



Eidgenössische Technische Hochschule Zürich  
Swiss Federal Institute of Technology Zurich

# Privacy Observatory: Collecting Privacy Policies and Terms of Service on a Regular Basis

Bachelor Thesis

Truong Hoang Long

January 18, 2023

Advisors: Prof. Dr. David Basin, Karel Kubicek  
Department of Information Security, ETH Zürich

---

## Abstract

Privacy and contractual regulations require websites to inform users about the data collection and terms of use. Websites do so in privacy policy (*PP*) and terms & conditions (*TC*), respectively.

We developed a way to continuously retrieve websites' PP/TC by means of web crawling to aid in the long-term research of the effects data privacy regulations such as the GDPR has on the data collection practices and their communication to users. We take inspiration from previous studies that already crawl privacy policies to develop a high-precision PP/TC crawler with support for all major European languages.

Our crawler manages to identify PP with 87.6% accuracy and TC with 85.6% accuracy. The PP/TC documents collected by our crawler are in Markdown format, suitable for analysis by humans and machines alike.

---

## Acknowledgements

I would like to thank Karel Kubicek, my supervisor. I am deeply grateful for your guidance, your continued support, and your invaluable feedback. This thesis was made possible because of you.

I would like to thank the Department of Information Security at ETH Zurich for providing me with the computational resources required to conduct this thesis.

Lastly, I would like to thank my friends and family. You encouraged and supported me through all the ups and downs during my years of study and through the process of researching and writing this thesis.

---

# Contents

---

<b>Contents</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Our contribution . . . . .	1
<b>2 Overview of the crawling process</b>	<b>3</b>
2.1 Loading the index page . . . . .	3
2.2 Locating the PP/TC pages . . . . .	3
2.3 Saving the result . . . . .	4
<b>3 Languages</b>	<b>5</b>
3.1 Supported languages . . . . .	5
3.2 Language detection . . . . .	6
3.3 Switching to a preferred language . . . . .	6
<b>4 Keyword matching</b>	<b>7</b>
4.1 Matching schema . . . . .	7
4.2 Keyword weights & scoring . . . . .	8
<b>5 Content classification</b>	<b>9</b>
5.1 Document preprocessing . . . . .	9
5.1.1 Content extraction . . . . .	9
5.1.2 Translation . . . . .	10
5.2 Privacy policy classification . . . . .	10
5.3 Terms & conditions classification . . . . .	10
<b>6 Manual evaluation</b>	<b>11</b>
6.1 Language detection accuracy . . . . .	11
6.2 PP/TC page identification accuracy . . . . .	11
6.2.1 Comparison with previous study . . . . .	13

<b>7</b>	<b>Deployment &amp; results</b>	<b>15</b>
7.1	Crawling data set . . . . .	15
7.2	Crawling infrastructure . . . . .	15
7.2.1	Hardware . . . . .	15
7.2.2	Software . . . . .	16
7.2.3	Network . . . . .	16
7.2.4	Database . . . . .	16
7.3	Statistics . . . . .	16
7.3.1	Crawling speed . . . . .	16
7.3.2	Page load statistics . . . . .	18
7.3.3	Language distribution . . . . .	18
7.3.4	Number of PPs and TCs found . . . . .	19
<b>8</b>	<b>Discussion</b>	<b>21</b>
8.1	Limitations . . . . .	21
8.2	Future work . . . . .	21
8.3	Conclusion . . . . .	22
	<b>Bibliography</b>	<b>23</b>

## Chapter 1

---

# Introduction

---

For years, online businesses have been collecting increasing amounts of personal data from their users. These businesses profit from the direct data sale, advertisement, or by providing targeted content to users. Thode et al. [1] showed that users without technical background are unaware of the scope of tracking, and Ur et al. [2] found that even if the users expect their data to be collected, they are not properly informed about the methods and the magnitude collection.

In response, governmental bodies have come forward with new regulations, demanding users to be informed and given control of the data they share. Privacy laws, like the EU's General Data Protection Regulation (*GDPR*) [3] or ePrivacy Directive [4] require websites to inform users about the data collection practices in a privacy policy (*PP*). Consumer protection laws on the other hand require websites to also provide the terms & conditions (*TC*).

Studies of personal data collections by websites are dependent on comparison of the observed data collection with the declared behavior in the *PP*. Similarly, studies of contracts need the comparison with the declared *TC*. Linden et al. [5] collected privacy policies from the most popular websites to analyze the effectiveness of the *GDPR*. Amos et al. [6] used a web crawler to retrieve *PPs* stored on the Internet Archive's Wayback Machine<sup>1</sup> and created a data set consisting of 1 million *PPs* to study the changes to *PPs* over time.

### 1.1 Our contribution

Since past studies did not publish the code used for *PP* and *TC* collection or their scope was limited to websites using specific web technologies or available in limited languages, we want to reproduce the crawler for *PP* and *TC* collection. Our crawler should allow studying the long-term effects that

---

<sup>1</sup><https://web.archive.org/>

regulations such as the GDPR have on the digital landscape by collecting and monitoring the privacy policy and terms & conditions of websites within the EU.

While the crawlers used by Linden et al. [5] and Amos et al. [6] only retrieve PP and only support English, our crawler also extracts TC and supports every major spoken language in Europe (detailed in Section 3.1).

The crawler in this thesis is based on the generic web forms crawler developed by Lodrant [7], which itself is based on the automatic account registration bot developed by Kast [8]. The new crawler uses a reworked keyword matching algorithm together with an ML-based content classifier to improve the PP/TC identification accuracy while re-using parts of the old crawler that already work very well, such as the bot-detection countermeasures.

Our crawler found PP on 44.5% and TC on 27.3% of websites that were selected from EU countries using Chrome user experience report. In a manual evaluation, we found that the accuracy of detecting PP and TC is 87.6% and 85.6%, respectively. In comparison, the crawler developed by Lodrant [7] achieves 75.0% and 63.6% accuracy for PP and TC respectively.

# Overview of the crawling process

---

In this section we shortly describe the procedure of crawling a single domain.

## 2.1 Loading the index page

If the domain supplied to the crawler is valid, the crawler begins the crawl at the index page of that domain. To avoid bot detection, the crawler uses the `undetected-chromedriver`<sup>1</sup> Python package to load websites. If the index page fails to load (HTTP error, DNS error, timeout, etc.), the crawl is stopped.

Otherwise, the crawler moves on to the next step, language detection and selection. If the website is not in a language supported by the crawler and it is not possible to switch languages, the crawl is stopped.

## 2.2 Locating the PP/TC pages

To find the pages containing PP/TC, the crawler looks at all links on the pages it has loaded and uses keyword matching to determine the likelihood of containing the PP/TC for each link.

For each type of content that has not been found, the crawler follows the link with the highest likelihood of leading to that content. Upon loading that page, a classifier will decide if the page indeed contains the content we are looking for. If it does, the content will be saved both as HTML and Markdown, and the result is written to the database. Otherwise, the crawler will keep exploring the website, following the link with the next highest likelihood.

The crawl will stop once both the PP and TC have been found, or if the crawler has reached a predetermined limit on the number of pages explored.

---

<sup>1</sup><https://pypi.org/project/undetected-chromedriver/>



## 2.3 Saving the result

For each domain crawled, the crawler creates a database entry with the following data:

- `domain`: the domain crawled.
- `uuid`: a random string that serves as a unique identifier of the crawl, enabling the same domain to be crawled multiple times.
- `index_status`: the load status of the index page.
- `lang`: language code of the language detected, or NULL if the language is either unsupported or not recognized.
- `pp_url`: the URL of the PP, if it is found.
- `tc_url`: the URL of the TC, if it is found.

A directory with the format `<domain>_<uuid>` is created on the machine executing the crawl that contains:

- `index.html` (the HTML of the index page).
- `privacy_policy.html` and `privacy_policy.md` (in case the PP was found).
- `terms_of_use.html` and `terms_of_use.md` (in case the TC was found).

## Chapter 3

---

# Languages

---

Because keyword matching and content classification require that we know the language of the website being crawled, it is imperative that the crawler can correctly identify the language of the current page and is able to change the language to something that has received more optimization if necessary.

### 3.1 Supported languages

As the crawler will be used to study European websites, it is necessary to support all major European languages. The full list is as follows.

- Albanian
- Basque
- Bosnian
- Bulgarian
- Catalan
- Croatian
- Czech<sup>(\*)</sup>
- Danish<sup>(\*)</sup>
- Dutch<sup>(\*)</sup>
- English<sup>(\*)</sup>
- Estonian
- Finnish<sup>(\*)</sup>
- French<sup>(\*)</sup>
- Galician
- German<sup>(\*)</sup>
- Greek<sup>(\*)</sup>
- Hungarian<sup>(\*)</sup>
- Icelandic
- Italian<sup>(\*)</sup>
- Latvian
- Lithuanian
- Luxembourgish
- Macedonian
- Maltese
- Norwegian
- Polish<sup>(\*)</sup>
- Portuguese<sup>(\*)</sup>
- Romanian
- Russian<sup>(\*)</sup>
- Serbian
- Slovak<sup>(\*)</sup>
- Slovenian
- Spanish<sup>(\*)</sup>
- Swedish<sup>(\*)</sup>
- Turkish<sup>(\*)</sup>
- Ukrainian<sup>(\*)</sup>
- Welsh

The language detection library employed by the crawler supports all of the above-mentioned languages. Our keyword matching algorithm also supports all of these languages, albeit with varying levels of completeness. However, LibreTranslate<sup>1</sup>, the translation back-end used to support the ML-classifier, only supports the 23 languages marked by (\*).

### 3.2 Language detection

The beautifulsoup<sup>2</sup> Python package is used to extract all text elements from the HTML of the index page, which gets passed to the language detector implemented in the polyglot<sup>3</sup> Python package.

### 3.3 Switching to a preferred language

There are multiple reasons why one would want to change the language on a domain.

- The detected language is unsupported.
- The crawler is unable to identify the current language.
- The keyword matching algorithm may perform better on a different language.

To switch to a different language, the crawler looks for links with URLs containing a language and/or country code in the path. These are some examples of URLs the crawler looks for when trying to switch to English:

- <https://domain.name/en/>
- <https://domain.name/en-US/>
- <https://domain.name/index.html?lang=en/>

Links that are found are visited and the resulting page goes through language detection to verify that the language change works.

This is a potentially time-consuming step, so the crawler will only attempt to switch to English, German, Spanish, and French.

---

<sup>1</sup><https://github.com/LibreTranslate/LibreTranslate>

<sup>2</sup><https://pypi.org/project/beautifulsoup4/>

<sup>3</sup><https://pypi.org/project/polyglot/>

## Keyword matching

---

The keyword matching algorithm serves as a preliminary classifier of the contents of a page and is used by the crawler as a guide for page exploration.

Pages are categorized into 4 types:

- INDEX – The landing page of the domain
- POLICY – Pages that may contain the privacy policy
- TERMS – Pages that may contain the terms & conditions
- OTHER – Pages likely to contain links leading to PP/TC

All links are given a confidence score for each category (except INDEX) based on how well the link text and URL match with the keyword lists associated with the respective categories. The confidence score defines the order in which the links are visited during page exploration.

To prevent over-exploration of a single page type, there is a hard limit on how many pages of a single type the crawler is allowed to load for each domain.

### 4.1 Matching schema

Keywords are sequences of words defined in a manner that resembles a simplified Backus-Naur form. Each position in the word sequence may be substituted by a word from a predetermined list. For example, a keyword defined as "Privacy [Statement|Policy]" may take the form of either "Privacy Statement" or "Privacy Policy". This design decision was made to simplify the definition of keywords with many synonyms.

Before the keyword matching algorithm can run on a piece of text, the text is stripped of all special characters and formatting. All stop words are also removed, and the text is lemmatized. Stop-word removal and lemmatization

are handled by the `nltk`<sup>1</sup> and `lemmagen`<sup>2</sup> Python packages respectively. Keyword matching is done on both the unlemmatized and lemmatized versions of the text.

We say that there is a matching of keyword  $k$  in text  $t$  if  $k$  appears as a sub-sequence of the words in  $t$ .

### 4.2 Keyword weights & scoring

Not all keywords are treated equally, as some keywords may be more indicative of the content of interest. For example, a link matching only "Privacy" is not as likely to lead to the PP as a link that matches "Privacy Policy". Generally, longer keywords are weighed more than shorter keywords.

Keywords of interest sometimes appear in links that lead to uninteresting content (e.g. there may be a link to an article discussing the privacy policy of some other site). Most of the time, those links will contain text other than the keyword to provide context. Thus it is a good idea to weigh keyword matches in longer text less than keyword matches in shorter text.

With these motivations in mind, we arrived at the following scoring schema

$$f(k, t) := \text{weight}(k) \cdot \frac{\text{length}(k)}{\text{length}(t)} \cdot \log(\text{length}(k))$$

where  $f(k, t)$  is the matching score for keyword  $k$  on text  $t$ ,  $\text{weight}(k)$  is a tailored weight modifier for the keyword  $k$ ,  $\text{length}(k)$  is the length of the keyword, and  $\text{length}(t)$  is the length of the text.

*Note.* Adjusting the value of  $\text{weight}(k)$  requires not only knowledge of the keyword's language, but also common practices regarding the usage of the keyword on the internet. Therefore it is only used for languages that we understand well, namely English and German. For keywords that do not have a weight defined, the default value of 1 is used.

The final confidence score of a link is determined by the total confidence score of both the link text and the URL across all keywords.

---

<sup>1</sup><https://www.nltk.org/>

<sup>2</sup><https://pypi.org/project/lemmagen3/>

# Content classification

---

After keyword matching, pages that have been classified as either PP or TC with a confidence score larger than the minimum threshold defined in `config.py` will skip the content classifiers and go straight to content extraction. This is done to save computation time.

## 5.1 Document preprocessing

### 5.1.1 Content extraction

Before a page's content can be classified, undesirable elements such as navigation, advertisements, social media widgets, etc. have to be removed. Mozilla already solved this problem with the `readability.js`<sup>1</sup> library that filters out distracting elements and keeps only the main text body of the document.

In our crawler, we use the `readabilipy`<sup>2</sup> Python package as a wrapper for Mozilla's `readability.js` library.

The output of `readabilipy` is a much slimmed-down version of the original page, but the result is still HTML that contains a lot of unnecessary elements such as `<div>`s and `<span>`s used for styling and structuring. The `html2text`<sup>3</sup> Python package converts HTML to Markdown, while maintaining structure elements such as titles, basic text formatting, lists, and tables.

The result is a clean Markdown document that is easy to read for both machines and humans.

---

<sup>1</sup><https://github.com/mozilla/readability>

<sup>2</sup><https://pypi.org/project/readabilipy/>

<sup>3</sup><https://pypi.org/project/html2text/>

### 5.1.2 Translation

Because the classifier can only handle English, the extracted content has to be translated to English if it is not already. LibreTranslate is used as the translation back-end. For languages that are not supported by LibreTranslate, the classifier is not used and pages with low confidence scores are skipped.

## 5.2 Privacy policy classification

To handle potential PP pages, we use a random forest classifier based on Amos et al. [6]. Unfortunately, the original classifier mentioned in the paper has been lost, so we are using a faithful recreation offered by the Information Systems Research Lab<sup>4</sup> at the Lucerne University of Applied Sciences and Arts.

The classifier takes as input a single markdown string and decides if the string's content should be labeled as PP.

## 5.3 Terms & conditions classification

In terms of content, variances between TC pages of different websites are vastly greater than variances between PP pages of different websites. We consider the task of creating a classifier for TC far more challenging than creating a classifier for PP. During the time developing the crawler, we have not found any publicly available TC classifiers and any of our attempts to create our own classifier ended with minimal success.

---

<sup>4</sup><https://www.hslu.ch/en/lucerne-school-of-information-technology/research/distributed-ledger-technology/>

## Manual evaluation

---

To assess the accuracy of the crawler during language detection and PP/TC identification, we crawled a list of 100 domains randomly sampled from Tranco [9], a list of top 1 million websites, and compared the crawl results against manually verified data.

This data set was selected for testing because we have access to the crawl result from the previous study by Lodrant [7], which allows us make direct comparisons between the two studies. Furthermore, these domains were never used during the development of the current crawler, and thus the result reflects the accuracy in real use.

Out of the 100 domains, three domains failed to load (DNS error/Timeout) and therefore will be excluded from the result.

### 6.1 Language detection accuracy

Table 6.1 shows the language distribution of the domains used for testing, and the amount of correctly identified domains for each language. The row labeled "NULL" represents all languages not supported by the crawler.

Overall, the crawler correctly identified the language 95.9% of the time, making only 4 mistakes:

- 2 Russian websites were identified as unsupported.
- 1 English website was identified as unsupported.
- 1 unsupported website was identified as English.

### 6.2 PP/TC page identification accuracy

Out of the 97 manually inspected websites, 67 websites had a privacy policy, and 52 websites had a terms & conditions document. By comparing the crawl



## 6.2. PP/TC page identification accuracy

Language	Total	Identified
"en"	65	64
"ru"	8	6
"it"	4	4
"de"	3	3
"fr"	2	2
"es"	2	2
"sv"	1	1
"hu"	1	1
"no"	1	1
"tr"	1	1
"uk"	1	1
"pl"	1	1
"NULL"	7	6
Total	97	93

**Table 6.1:** Language distribution and language detection accuracy of test domains

result against the ground truth, we observed the following:

### Privacy policy identification

- The crawler correctly identified the PPs of 58 websites.
- The crawler correctly identified 27 websites that have no PP.
- On three websites, the privacy policies that the crawler identified were incorrect: 1 URL led to a page that did not exist, and 2 URLs led to pages listing all policies instead of a page containing the PP itself.
- On nine websites, the crawler failed to find the PP:
  - Seven PPs were missed due to insufficient keyword data for some languages (4 Russian, 1 Spanish, 1 Swedish, 1 Norwegian).
  - One PP was missed because it is on the same page as the TC.
  - One PP was missed because it is on a page the crawler is not configured to explore (page listing all legal documents).

Overall, the crawler achieved 87.6% accuracy (TP+TN) in this category.

### Terms & conditions identification

- The crawler correctly identified the TCs of 38 websites.
- The crawler correctly identified 45 websites that have no TC.

- There were no websites for which the identified TC is wrong.
- On 14 websites, the crawler failed to find the TC:
  - Eight TCs were missed due to insufficient keyword data for some languages (2 English, 2 German, 2 Russian, 1 Italian, 1 French)
  - Two TCs were missed because they are embedded in a text box on the website’s registration page.
  - Two TCs were missed because they are on a page the crawler is not configured to explore (page listing all legal documents).

Overall, the crawler achieved 85.6% accuracy (TP+TN) in this category.

### 6.2.1 Comparison with previous study

The crawl data from the previous study was collected before our study began. During the time period between the crawl from the previous study and our crawl, some websites may have updated their page contents and structure.

To provide a fair comparison, we made the following changes to the scoring criteria for the previous crawler:

- URLs that lead to pages that no longer exist but look correct are treated as correct.
- Websites that failed to load during the previous study are excluded from the data set.

As a result, the data set for the previous study contains only 88 domains. On these domains, the crawler from the previous study achieved 75.0% accuracy for identifying PPs, and 63.6% accuracy for identifying TCs. Notably, the results contain a lot of false-positives from irrelevant articles (often blog posts and news articles) being tagged as PP/TC because the old keyword matching algorithm isn’t strict enough.

Document	TP	TN	FP	FN	Acc.
Privacy policy	48	18	7	15	75.0%
Terms & conditions	35	21	21	11	63.6%

**Table 6.2:** PP and TC identification accuracy of the previous study

Comparing the result of the previous study (Table 6.2) and the result of our study (Table 6.3) shows that our study manages to significantly improve PP and TC identification accuracy. Most notably:

- Our crawler finds more PPs and TCs, which means that it will produce a bigger data set for future researches that analyze PP/TC.

---

## 6.2. PP/TC page identification accuracy

---

<b>Document</b>	<b>TP</b>	<b>TN</b>	<b>FP</b>	<b>FN</b>	<b>Acc.</b>
Privacy policy	58	27	3	9	87.6%
Terms & conditions	38	45	0	14	85.6%

**Table 6.3:** PP and TC identification accuracy of our study

- Our crawler produces far fewer false-positives, which translates into cleaner data for future research.

## Deployment & results

---

### 7.1 Crawling data set

To simulate real use, a data set consisting of 73,750 domains was sampled from the Chrome User Experience Report (*CrUX*). The *CrUX* was used for sampling because it contains a ranking of the most popular websites per country. Ruth et al. [10] showed that *CrUX* represents websites popularity much more accurately than other DNS-based rankings such as Alexa Top Million.

*Note.* A bug was discovered in the script used to generate the data set, causing a lot of domains to be included multiple times (up to seven times for some domains). As a result, the data set only contained 46,733 unique domains, with 19,357 domains being included more than once. Because the bug was discovered at the last minute, there was no time to fix the data set and restart the crawl. However, the duplication happened uniformly at random, so the final statistics remain unbiased.

### 7.2 Crawling infrastructure

#### 7.2.1 Hardware

The crawl was performed on a server with 4 Intel(R) Xeon(R) E7-8870 CPUs totalling 80 threads and 512 GB of RAM located within ETH Zürich. Testing done prior to this thesis showed that the server was unstable at high loads, so the server was put on power-saving mode and the crawl was scaled down to only reach 60% average CPU utilization.

### 7.2.2 Software

To deploy the crawler at scale we used Docker<sup>1</sup> to create multiple containers that each run an instance of the crawler. Additionally, a single Docker container was created to run the LibreTranslate back-end. Crawlers communicated with the translation instance over the local network.

We used `docker-compose` to dynamically re-scale the number of containers running the crawler and found that 30 containers was the sweet-spot to keep the average CPU utilization at 60%.

### 7.2.3 Network

Since one of the primary goals of this crawler is to study the effect of the GDPR, the generated traffic has to originate from the European Union. Because Switzerland is not part of the European Union, we used a VPN to route our traffic through Germany.

### 7.2.4 Database

A separate machine (also located within the ETHZ network) running a PostgreSQL database was used to keep track of the domains that needed to be crawled and the crawl results. There were two tables.

- `jobs`: This table was populated before the start of the crawl. Among other things, each entry stores a domain that needs to be crawled as well as the timestamps of start and finish time of that domain. Jobs with no start date and jobs with a start date longer than 8 hours ago and no end date will be randomly selected for crawling.
- `websites`: This table stores all the crawl results and is filled gradually during the course of the crawl. It is worth noting that each entry does not store the entire PP/TC of each domain, but only the URL where the PP/TC was found. The extracted PP/TC contents (both HTML and Markdown) are stored in the file system of the machine that executed the crawl.

## 7.3 Statistics

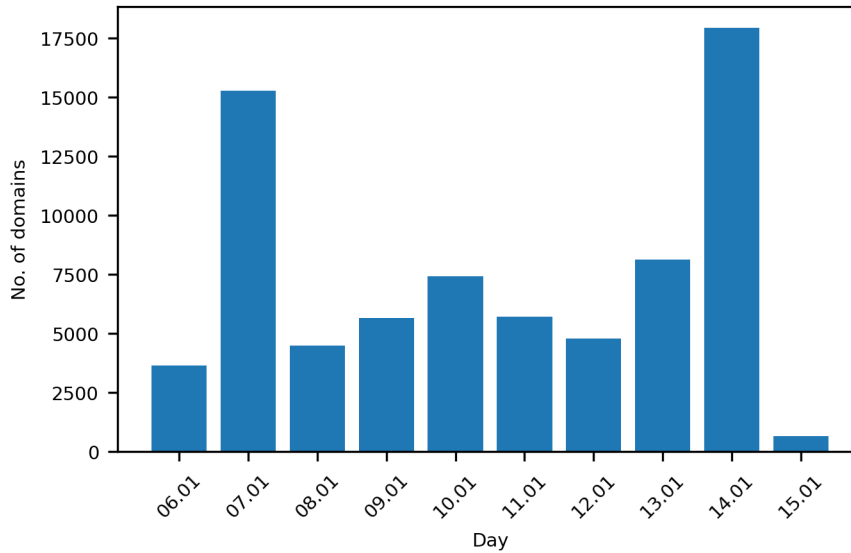
### 7.3.1 Crawling speed

The crawl was started on 06.01.2023 and finished on 15.01.2023. During that time, 73,712 out of 73,750 domains were processed at an average speed of 8'373 domains/day, peaking at 17,929 domains processed on one day on

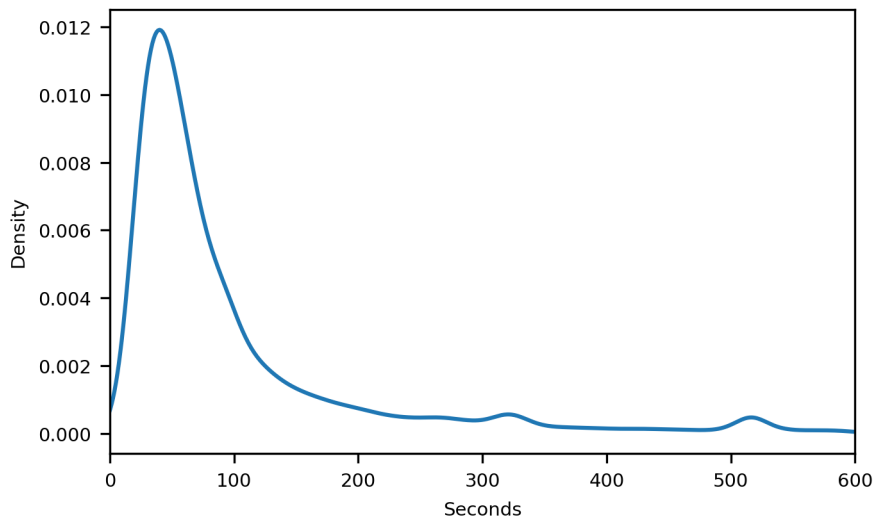
---

<sup>1</sup><https://www.docker.com/>

January 14th. The crawler rejected 48 domains from the data set because of invalid formatting.



**Figure 7.1:** Number of domains crawled per day



**Figure 7.2:** Time (in seconds) to crawl a single domain

From January 8th onward, due to unknown reasons, the crawler was no longer able to establish a connection with Google's Safe Browsing API<sup>2</sup>.

<sup>2</sup><https://developers.google.com/safe-browsing/v4/>

Because the crawler uses the API to check if a domain is safe before crawling, this issue caused the crawler to spend a long time waiting for a response from the API. The request would timeout, and the domain flagged as unsafe. This caused the overall crawling speed to slow down drastically. On January 13th we marked the domains flagged as unsafe for re-crawling, removed the API query from the crawling pipeline, and restarted the crawler.

### 7.3.2 Page load statistics

Loading the index page of each domain was not always successful. Table 7.1 gives a detailed overview of the load status of all domains. In total, the crawler encountered an error for 4,019 domains, with timeouts and HTTP errors being the most common. Timeouts hint at overloaded servers or overloaded networks, and the most likely cause for HTTP errors is anti-bot measurements such as Cloudflare.<sup>3</sup>

Index status	Count	Count (%)
"OK"	69,593	94.4%
"TCP_ERROR"	258	0.4%
"TIMEOUT"	1,841	2.5%
"HTTP_ERROR"	1,609	2.2%
"BAD_CONTENT_TYPE"	19	0.0%
"DNS_ERROR"	199	0.3%
"NULL"	193	0.3%

**Table 7.1:** Load status of landing pages during the crawl

### 7.3.3 Language distribution

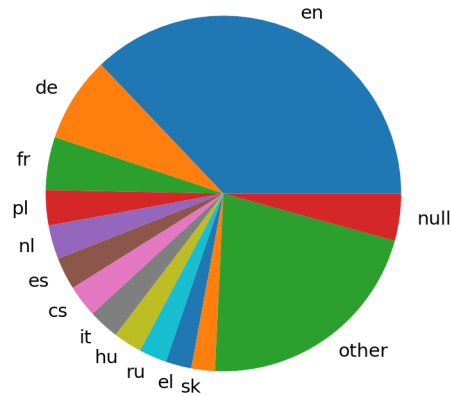
Fig. 7.3 shows the language of detected languages of successfully crawled domains. Only the 12 most popular languages are shown, while the other languages are grouped into a single category we call "other". Unidentifiable languages are labeled as "null".

Unsurprisingly, English is the most common language with 25818 domains (35.0%), followed by German with 5413 domains (7.3%), and French with 3348 domains (4.5%). The least popular language (not shown in the figure) is Macedonian with only 12 domains identified.

For domains which the crawler failed to identify the language, manual analysis showed that common causes were:

- Multiple languages being used on the website.

<sup>3</sup><https://www.cloudflare.com/>



**Figure 7.3:** Distribution of detected languages of successfully crawled domains

- Use of non-standard words, also known as "slangs".
- Little to no text detected on the website (either because there was no text, or because the text was embedded in <iframe>s).

### 7.3.4 Number of PPs and TCs found

Language	Total domains	PP found	TC found
"en"	25818	14254	10803
"de"	5413	4412	2638
"fr"	3348	1714	1170
"pl"	2231	1171	365
"nl"	2150	1364	476
"es"	2035	1323	577
"cs"	1981	922	73
"it"	1956	1296	457
...	...	...	...
Total	73712	32829	20132

**Table 7.2:** Number of privacy policies and terms & conditions found per language

Out of the 73,712 websites crawled, we detected a PP on 31,356 websites, and a TC on 20,094 websites.

Our PP-discovery rate is significantly higher than the 37.2% Amos et al. [6] found for the top 1,000 Alexa websites. Even though our results are not directly comparable due to a difference in the domains crawled, the discrep-



ancy is significant enough to warrant an investigation. We found 2 potential reasons for the discrepancy between our result that that of Amos et al.:

- Their study only considers PPs in English and ignores other languages.
- Their data is older. Amos et al. showed that an increased percentage of websites include a PP as time goes on.

# Discussion

---

## 8.1 Limitations

### Link discovery

Currently, the crawler only explores pages by following links defined by the `<a>` HTML tag. This works for the vast majority of websites. Websites that use less-common methods to link pages (such as using JavaScript `onClick()` events to load a page) will have their links ignored, leading to potentially missed PP/TC.

### Content extraction

It is not rare for websites to have their PP/TC and other legal documents on the same page, which our crawler is not equipped to deal with. PP/TC extracted from these pages will include potentially unwanted content. While this might be an issue depending on the type of analysis the resulting data set is used for, it is something worth mentioning.

## 8.2 Future work

### Use sitemap data to discover PP/TC pages

Websites may have their PP/TC in places not easily accessible from the index page. The crawler sometimes has to follow multiple links to access the documents. Increasing the search depth of the crawler will increase the likelihood of finding the PP/TC on such websites, but it will come at a cost to the run time.

Many websites have a sitemap to help search engines index the website's content. One could utilize this overview page to find otherwise hard-to-reach

PP/TC. However, sitemaps come in many standards, and implementing them all could turn out to be time-consuming.

### **Develop classifiers for PP and TC with support for multiple languages**

The crawler used a PP-classifier that only works for English. To use the PP classifier, other languages first have to be translated. During testing we noticed that around one-third of the CPU time was used by the translation back-end. Having classifiers trained for other languages would significantly improve crawling speed.

## **8.3 Conclusion**

We created a web crawler that uses a combination of keyword-based and ML-based classification to identify privacy policies and terms & conditions, then extracts the documents in Markdown format that is easy to analyze and interpret. The crawler achieves high accuracy for identifying privacy policies (87.6%) and terms & conditions (85.6%) while supported all major European languages.

Because web crawling is such a complex topic, we did not have time to implement all the features and optimizations we planned for our crawler, leaving still a lot of room for improvement.

---

## Bibliography

---

- [1] W. Thode, J. Griesbaum, and T. Mandl, ““I would have never allowed it”: User perception of third-party tracking and implications for display advertising.”, in *ISI*, 2015, pp. 445–456.
- [2] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang, “Smart, useful, scary, creepy: Perceptions of online behavioral advertising”, in *Proceedings of the Eighth Symposium on Usable Privacy and Security*, ser. SOUPS ’12, Washington, D.C.: Association for Computing Machinery, 2012, ISBN: 9781450315326. DOI: [10.1145/2335356.2335362](https://doi.org/10.1145/2335356.2335362). [Online]. Available: <https://doi.org/10.1145/2335356.2335362>.
- [3] European Parliament, Council of the European Union, *Regulation (EU) 2016/679 Of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)*, <http://data.europa.eu/eli/reg/2016/679/2016-05-04>; Last accessed on: 2021.02.06, Apr. 2016.
- [4] —, *Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications)*, <http://data.europa.eu/eli/dir/2002/58/oj>; Last accessed on: 2021.02.06, Jul. 2002.
- [5] T. Linden, R. Khandelwal, H. Harkous, and K. Fawaz, *The privacy policy landscape after the GDPR*, 2018. DOI: [10.48550/ARXIV.1809.08396](https://arxiv.org/abs/1809.08396). [Online]. Available: <https://arxiv.org/abs/1809.08396>.
- [6] R. Amos, G. Acar, E. Lucherini, M. Kshirsagar, A. Narayanan, and J. Mayer, “Privacy policies over time: Curation and analysis of a million-document dataset”, in *Proceedings of the Web Conference 2021*, ACM, Apr. 2021. DOI: [10.1145/3442381.3450048](https://doi.org/10.1145/3442381.3450048). [Online]. Available: <https://doi.org/10.1145/3442381.3450048>.

- [7] L. Lodrant, “Designing a generic web forms crawler to enable legal compliance analysis of authentication sections”, en, Master Thesis, ETH Zurich, Zurich, Jan. 2022. DOI: [10.3929/ethz-b-000534764](https://doi.org/10.3929/ethz-b-000534764).
- [8] P. M. Kast, “Enforcement bots: Nothing can block us! Automating website registration for GDPR compliance analysis”, Bachelor Thesis, ETH Zurich, Zurich, Mar. 2021.
- [9] V. L. Pochat, T. V. Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation”, in *Proceedings 2019 Network and Distributed System Security Symposium*, Internet Society, 2019. DOI: [10.14722/ndss.2019.23386](https://doi.org/10.14722/ndss.2019.23386). [Online]. Available: <https://doi.org/10.14722/ndss.2019.23386>.
- [10] K. Ruth, D. Kumar, B. Wang, L. Valenta, and Z. Durumeric, “Toppling top lists: Evaluating the accuracy of popular website lists”, in *Proceedings of the 22nd ACM Internet Measurement Conference*, ser. IMC '22, Nice, France: Association for Computing Machinery, 2022, pp. 374–387, ISBN: 9781450392594. DOI: [10.1145/3517745.3561444](https://doi.org/10.1145/3517745.3561444). [Online]. Available: <https://doi.org/10.1145/3517745.3561444>.

## Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

**Title of work** (in block letters):

PRIVACY OBSERVATORY:  
COLLECTING PRIVACY POLICIES AND TERMS OF SERVICE ON A REGULAR BASIS

**Authored by** (in block letters):

*For papers written by groups the names of all authors are required.*

**Name(s):**

TRUONG

**First name(s):**

HOANG LONG

With my signature I confirm that

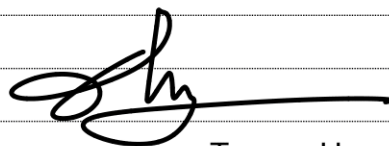
- I have committed none of the forms of plagiarism described in the '[Citation etiquette](#)' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

**Place, date**

Dübendorf, 18.01.2023

**Signature(s)**



Truong Hoang Long

*For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.*