extracted texts after the 4 previously described steps of the WARC process. First, we split the text into sentences using the Moses sentence splitter (Koehn et al., 2007) and normalize quotes, dashes and other punctuation. Then, we tokenize the sentences using SpaCy[13] and only keep those with more than 6 and less than 50 tokens. This is the step where we lose the majority of the crawled sentences, as the crawls often contain short texts that are likely headers, links or menu options which are not filtered out by Boilerpipe. Finally, we filter out sentences that are (near)-duplicates, sentences that do not end with punctuation and sentences that are classified as a different language according to CLD3.[14] In Table 1, *Clean* shows the number of sentences per DSI, per language that are left after this final cleaning process. Multiple authors carried out a manual inspection on a sample of the cleaned data, which confirmed that the data was of high quality and relevant for the selected DSIs, according to our criteria.

2.2. Splits

We did not find sufficient training data for all DSI-language pairs. For English, we do not train and evaluate on BRIS and E-procurement. For Dutch and Spanish we do not need training data (since we perform zero-shot classification), but even so we only find sufficient data in 5 out of 10 DSIs (see Table 2). For each DSI, we take (at most) 1,000 sentences for the development and test set. We split the data sequentially, e.g. the first 11,891 crawled sentences of *E-health* are put in the training set, while the last 2,000 are put in the dev and test set, respectively. We do this to minimize train-test overlap: this way, sentences from the same webpage will not occur in both train and test. We did experiment with random splitting (where this overlap would be possible) and found higher F_1-scores, indicating that this indeed had an effect.

We also want our model to be able to recognize sentences that do not belong to any of the DSIs. To this end, we introduce the *Other* category, which consists of random sentences taken from Paracrawl (Bañón et al., 2020b) release v9. For English, the sentences are taken from the parallel side of the Spanish and Dutch releases. We actually expect that most of the randomly

[10] See Figure 4 in Appendix A for exact settings.

[11] `https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1/`

[12] `https://github.com/CLD2Owners/cld2`

[13] `https://spacy.io/`

[14] CLD2 was used only at the document level, as it can parse HTML and detect language of text blocks; CLD3 was used at the segment level for robustness, as it is more accurate than CLD2 but cannot be used on HTML documents.

crawled sentences would fit this non-DSI category best (and our analyses in Section 4 seem to confirm this). To strike a balance between mimicking this expected distribution and enabling the model to learn about DSIs specifically, we ensure that half of the training, development and test set sentences belong to this category. An important thing to note about this *Other* category is that it might contain instances that could well belong to a DSI. In other words: predicting a DSI instead of *Other* is not necessarily a mistake, though we do treat it as such throughout the paper.

Down-sampling Our training set distribution is quite different from that of the development and test sets. Therefore, it is likely that it is suboptimal (or at least inefficient) to maintain all training instances during training. We experiment with *down-sampling* the majority categories during training, i.e. randomly selecting a subset of instances per DSI. Importantly, the *Other* category gets a special treatment: we ensure it is always the same size as the DSI-instances combined (similar as the initial division in Table 2). As an example, down-sampling to 10,000 sentences per DSI means a total training set of 80,000 + 80,000 = 160,000 instances.

3. Experiments

This section outlines our experimental setup and experiments we performed. All code to reproduce our results is publicly available at: `https://github.com/RikVN/DSI`

Baseline As a baseline system, we use a simple bag-of-words support vector machine (SVM) model implemented using scikit-learn (Pedregosa et al., 2011). Our best baseline model is a linear SVM that uses unigrams and bigrams with a tf-idf vectorizer. Each feature has to occur at least five times (regardless of corpus size) to be included and we use a C-value of 1. Other settings are left at default.

Language models Our main classification method is fine-tuning a pretrained (multilingual) neural language model (LM). We use the (de facto) default method of fine-tuning such an LM: adding a single classification layer (with dropout) on top of the pooled layers, as implemented in the `transformers` library of Huggingface (Wolf et al., 2020). To determine which pretrained LM is the most suitable for our task, we experiment with quite a number of LMs that are well-established in the literature. For English, we experiment with BART (Lewis et al., 2020), BERT (Devlin et al., 2019), CANINE (Clark et al., 2021), DEBERTA (He et al., 2021), ELECTRA (Clark et al., 2020), Longformer (Beltagy et al., 2020), ROBERTA (Liu et al., 2019), XLM-en (Conneau et al., 2020) and XLNET (Yang et al., 2019), while for the zero-shot experiments for Spanish and Dutch we experiment with M-BART, M-BERT, M-DEBERTA and XLM-R. For models that have a base and large variant available, we experimented only with the large models. We apply temperature scaling (Guo et al., 2017) to ensure a better probability distribution in the final classification layer. We select the best models in Section 3.1.

	Acc.	Prec.	Rec.	F_1
BART-large	77.3	67.7	65.3	65.9
BERT-large	75.8	66.1	63.3	64.0
CANINE	68.8	56.6	54.0	54.9
DEBERTA-v3-large	**77.5**	**68.1**	**66.1**	**66.4**
ELECTRA-large	74.4	64.3	61.7	62.3
Longformer-large	76.3	67.0	63.9	64.7
ROBERTA-large	75.8	66.3	63.2	64.1
XLM-en	65.1	52.6	50.5	51.0
XLNET-large	77.0	67.9	65.3	66.1

Table 3: Development set results (all in %) for English DSI-classification for a number of pretrained LMs. Precision, recall and F_1-score are macro-averaged.

Evaluation As stated previously, we ultimately intend to release probability distributions of the classifier. However, for evaluation purposes, we still evaluate our models by using hard classification (i.e. by taking the argmax of the probability distribution). For each experiment we report both the accuracy as well the macro-averaged precision, recall and F_1-score. Numbers are single runs, unless otherwise indicated.

3.1. English DSI classification

First, we try to find the LM that is most suitable for this task. For efficiency reasons we perform these experiments on a subset of our data set: down-sampling each DSI-category to 3,000 instances and therefore using 24,000 instances for *Other* (see last paragraph of Section 2.2). The development and test sets are not changed. For each LM, we tune the learning rate, as the default learning rate is often far from optimal. The results of this experiment are reported in Table 3.
We take the best performing system (DEBERTA-large) and tune the other hyper-parameters. Specifically, we experiment with warm-up ratio, label smoothing, dropout, batch size and gradient clipping (see Appendix B for best settings and range of values tried). Our best performing system obtained an accuracy and F_1-score of 77.5% and 66.4%, respectively. We also experimented with freezing the LM layers and only training the classification layer, but this did not lead to improved performance.

3.2. Zero-shot DSI classification

We also perform zero-shot multilingual DSI classification by fine-tuning pretrained multilingual language models (MLMs). We train only on the English data set, and test on the Spanish and Dutch sets. We apply similar steps as for the English language models: we experiment with different pre-trained MLMs, for which we only tune the learning rate. The other hyperparameters are set to the best values we found in the English experiments. Note that for both Spanish and Dutch, this is only 6-class classification, as opposed to

	Spanish				Dutch			
	Acc	P	R	F_1	Acc	P	R	F_1
M-BART	73.5	**77.7**	58.5	64.4	63.5	52.1	47.4	46.5
M-BERT	70.8	70.6	56.5	61.3	60.7	42.0	42.2	40.6
M-DEBERTA	74.3	74.5	63.6	68.0	62.8	54.8	49.7	48.4
XLM-R	**76.1**	77.4	**65.4**	**70.5**	**64.9**	**55.4**	**53.2**	**50.8**

Table 4: Development set results (all in %) for zero-shot DSI-classification for Spanish and Dutch. Precision (P), recall (R), and F_1 score are macro-averaged.

the 9-class classification task for English. The results are shown in Table 4. We find that XLM-R is the best model for both languages, though the difference with M-DEBERTA is modest. Generally, we find the scores to be promising, given that it is a zero-shot multi-class classification. Interestingly, the best Spanish model obtains higher F_1-scores than the best English model, though the task is also somewhat easier since Spanish only has six classes. Moreover, Spanish has no data for *Open Data Portal*, which was the hardest DSI for the English model (see Appendix D).

3.3. Down-sampling ratio

Previous experiments were performed using down-sampled data sets of at most 3,000 instances per DSI in the training set. To get the best performance, we aim to find the optimal down-sampling size for the best model per language. We plot the performance in Figure 1. Interestingly, even the LMs still benefit from large amounts of extra data, though the differences are modest. Best performance for the models is obtained for down-sampling the categories to between $20,000$ and $50,000$ instances. Note that all models were tested for $> 50,000$ instances, but always decreased in performance. For each language, we select the best model and evaluate on the test set. These scores are shown in Table 5. For English and Spanish, the model performs quite well, with accuracies around 80%. Interestingly,

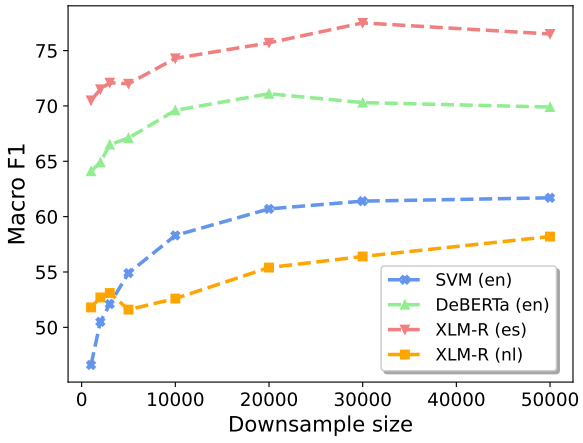

Figure 1: Dev set macro F_1 scores (in %) per down-sampled size per category for the different languages.

		Acc.	Prec.	Rec.	F_1
EN	Dev.	79.8 ± 0.6	73.7 ± 0.7	68.6 ± 1.7	70.4 ± 0.7
	Test	77.3 ± 0.3	71.6 ± 0.9	64.4 ± 1.1	67.1 ± 0.3
ES	Dev.	81.2 ± 0.4	81.3 ± 0.9	74.5 ± 0.7	77.5 ± 0.4
	Test	80.2 ± 0.2	80.0 ± 0.8	72.7 ± 0.7	75.9 ± 0.3
NL	Dev.	70.9 ± 1.0	61.2 ± 1.3	63.0 ± 1.4	57.9 ± 0.8
	Test	74.1 ± 1.1	67.0 ± 1.2	65.2 ± 0.8	62.8 ± 0.5

Table 5: Final development and test set scores (in %) of our best model per language. Results are averaged over three runs. Note that since we calculate the macro average, F_1 is not necessarily between *Prec.* and *Rec.*

the scores for Dutch actually increase on the test set. The detailed scores per DSI are shown in Appendix D. The hardest DSI to classify for the English model is *Open Data Portal*. This is not unexpected; as we noted previously, this is a very broad DSI that consists of multiple domains (see Table 1). The model does quite well on *Other*, which is likely due to it being the majority class, but also on *Europeana*. We hypothesize that this is due to *Europeana* being the most dissimilar DSI, as compared to the other DSIs, since it is not related to any legal or digital EU domains.

It is interesting to observe which categories are most difficult to distinguish for the model. The confusion matrix of our best English model is shown in Figure 2. Curiously, Cybersecurity and Safer Internet are actually not confused often, even though they are both in the ICT domain. Cybersecurity is, however, the most common wrong prediction for E-justice, which is also surprising, as the two do not seem directly related. Lastly, *Open Data Portal* seems to be a very broad DSI, since it is confused with a lot of different DSIs.

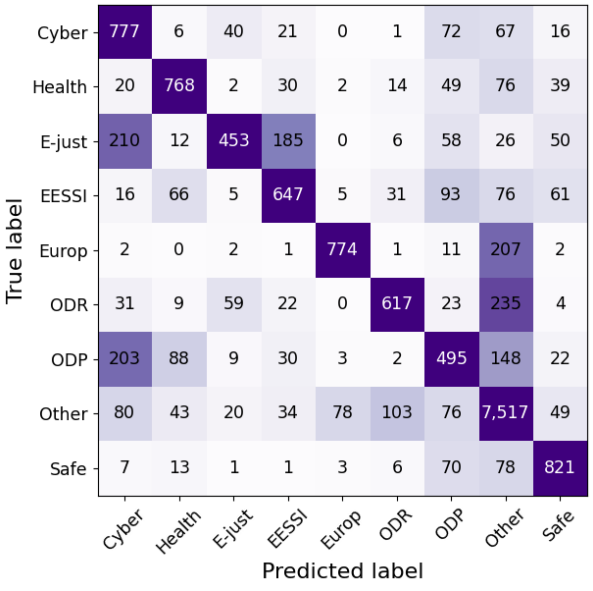

Figure 2: Confusion matrix of development set performance of our best English model.

	Dutch-English					Spanish-English				
	>0.3	**>0.5**	**>0.7**	**>0.8**	**>0.9**	**>0.3**	**>0.5**	**>0.7**	**>0.8**	**>0.9**
Cybersecurity	1.1	0.8	0.6	0.5	0.3	1.3	1.0	0.8	0.7	0.4
EESSI	0.7	0.5	0.4	0.3	0.1	0.7	0.6	0.4	0.3	0.1
E-health	0.9	0.5	0.3	0.2	0.1	0.8	0.4	0.3	0.2	0.1
E-justice	0.6	0.5	0.4	0.3	0.1	0.6	0.5	0.4	0.3	0.1
Europeana	2.1	1.7	1.4	1.3	0.6	2.2	1.8	1.5	1.3	0.6
Online Dispute Resolution	3.0	2.5	2.1	1.8	1.2	3.3	2.7	2.3	2.0	1.3
Open Data Portal	0.6	0.5	0.4	0.3	0.2	0.7	0.6	0.5	0.4	0.2
Safer Internet	0.4	0.3	0.3	0.2	0.1	0.6	0.5	0.4	0.3	0.2
Other	90.7	90.0	89.1	88.2	82.9	89.9	89.1	88.2	87.3	82.1

Table 6: Percentage of total instances per DSI per softmax threshold, when classifying 89 million and 269 million sentences for the English-Dutch and English-Spanish ParaCrawl releases with our best English model. Note that the columns do not necessarily sum to 100%.

4. Analysis

Classifying unseen data The ultimate goal of our system is to classify previously unseen generic web-crawled data. To get a sense of how many DSI-specific instances we can find in such randomly crawled data, we use our best English model to classify the English sentences from the latest Dutch-English and Spanish-English ParaCrawl releases, consisting of 89 million and 269 million sentences, respectively.[15] Note that since we used this data also to create the *Other* category, we actually train two models to ensure the model that is used never saw any of the ParaCrawl data as *Other* during training. The results for using different softmax thresholds are shown in Table 6. As expected, the vast majority of the data does not get classified as belonging to a specific DSI. Around 8% of the sentences get classified as a DSI for a softmax threshold value > 0.5, which quickly decreases for higher values. Though small, this is not necessarily a problem, since there are billions of English sentences publicly available, potentially allowing us to still create large corpora per DSI for this language.

Manual annotation However, this method will only work well if the predictions on unseen data are of reasonable quality. To evaluate this, we asked an expert annotator to manually annotate 800 of the ParaCrawl predictions, 100 for each DSI. We asked the annotator: *Does this sentence fit in DSI X*? For 400 sentences, *X* is actually the predicted DSI by our best English model. In the other 400, the DSI is chosen randomly. This lets us compare how meaningful the predicted DSIs are, without having to annotate from scratch, which greatly speeds up the process. We do not annotate *Other*, as this is meaningless: all sentences potentially fit this DSI, so annotators by definition should always answer "yes" to whether the sentence fits this category.

The results are shown in Table 7, and are mostly reassuring. As an example, let us look at the DSI *Cybersecurity*. For the 100 instances the model predicted this DSI, the annotator was asked 50 times whether the sentences actually belonged in *Cybersecurity*, answering "yes" in 50% of those instances. For the other 50, the annotator was asked whether the sentence belonged to a randomly selected different DSI. Of those 50, only 10% of the sentences were accepted as belonging to that DSI. We found similar results for all DSIs, as on average, predicted DSIs by the model are about 5 times as likely to fit that DSI than randomly selected DSIs. On the other hand, for 4 out of 8 DSIs less than half of the predictions are actually annotated to fit the respective DSI (first column of results).

Model confidence We can now also analyse the importance of the softmax probability (i.e. the confidence) of the model. In other words: does the model get more accurate as it gets more confident? For 400 annotations, where *X* was the predicted DSI, we now know whether the model made a fitting prediction. For the other 400, answering "no" during the annotation process does not tell us if the prediction of the model was correct, only that the randomly picked DSI was incorrect. Using the former 400 instances, we plot the accuracy of the model over minimum confidence val-

	Pred. (%)	**Random (%)**
Cybersecurity	50.0	10.0
EESSI	42.0	10.0
E-health	44.0	12.0
E-justice	78.0	8.0
Europeana	88.0	2.0
ODR	54.0	14.0
Open Data Portal	36.0	14.0
Safer Internet	66.0	20.0
Total	57.2	11.2

Table 7: Percentage of "yes" annotations per DSI. **Pred** means the model actually predicted this DSI, while **Random** means we picked a random DSI to annotate for the respective sentence.

[15] Predictions available at `https://macocu.eu`

DSI	Type	Best features
Cybersecurity	All	enisa, cybersecurity, concordia, cert, nis, cyber, vulnerability, attacker
	Word	cert, nis, vulnerability, attacker, vulnerabilities, security, attackers, the agency
EESSI	All	eures, egf, fead, easi, administrative commission, movers, social partners, etuc
	Word	administrative commission, movers, social partners, posting industrial relations, apprenticeships, vet, workers
E-health	All	ehtel, digitalhealtheurope, twinning, ehealth, mhealth, digital health, dhe, telemedicine
	Word	twinning, digital health, health data, twinnings, healthcare, scirocco, patient, health
E-justice	All	eurojust, ccbe, jits, jit, ocg, isil, lawyers, videoconferencing
	Word	lawyers, debtor, court, creditors, judicial, this treaty, prosecutor, casework
Europeana	All	europeana, beavers, beaver, lindgren, hotjar, simberg, this gallery, merian
	Word	beavers, beaver, this gallery, curie, kimono, counterculture, digital object, rights statement
ODR	All	eni, fastweb, sncf, amf, riai, cssf, cru, ecogra
	Word	cru, uke, issuers, lithuania, management company, irish water, nais, state legal
Open Data Portal	All	open data, psi, datasets, technical purpose, dataset, edp, portals, re users
	Word	open data, psi, datasets, technical purpose, dataset, edp, portals, re users
Safer-internet	All	inhope, csam, hotline, bik, hotlines, sic, helpline, aviator
	Word	hotline, sic, aviator, better internet, bee secure, sid, media literacy, young people
Other	All	your, the, you, god, triodos, is, click, hotel
	Word	your, the, you, god, is, click, hotel, reserves the

Table 8: Most important SVM-features per DSI for English DSI classification. The row "word" shows the 8 best features that are also English words.

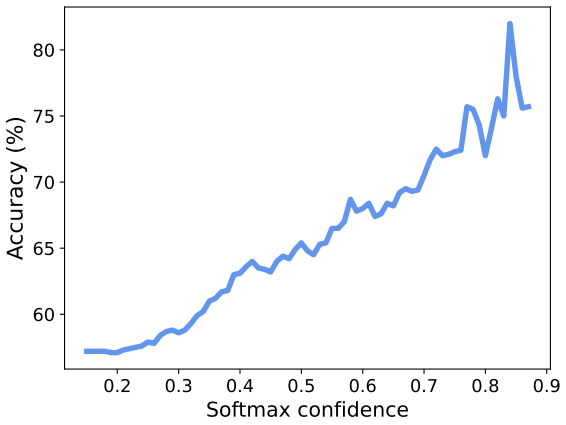

Figure 3: Accuracy of the best English model for 400 annotated instances, with a minimum confidence.

ues in Figure 3, with a confidence of 0.15 including all 400 instances, while a confidence of 0.85 only includes 50. This gives us a clear answer to our question: the model indeed gets more accurate as it gets more confident. This means that it is possible for users to determine their own data/quality trade-off, with higher softmax thresholds leading to fewer data that is of higher quality. It is hard for us to suggest an optimal threshold value, as it will likely differ per task, but 0.5 seems like a good default value.

Best features To get some insight in the data, we show the most important SVM features in Table 8. Since the best features were often specific abbreviations, we also show the best features that are also English words.[16] For some DSIs, such as *Cybersecurity*, *E-health* and *E-justice*, the best features make intuitively a lot of sense, and we can be reasonably sure that the model will be able to detect correct documents for this DSI. However, for other DSIs the best features seem overly specific. For example, we do not expect that *beaver* and *lindgren* are good general indicators for *Europeana*, though it does also include more intuitive features, such as *this gallery* and *digital object*. Especially the features for *Online Dispute Resolution* are a bit concerning, since the actual features are mainly abbreviations (that are not that likely to occur in randomly crawled texts), while the word-features do not seem to point to general disputes.

[16] https://github.com/dwyl/english-words

5. Conclusion

One of the goals of the MaCoCu project is improving EU-specific NLP systems that work with Digital Service Infrastructures (DSIs). In this paper, as a necessary and vital first step, we focused on creating a system that can classify texts into specific DSIs. First, we introduced a data set for DSI classification by crawling DSI-specific web domains. We then trained classifiers for English, Spanish and Dutch by fine-tuning a (multilingual) pre-trained language model. The models performed quite well on in-domain data. A manual evaluation of out-of-domain data showed that while DSI-specific data is scarce, we can still find such data with reasonable accuracy. We have already applied our model on two large corpora and made all data, models and predictions publicly available. Future work can then determine whether exploiting such DSI-specific data will indeed lead to improved performance. Finally, we plan to extend our method to more EU (or related) languages, such as Icelandic, Croatian, Bulgarian, Turkish and Slovene.

6. Acknowledgements

The MaCoCu project has received funding from the European Union's Connecting Europe Facility 2014-2020 - CEF Telecom, under Grant Agreement No. INEA/CEF/ICT/A2020/2278341. This communication reflects only the authors' views. The Agency is not responsible for any use that may be made of the information it contains. We thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster. Lastly, we thank Mikel L. Forcada for providing us with valuable comments on a draft of this paper.

7. Bibliographical References

Bañón, M., Chen, P., Haddow, B., Heafield, K., Hoang, H., Esplà-Gomis, M., Forcada, M. L., Kamran, A., Kirefu, F., Koehn, P., et al. (2020a). Paracrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567.

Bañón, M., Chen, P., Haddow, B., Heafield, K., Hoang, H., Esplà-Gomis, M., Forcada, M. L., Kamran, A., Kirefu, F., Koehn, P., Ortiz Rojas, S., Pla Sempere, L., Ramírez-Sánchez, G., Sarrías, E., Strelec, M., Thompson, B., Waites, W., Wiggins, D., and Zaragoza, J. (2020b). ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online, July. Association for Computational Linguistics.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Ben-David, E., Rabinovitz, C., and Reichart, R. (2020). PERL: Pivot-based domain adaptation for pre-trained deep contextualized embedding models. *Transactions of the Association for Computational Linguistics*, 8:504–521.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. (2020). ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Clark, J. H., Garrette, D., Turc, I., and Wieting, J. (2021). Canine: Pre-training an efficient tokenization-free encoder for language representation. *arXiv preprint arXiv:2103.06874*.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online, July. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Freitag, M. and Al-Onaizan, Y. (2016). Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

He, P., Liu, X., Gao, J., and Chen, W. (2021). DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.

Kohlschütter, C., Fankhauser, P., and Nejdl, W. (2010). Boilerplate detection using shallow text features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July. Association for Computational Linguistics.

Li, Z., Wei, Y., Zhang, Y., and Yang, Q. (2018). Hierarchical attention transfer network for cross-domain sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lösch, A., Mapelli, V., Piperidis, S., Vasiļjevs, A., Smal, L., Declerck, T., Schnur, E., Choukri, K., and van Genabith, J. (2018). European language resource coordination: Collecting language resources for public sector multilingual information management. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Luong, M.-T. and Manning, C. (2015). Stanford neural machine translation systems for spoken language

domains. In *Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 76–79, Da Nang, Vietnam, December 3-4.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Speer, R. (2019). ftfy. Zenodo. Version 5.5.

Wang, R., Utiyama, M., Liu, L., Chen, K., and Sumita, E. (2017). Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark, September. Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October. Association for Computational Linguistics.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

A. Bitextor settings

```
# BASIC VARIABLES
dataDir: ~/dsis/e-health/perm/data
permanentDir: ~/dsis/e-health/perm
transientDir: ~/dsis/e-health/trans

until: "split"
profiling: true

# DATA SOURCES - CRAWLING
hostsFile: ~/dsis/e-health.txt
crawler: "wget"
crawlTimeLimit: "96h"

# PREPROCESSING
shards: 8 # 2^8 = 256 shards
batches: 1024 # chunks of 1024 MB

langs: ['en', 'es', 'nl']

preprocessor: "warc2preprocess"
ftfy: true
boilerplateCleaning: true
parser: "simple"
```

Figure 4: Bitextor configuration file.

B. Hyperparameters

Parameter	Range
Learning rate	$10^{-7}, 10^{-6}, 5 \times 10^{-6}, \mathbf{10^{-5}}, 5 \times 10^{-5}$
Batch size	$\{8, \mathbf{12}, 16, 24, 32\}$
Warmup	$\{0.05, \mathbf{0.1}, 0.2, 0.3, 0.5\}$
Label smoothing	$\{0.05, \mathbf{0.1}\}$
Dropout	$\{0.0, 0.05, \mathbf{0.1}, 0.15, 0.2, 0.3\}$
LR decay	$\{\mathbf{0}, 0.01, 0.05, 0.1\}$
Max grad norm	$\{0.5, \mathbf{1}, 1.5, 2\}$

Table 9: Hyperparameter range and final values (bold) for our final English (DEBERTA) and multilingual Spanish/Dutch models (XLM-R). Hyperparameters not included are left at their default value.

C. Web-crawled domains

DSI	Domains
Cybersecurity	www.enisa.europa.eu, ecsc.eu, www.concordia-h2020.eu, www.ccn-cert.cni.es www.incibe-cert.es, maltacip.gov.mt, csirt.cy, csirt.cynet.ac.cy
EESSI	ec.europa.eu
E-health	ehealth-hub.eu, ehtel.eu, digitalhealtheurope.eu
E-justice	e-justice.europa.eu, www.notariesofeurope.eu, www.ejn-crimjust.europa.eu, www.ejnforum.eu www.eurojust.europa.eu, www.ccbe.eu, eubailiff.eu, eur-lex.europa.eu
Europeana	europeana.eu
ODR	accademiadr.it, atlantique-mediation.org, batirmediation-conso.fr, begravningar.se, bekeltetes-csongrad.hu, bekeltetes.hu, conciliazione.a2a.eu conciliazione.gruppoiren.it, conso.immomediateurs.com, ...
Open Data Portal	data.europa.eu, stirdata.eu
Safer-internet	www.betterinternetforkids.eu, www.saferinternetday.org, inhope.org

Table 10: Web-crawled domains. All the domains will be available at the repository provided in Section 3.

D. Detailed scores

	English						Spanish						Dutch					
	Dev			Test			Dev			Test			Dev			Test		
DSI	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Cybersecurity	60.5	73.2	66.2	55.3	58.9	57.0	77.3	82.4	79.4	77.8	77.7	77.7	64.6	58.3	61.3	71.5	63.9	67.5
EESSI	64.6	66.9	65.7	66.7	69.9	68.8	79.7	76.8	78.2	79.8	81.6	80.7	45.8	81.8	58.7	43.4	77.8	55.8
E-health	75.0	77.2	76.1	70.4	75.7	72.9	—	—	—	—	—	—	—	—	—	—	—	—
E-justice	70.9	47.7	57.0	69.5	58.6	63.6	68.7	60.7	64.4	69.1	60.4	64.5	85.7	64.0	73.3	84.1	69.5	76.1
Europeana	89.6	78.3	83.6	85.6	59.6	70.3	89.4	82.8	85.9	87.5	81.1	84.2	—	—	—	—	—	—
ODR	75.7	64.6	69.1	71.0	52.7	60.5	74.0	58.3	65.2	71.9	51.9	60.3	37.0	6.4	10.9	70.7	25.3	37.3
Open Data Portal	55.4	50.1	52.6	51.2	45.6	48.3	—	—	—	—	—	—	38.2	68.4	49.1	38.0	62.3	47.2
Safer Internet	74.6	82.4	78.3	77.1	81.3	79.2	—	—	—	—	—	—	—	—	—	—	—	—
Other	89.8	93.5	91.6	86.7	93.2	89.8	92.4	90.0	91.2	89.2	89.8	89.5	79.2	92.1	85.2	84.4	92.0	88.0
Macro	72.9	70.3	71.1	70.5	66.2	67.8	80.3	75.2	77.5	79.2	73.7	76.2	58.4	61.8	56.4	65.3	65.1	62.0

Table 11: Full results per DSI for using the best model for all three languages. Results are on the first run of the system, not averaged over three runs as in Table 5.