De transnationale Universiteit Limburg is een uniek samenwerkingsverband van twee universiteiten in twee landen: de Universiteit Hasselt en Maastricht University.

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculteit Wetenschappen
School voor Informatietechnologie
master in de informatica

Masterthesis

Automating Subject Access Requests

Sven Theunissen
Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :
Prof. dr. Peter QUAX

COPROMOTOR :
Prof. dr. Wim LAMOTTE

BEGELEIDER :
De heer Mariano DI MARTINO

2020
2021

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculteit Wetenschappen
School voor Informatietechnologie
master in de informatica

Masterthesis

Automating Subject Access Requests

Sven Theunissen
Scriptie ingediend tot het behalen van de graad van master in de informatica

PROMOTOR :
Prof. dr. Peter QUAX

COPROMOTOR :
Prof. dr. Wim LAMOTTE

BEGELEIDER :
De heer Mariano DI MARTINO

Acknowledgements

During the writing and research of this thesis, I have received lots of support and advice. Firstly, I would like to thank Prof. dr. Peter Quax for allowing me and giving me the opportunity to work on this thesis. I definitely also would like to thank my mentor Mariano Di Martino for giving constructive advice and feedback, helping me whenever I asked and proofreading the text.

Additionally, I want to thank family and friends for giving support and encouragement throughout this year and studies in general.

Abstract

In May 2018 the European unions General Data Protection Regulation (GDPR) went into effect. The right of access within the GDPR gives citizens the right to request a copy of their data via a Subject Access Request (SAR). SARs can be sent using any method. Some companies offer a web-form through which the data can be requested but if this is not available the SAR can be sent via, for example, e-mail. Relatively few users actually request their data, and it can be assumed to be even fewer for requests that have to be done manually via e-mail. The goal is to develop a proof-of-concept browser extension that automates (part of) the right of access process. The searching of the actual privacy policy and extracting contact information are the two main automated steps. Limited assistance was added in the actual correspondence of mails (starting template and user initiated response generation). Possible ways to add automatic response mail suggestions are discussed but left to future work due to lack of data. The browser extension is then tested by way of a usability test with ten participants. The results suggest, with the current implementation, that the layman user might not get much use out of it. There are a few jargon and domain knowledge barriers that still exist within the process and extension. The argument is made that non-layman users might be the optimal target audience either way since there exist multiple other usability issues outside the main process such as the understanding of the actual data.

Contents

Chapter 1

Introduction

In 2016 the European Union introduced the General Data Protection Regulation (GDPR) [1]. The aim was to unify the data protection regulation within the EU. With this unification data subjects gained the same privacy rights everywhere in Europe. On the 25th of May 2018 the regulation became enforceable. The GDPR gave citizens of Europe additional rights regarding their online privacy. Some examples include the right to be forgotten, the right to object to processing or, most relevant, the right of access.

The GDPR was, however, not the first mention of the right of access. Since 1995, the Data Protection Directive (DPD) also gave European citizens the right to access their data [2]. The DPD was a framework on which member states could base their national privacy laws. One main issue of the DPD was the inconsistency between member states. Additionally, technologies such as big data analytics were only in their early stages during the creation of the DPD. The GDPR was created, in part, to alleviate these issues [3].

The right of access, specifically, is defined in article 15 of the GDPR [1]. This gives any data subject the right to obtain confirmation that there is personal data being processed by the data controller. If this is the case, the data subject has the right to access this personal data. Additionally, the data controller has to give access to information regarding the purposes of the processing, the categories of personal data that is being processed, the time period for which the data will be stored and more. As defined by article 12(3), when a data controller received such a request from a data subject, the data controller has one month to respond to this request. The data subject can request to exercise their right by any means, such as e-mail or via post. When a request arrives, the data controller needs to use any reasonable measures to verify the identity of the data subject requesting access (Recital 64). What this, in essence, means is that the data controller may request additional info such that the data subject can verify their identity.

Wile the right of access is certainly a positive for the citizens of the EU, a right is not really useful if no one actually exercises it. One main way of communicating these rights from data controller to data subject is the privacy policy of said data controller. However, very few people actually read the privacy policy and even when they do, they miss important details in over 90% of the cases [4]. You can't really blame the average user for this, though. McDonald et al. calculated that if a user were to read all privacy policies for every new website they visit, each user would spend 244 hours per year reading policies [5]. This study was done in 2008, it would be reasonable to assume that the situation probably has not improved in the meantime. Even if a user would spend their time reading these privacy policies, many would not understand them. Proctor et al. found that people who have less than a college level education will generally not be able to comprehend the privacy policies [6].

Past work has looked at making privacy policies or privacy rights more usable. These will be discussed in Section 3. The privacy right this thesis will focus on is the right of access. The aim will be to automate the process of exercising this right. This automated process is implemented by a proof-of-concept browser extension. First, Section 2 will look at the user survey that was performed. In Section 4 some background is given on machine learning concepts that will be used in Section 6. Section 5 details how the process of right of access will be split into different steps. These steps will then be tackled one by one in Section 6. The developed application is evaluated using a usability test and with user submitted URLs in Section 7. Section 8 will discuss some smaller topics such as third-party trackers, FLoC and a standardised alternative. Lastly, Section 9 is a conclusion and self-reflection.

Concretely, the following research questions will be researched:

- Is it possible, and what is needed, to make the right of access more accessible and user-friendly?
- To what degree are users able to understand and work with the developed proof-of-concept browser extension?

These questions will be answered via a feasibility study. A proof-of-concept application (browser extension) will be developed. Multiple techniques will be combined as to solve the (sub-)problems of automating the process of exercising the right of access. The proof-of-concept application is then evaluated in a usability test. The essential parts of the application are summarised as follows: finding a privacy policy, extracting info from the policy and communicating with the company. Additionally, the trust to put into a company will be quantified to some degree via a "trust rating". Contribution is made by searching for and improving upon existing algorithms which are then combined to achieve the different sub-goal in each step of the right of access process. The combined system results in the earlier mentioned proof-of-concept browser extension.

Chapter 2

User survey

To acquire a sense of the stance a user has regarding the topic of subject access requests (SAR) and an application of this kind, a user survey was performed in early November 2020. In this survey, the respondents were asked multiple opinion-based questions regarding their knowledge relating to the GDPR and the "right of access". Secondly, respondents were also asked their position regarding an application such as the one proposed in this thesis. The survey took less than 10 minutes to fill and was available for a week, after which entries were no longer allowed.

Participants were gathered from the following groups:

- (ex-)Computer science students from Hasselt University. (18)
- The Reddit community of r/Belgium[1]. (58)

First and foremost, it should be noted that the sample taken for this user survey does not represent the average citizen or computer user. There is a relatively heavy bias towards both users in IT-related occupations and users that generally spend more time online. This is represented by 100% of the first group (students) selecting IT as their area of study or work. In the second group 24 (41%) of the participants selected IT, the remaining participants were split over many subjects such as sciences, education, law or accounting. The participants were also asked for their current employment status (student, (self-)employed or retired/unemployed). In this case group one had a split of 17 students to 1 employed. Group two was more split with 42 participants (72.4%) stating they are (self-)employed and 16 reporting they are students.

[1]https://www.reddit.com/r/belgium/

Results

Across both groups, when asked to rate their knowledge on the GDPR legislation, 47.37% stated to know some ins and outs, 14.47% stated to have some ideas, 26.32% stated to know quite a lot, 9.21% said that GDPR is within their area of expertise and only 2.63% said to have heard about GDPR somewhere. Even knowing the sample set is biased towards more IT-knowledgeable users, the amount of people who say it is within their area of expertise is surprisingly high, coming down to 7 out of 76 participants. Less surprisingly, 89.48% of participants claimed to be aware of their privacy online by some or a great extent. In this group of 89.48%, 63.23% of participants still agreed or strongly agreed that they should be more aware of their privacy.

After this, participants were asked if they know what the “right of access”entails. To this 71.05% stated to know, 17.11% was not sure and 11.84% did not know. The participants were then presented a short explanation as to the meaning and process of exercising their right of access.

When queried on the participant’s feeling regarding the difficulty of exercising their right, just under half (48.68%) of participants said it was difficult but not impossible. 44.74% said that the process is doable. Similarly, the amount of participants that have or have not requested their data in the past was also almost split 50/50 (52.63% have, 47.37% have not). The participants were also given an optional field to explain why they have or have not exercised their right in the past. While most people who responded said to have sent a SAR out of curiosity for the data a company has, one specific user did so in a situation where an organisation shared their contact info without their consent. On the side of people who have not requested their data, one user mentioned they never knew they had the right to do so.

The most divisive question was probably whether the participant would use a browser extension that would assist them in exercising their right of access. Figure 2.1 shows the percentage of participants for each answer. The fact that around 42% of participants would definitely or probably use the browser extension is certainly encouraging. The other 31.58% who would probably not or definitely not use the application could seem discouraging at face value. However, similarly to the previous question the participants were allowed to give an optional motivation behind their answer. From the participants who responded “Definitely not” (9), all who filled in the optional field mentioned the trust factor that is involved in introducing a third party (namely the extension) into the process. This third party could easily introduce a privacy concern.

Within the participants, there were many mentions of the extension having to be open source to garner enough trust. With the more IT-biased participant sample, it is not really a surprise that this topic was mentioned. It does, however, raise the significance of trust the extension needs from the user. This leads to the conclusion that it would be preferable if as much as possible of the process happens and stays on the users computer or within their mailbox. From the user's perspective, any data that is sent to a server essentially ends up in a black box, the user has very little knowledge of what actually happens with their data on these servers.

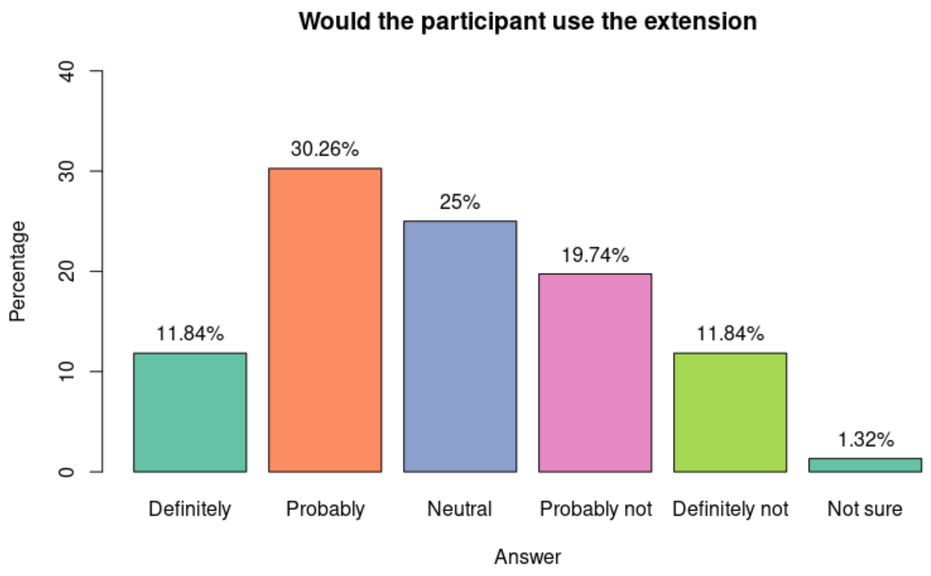


Figure 2.1: Would the participant use the proposed application:
Percentage of participants for each answer

Chapter 3

Related work/similar applications

At the time of writing and to our knowledge there exists no public work that attempts to achieve the same kind of automation of right of access. There exists, however, multiple other works that assist the user in controlling their online privacy, making privacy policies more approachable, automating other privacy rights or attempting the automation in some other, unknown, way. This section describes multiple of these works as to capture a subset of similar work.

Summarization of privacy policies

One angle to look at simplifying privacy policies for users is to summarize, to some degree, the content of the policy. Two main branches of achieving the summarization are on one hand a crowdsourcing approach and on the other an automated approach that uses data mining. Each branch has its own advantages and disadvantages.

Crowdsourcing ToS;DR[1] or "terms of service; didn't read" is a community project created in 2012. The service provides a crowdsourcing database of (privacy) policy ratings and labels. Members of the community read, rate and discuss the content and earlier ratings. The database is freely available online in the forms of the website, a publicly available API and a web browser extension available for most major web browsers (Google Chrome, Mozilla Firefox, Microsoft Edge and opera).

[1]https://tosdr.org/, accessed on 11 dec. 2020

Similarly, justdeleteme.xyz[2] also offers a public database of ways or links to delete accounts at a growing list of companies. In a kind of different vein of crowdsourcing are TOSBack[3] and its reimplementation TOSBack2 [4]. TOSBack tracks changes to privacy policies by crawling these daily. The project is crowdsourced since the policies that are crawled and what to extract from these policies is specified by the contributors.

While the human aspect of the crowd-sourcing greatly increases the general accuracy of the ratings by having an open discussion about the exact meaning of, often, nuanced policy statements. Many of these project suffer from scaling issues. While the amount of contributors can certainly be considerable, it will hardly be enough to cover the majority of the World Wide Web. The most popular services will without a doubt be covered but the more fringe services and, for example, websites in other languages will not be covered. In the specific case of TOSBack, the system is automated once the websites has been entered into the system. Save for websites completely overhauling the code of the policy page, causing the extraction to malfunction. The other services will have to manually update the ratings if the policy were to change.

Data mining The other branch that was mentioned earlier is an automated manner of processing privacy policies to present the users with a more consumable presentation of the generally difficult legal documents. Two relatively similar previous works regarding this topic are PrivacyGuide [7] and PrivacyCheck [8].

Both applications rely upon machine learning techniques to rate certain aspects of a privacy policy. To give a concrete example, PrivacyCheck shows the user the level of risk regarding the way a company uses e-mail addresses. Where the highest level of risk, in this case, is the information being shared with third parties. The risk level is shown using a colour-coding system (green, orange and red) to give the user a simple overview of a website their policy in a single glance. The main differentiating point between the two works is the set of aspects that are measured for privacy policies. PrivacyCheck addresses more specific measures such as the handling of e-mail addresses, credit card numbers and data regarding children. PrivacyGuide on the other hand handles more generic points based on the exact regulations written in the GDPR laws. Some examples include: Account deletion, Privacy settings and Data collection.

[2]https://justdeleteme.xyz, accessed on 11 dec. 2020
[3]https://tosback.org, accessed on 11 dec. 2020
[4]https://github.com/pde/tosback2, accessed on 11 dec. 2020

A third work in line with this work is Privee [9]. Privee makes the argument that, as mentioned before, on one hand crowdsourcing solutions might lack in participation and have issues scaling to the size of the internet. However, automated approaches are not a solution without drawbacks. In this work, S. Zimmeck and S. Bellovin [9] found that the automated approach had similar issues with the ambiguity of natural language. On a positive note, it was also observed that ambiguity of privacy policies decreased over time. This was attributed to more enforcement of the need for clarity by government bodies and also that writers of policies tend to base the content on previous texts or similar statements. Privee was implemented in a proof-of-concept browser extension which analyses privacy policies and summarises them into a set of labels for some categories such as collection of data and ad tracking. To achieve this, Privee uses both ToS;DR and a machine learning system, using the latter when a ToS;DR entry is not available.

Probably the most advanced of these applications is Polisis [10]. This work consists of two applications that were built on top of the Polisis system. The works mentioned in previous paragraphs often use a relatively limited set of policy-attributes and one or two machine learning systems to decide on a certain attribute (for example: one machine learning system to decide which parameter is handled in that piece of text and another for the severity). Polisis on the other hand has 10 high-level classes and 122 fine-grained classes to classify sections of a privacy policy into. This greatly expands the amount of info that is gathered from a policy. Using this system, both a summarization application and a chatbot application were implemented. The summarization application does the summarization into the privacy icons based on those defined by Disconnect.me [11]. These icons are then presented to the user so the policy can be interpreted in a single glance. The second application named pribot, which is also available online[5], allows the user to ask free form queries about the privacy policies. This allows users to quickly ask specific questions about what they want to know, without having to parse through the other (at that moment) irrelevant info. It also allows the user to get complete information about the relevant attribute instead of a summarized version.

Standardisation of privacy policies

There was, in the past, an attempt at standardising privacy policies in a way that would lend itself to automatic interpretation of said policies. Namely the Platform for Privacy Preferences Project or P3P [12]. However, the protocol never saw widespread support as it was only ever implemented in the Internet Explorer and Edge browsers.

[5]https://pribot.org/bot, accessed on 17 dec. 2020

The standard was implemented in the sense that if certain requirements were not met, the site would not be allowed to place cookies. To circumvent this restriction, many sites would simply set up a P3P policy that misrepresented the actual privacy policy [13]. There was no real legal binding to the policies written using the P3P system [14]. Thus misrepresenting the real privacy policy to evade the restrictions put up by Internet Explorer, was a relatively minor risk companies certainly were willing to take. More general support from user-side for P3P, however, was also implemented in a browser extension for Internet Explorer [15]. Due to the aforementioned lack of support, the P3P working group was closed in 2006.

Further work related to P3P was made to create a graphical representation of the content of P3P policies [16]. Using the Expandable Grids [17] visualisation technique, R. Reeder et al. presented the P3P policy to users. This visualisation can be seen in figure 3.1. The expandable grid visualisation is, in essence, a grid which has by default collapsed rows and column headers showing a summarization at the cross-points of the two. The rows and columns can be expanded to show more or less data. This should allow for both a quick glance of top level information shown in the grid and more in detail information processing by opening the collapsible elements. However, R. Reeder et al. found that this is not necessarily the truth when this technique is applied to P3P privacy policies. Via a user study it was found that, using the expandable grid, users performed neither faster nor more correctly answering comprehension questions about the privacy policy when compared to natural language texts. Subjective satisfaction measurement also showed that many users dislike the expandable grid. This result was, among other things, attributed to the lack of a focal point in the grid and the usage of words that are still unfamiliar or confusing to the average internet user.

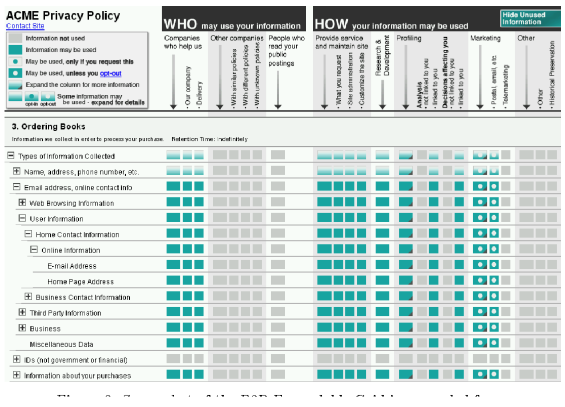
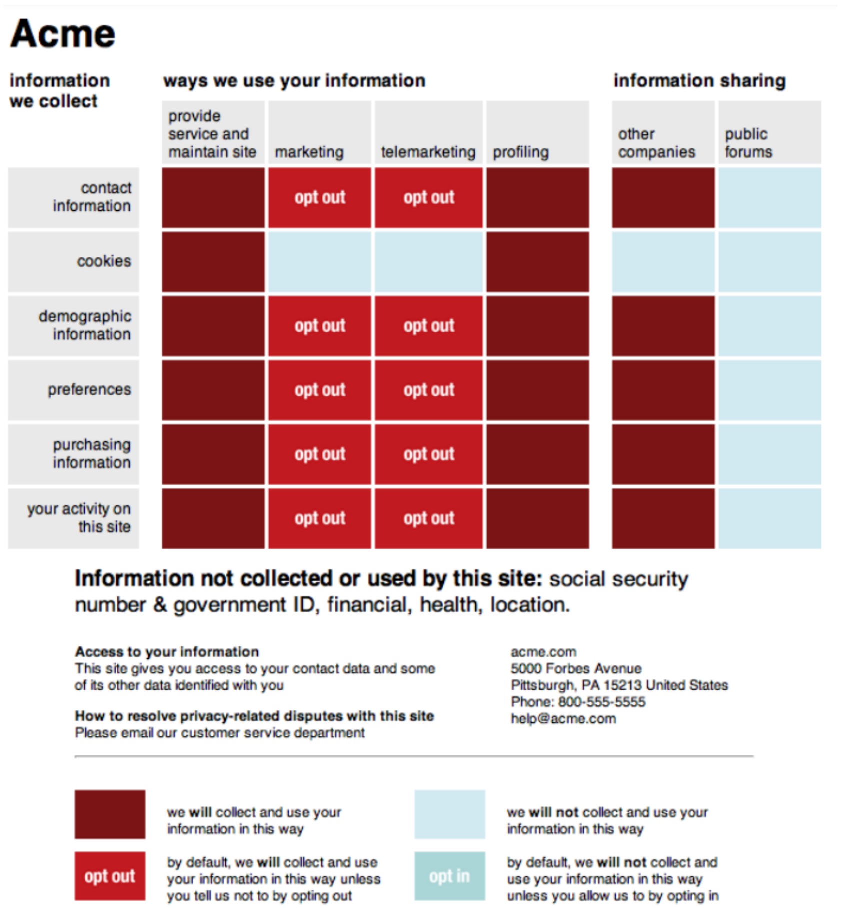


Figure 3.1: Screenshot of the P3P Expandable Grid in expanded form. [16]

In line with the results found by the P3P expendable grid, P. Kelley took inspiration from existing systems to present privacy policies to users in both a standardised and user-friendly manner. This system being food nutrition labelling [18, 19]. While still using a grid to display the information to the user. The amount and complexity of the information was greatly reduced. This was achieved in the following ways, among others: simplifying the terms used, hiding completely unused datapoints and cutting down on the amount of icons that are used. In a user study tied to this work, it was shown that the standardised formats performed better than the full text policy, both in the amount of correct information-finding tasks and subjective preference by users. An example of the short table is shown in figure 3.2.


Acme

information we collect	ways we use your information				information sharing	
	provide service and maintain site	marketing	telemarketing	profiling	other companies	public forums
contact information	■	opt out	opt out	■	■	□
cookies	■	□	□	■	□	□
demographic information	■	opt out	opt out	■	■	□
preferences	■	opt out	opt out	■	■	□
purchasing information	■	opt out	opt out	■	■	□
your activity on this site	■	opt out	opt out	■	■	□

Information not collected or used by this site: social security number & government ID, financial, health, location.

Access to your information
This site gives you access to your contact data and some of its other data identified with you

How to resolve privacy-related disputes with this site
Please email our customer service department

acme.com
5000 Forbes Avenue
Pittsburgh, PA 15213 United States
Phone: 800-555-5555
help@acme.com

■	we **will** collect and use your information in this way	□	we **will not** collect and use your information in this way
opt out	by default, we **will** collect and use your information in this way unless you tell us not to by opting out	opt in	by default, we **will not** collect and use your information in this way unless you allow us to by opting in

Figure 3.2: An example of a standardised “food label” short table. [19]

Automating privacy rights

Besides previous human-powered works (e.g. justdeleteme.xyz), there are multiple other automated works. Two applications produced by private companies are Tapmydata[6] and Mine[7]. Tapmydata offers users a mobile app from which they can search and contact companies. Via this app, users can initiate a chatroom-like conversation with the company. Companies, however, need also show initiative when it comes to Tapmydata. The platform can only be used by users if the company has integrated the platform, either via the Tapmydata platform or an integration with existing CRM platforms. This makes the system streamlined, but it might halt widespread usage. Similarly to P3P, if there is no support from the industry, the application would not work. However, Tapmydata seems to have garnered some support and already has a some amount of support from multiple big companies. For example, at time of writing, YouTube and Amazon UK are both available in the app.

Mine is relatively comparable to the application that is being developed in this thesis. By scanning the users e-mail inboxes, a list of companies that hold their data is generated. These companies are then listed for the user including the type of data that the company probably has about the user. The platform also supports the possibility of automatically asking the company to delete any and all data. The manner in which the correct contact-address is found is currently unknown.

While not necessarily a privacy right, Gmail's automatic unsubscribe functionality aims to give a user some power in their relations with a company. The idea existed first as an unsubscribe button in the header of e-mails. In 2018, however, the functionality was upgraded to a suggestion system where if a user had not opened mails from a certain sender in a month or so, Gmail would automatically suggest to unsubscribe in the general mailbox. The actual unsubscribe action could be performed in two ways: Gmail could be searching in the mail-body for the unsubscribe link that is often present in the footer or, when added by the sender, using a List-Unsubscribe header in the mail (defined by RFC2369).

[6]https://tapmydata.com, last accessed on 24 march 2021
[7]https://saymineapp.com, last accessed on 24 march 2021

Chapter 4

Background

The following subsections aim to provide some background knowledge on machine learning that will be used in Section 6.1.2. The context in which machine learning will be used is that of determining whether a webpage is a privacy policy.

High-level description

It might be interesting to first position the type of machine learning algorithms that will be used in the wider space of machine learning. A main differentiation in machine learning algorithms is supervised, unsupervised and reinforcement algorithms [20]. Supervised algorithms attempt to solve classification and regression problems. The latter of which tries to predict continuous values, given some value X which value Y should be expected. A real world example could be that of weather prediction, based on many parameters such as the temperature of the past days and air pressure, what temperature can we expect today. Classification problems are similar to regression in the sense that it tries to predict some value based on the input parameters, the output values, however, are discrete values or classes. An example would be: given info about an insect, is it a cricket or a beetle. The main property of supervised learning is that the training data needs to be labelled. In other words, to create the classifier we need a set of data for which the result is already known. Unsurprisingly, it is this class of machine learning algorithms that will be used for the issue of deciding whether a webpage is a privacy policy or not. The other two categories of machine learning, logically, do not use labelled data. This could, for instance, be used in cases where labelling a big data set is simply not feasible or too expensive. Unsupervised algorithms mainly deal with problems such as clustering, where the aim is to group inputs together, and autoencoders, which aim to compress and decompress data with as little loss of info as possible. Reinforcement learning tries to find the most optimal way to reach some goal. The most self-explanatory example would be that of video games: an artificial intelligence could aim to find the most optimal way to reach the end of a level.

On figure 4.1 the high-level function of a supervised classifier is shown. During the first phase a classifier model is created using the machine learning (ML) algorithm. This algorithm is fed the label and a vector of features extracted from the input. The step of extracting a feature vector could be just as important as the rest of the algorithm due to the fact that bad feature selection hinders the performance of the ML algorithms. To give an example, counting the amount of each letter in a word could be useful to possibly infer the language the word is from (since some letters get used more in particular languages, such as 'W' in Polish). However, if we were to use this feature extraction to, for example, guess the subject of a text, the results would probably be suboptimal.

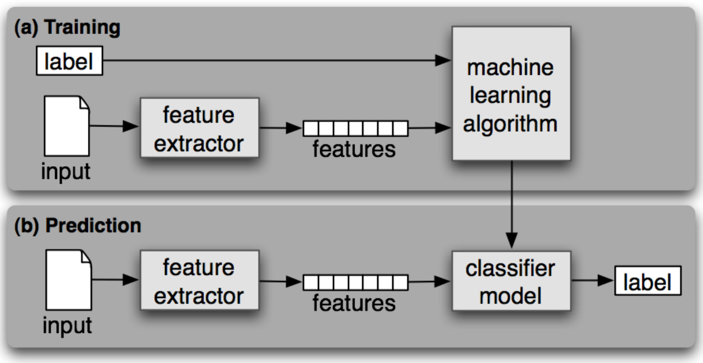


Figure 4.1: A high-level overview of how supervised classification works. [21]

Machine learning algorithms: theory

Due to the process of feature extraction, text documents can be transformed into a form that is usable in the machine learning algorithms. This mostly means that the text has been transformed into numerical values which are much more interpretable for computers than natural language. One very simple interpretation of how a vector of values could be used to calculate a class (which in the case of this thesis is “privacy policy” or “not privacy policy”) would be:

$$y = w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + ... = w_0 + \sum_{i=1}^{n} w_i * x_i \tag{4.1}$$

Where x_i are the elements of the input feature vector and the different w_i's are weights. The exact value of these weights are what the ML algorithm aims to calculate. The resulting value y could then be used to classify the input document based by, for example, deciding that $y > 0$ is one class and $y <= 0$ is the other. This type of classification is a linear model for binary classification. This can be imagined as a line splitting the space in two in the case of an input with a single dimension. In higher dimension situations this line becomes a plane or hyperplane.

Logistic regression, which in contrast to its name is a classification algorithm, is one such way to create weights and classify vectors. Logistic regression is used to model the probability of a class or the probability of an event happening. This can be written as $p(y = 1)$ or the chance of the class being 1. Using this the odds ratio can be defined $\frac{p(y=1)}{(1-p(y=1))}$, which is simply the ratio of an event happening to it not happening. Taking the logarithm of this function is called a logit function: $logit(p) = log\frac{p(y=1)}{(1-p(y=1))}$. This logit function relates to equation 4.1 in the following way: $logit(p(y = 1|X)) = w_0 + \sum_{i=1}^{n} w_i * x_i$. The probability $p(y = 1|X)$ is the chance of the result being class 1 given the feature vector X. Within logistic regression, however, it is exactly this probability that is of interest, it is this probability that needs to be predicted. Inverting the logit function gives the logistic function:

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

Where z is the result of equation 4.1. When the logistic function is plotted over values of z, the result is as is shown in figure 4.2. As the value of z increases, the result nears 1 and as the value decreases, the result nears 0. This can be interpreted as a probability or in other words $\phi(z) = p(y = 1|X)$. This probability can then simply be transformed into an actual prediction by outputting class 1 if $\phi(z) >= 0.5$ and class 0 in the other case. This probability is certainly of interest to some use cases. One particular example of this would be weather forecasting, with logistic regression it can not only be predicted whether or not it could rain but also the chance of this happening [22].

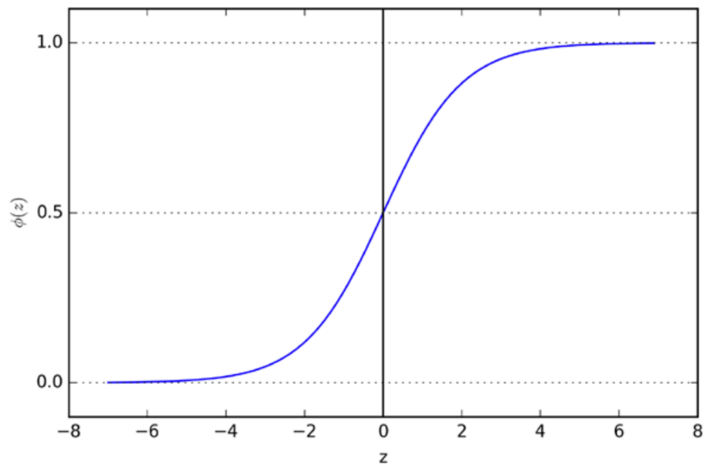


Figure 4.2: A plotted logistic function [22]

Currently, there is an equation to take the input feature vector and calculate a resulting probability of an event happening. There is, however, still the issue of deciding the weights of the classifier. One, possible, core idea behind deciding on weights is gradient descent/ascent. Figure 4.3 shows a visualisation of this process. Firstly a cost function $J(W)$ is defined. This cost function will model the error of the classifier with the current weights. One very simple example would be the sum of squared errors $J(W) = \frac{1}{2}\sum_i(y_i - \hat{y}_i)^2$, where y_i is the actual class and $\hat{y}_i$ is the prediction.

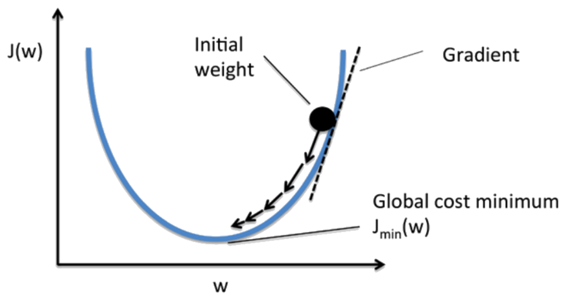


Figure 4.3: Visualisation of gradient descent [22]

When taking the derivative of this cost function, the gradient can be calculated. As the name gradient ascent/descent implies, this gradient will be used to calculate the update to the weights. The weights can then iteratively be updated by calculating $W = W - \eta * \Delta J(W)$, where η is a learning rate which can be chosen before training and $\Delta J(W)$ is the gradient. This specific equation is for gradient descent, if the aim was to maximise some cost function, the gradient would be added instead of subtracted and the process would be gradient ascent. Each iteration, the gradient of the cost function is calculated (based on a set of training feature vectors and labels) and the weights are then updated using this gradient after which the process can be repeated with these new weights. This repeats some number of repetitions or until a certain threshold of cost is hit.

All that remains now is to choose a cost function for the logistic regression classifier. Since the logistic regression classifier is essentially based on the probability of a class, it could be best to maximise the total probability or likelihood $L(W)$:

$$L(W) = \prod_i (\phi(z_i))^{y_i} * (1 - \phi(z_i))^{1-y_i} \tag{4.2}$$

What is essentially happening in this equation is the following. If the actual class of the entry is 1 the first half of the product is used since the exponent of the second half would be 0. Naturally, if the class is 0 the second half is used. With this distinction, the likelihood is simply the product of the probabilities of each training-entry being its correct class. To make equation 4.2 easier to calculate, the logarithm is taken. This removes the need for many multiplications and exponentiation. This leads to the following equation:

$$l(W) = \sum_i (y_i) log(\phi(z_i)) + (1 - y_1) log(1 - \phi(z_i)) \tag{4.3}$$

Equation 4.3 is simply called the log likelihood. This equation would have to be used in gradient ascent but can be simply transformed into a gradient descent equation by negating the whole equation. The results in a cost function that can be used to calculate the weights for logistic regression using gradient descent: $J(w) = -\sum_i (y_i) log(\phi(z_i)) + (1 - y_i) log(1 - \phi(z_i))$.

One last change to the algorithm can be made to alleviate the issue of overfitting. This issue develops itself when a machine learning classifier is tuned too much to the training dataset. The classifier will reach almost perfect accuracy on the training data but will not perform as greatly on unseen or more general data. A solution to overfitting is regularization, which adds an extra penalty to the cost function. This penalty prevents the weights from becoming too large. Additionally, regularization handles high correlation between different features in the input vector and helps filter out noise. One way to add regularization is $L2$ regularization:

$$L2regul. = \frac{\lambda}{2}||W||^2 = \frac{\lambda}{2}\sum_i w_i^2$$

Where λ is the regularization parameter which controls the amount of regularization that is applied. This regularization is simply added to the cost equation. An alternative regularization is $L1$ regularization, which is calculated as follows: $L1regul. = \lambda \sum_i |w_i|$. The final cost function will thus look as follows:

$$J(w) = [-\sum_i (y_i)log(\phi(z_i)) + (1 - y_i)log(1 - \phi(z_i))] + \frac{\lambda}{2}||W||^2$$

The derivative of this cost function can then be used to calculate the weights using gradient descent.

This approach to training is only the tip of the iceberg when it comes to deciding the weights. There are multiple other manners in which they can be calculated.

Chapter 5

Structure of the application

The goal of the application is to assist the user in the process of requesting their data from a company. The high-level process is visible in figure 5.1. The obvious first step is for the user to choose a company to which they want to send a subject access request. As required by Section 1 Article 12 of the GDPR [1], any data controller must communicate any and all info regarding Article 15, among others, in a concise, transparent, intelligible and easily accessible form. This is generally done via the privacy policy on the website of the data controller. Logically, the next step is searching for the privacy policy of the data controller. From the policy the relevant info, such as where to contact the controller, is extracted. This most often boils down to an e-mail address or a URL to a web-form. Figure 5.1 shows the situation where an e-mail address is used to exercise the right of access. After sending the Subject Access Request, there is a 1-month period in which a data controller must respond. It is possible to send a reminder mail when this 1-month deadline nears. Besides that, data controllers can ask for extra information needed in correctly identifying the user. Lastly (if provided in such a way) a user needs to download their information from the attachment.

Another part of the process would be deciding, or at least hinting at, a “trust rating” for the website or company in question. This trust rating would be useful for users to decide what authorisation to send to the company. For example, a user does not want to send sensitive data such as their ID card to an untrustworthy data controller. This trust rating is a rather subjective measure but can assist more novice users in making decisions as they attempt to exercise their right.

There is also a need for the application to be privacy-friendly. For this reason as much as possible of the process should happen locally on the user's device. As seen on figure 5.1, all but the policy interaction happens within the browser extension. The reasoning behind the policy interaction happening on a server is that, for example, part of the process is visiting multiple webpages and finding the policy. It is not hard to assume most if not all users would be confused when their browser suddenly starts redirecting automatically.

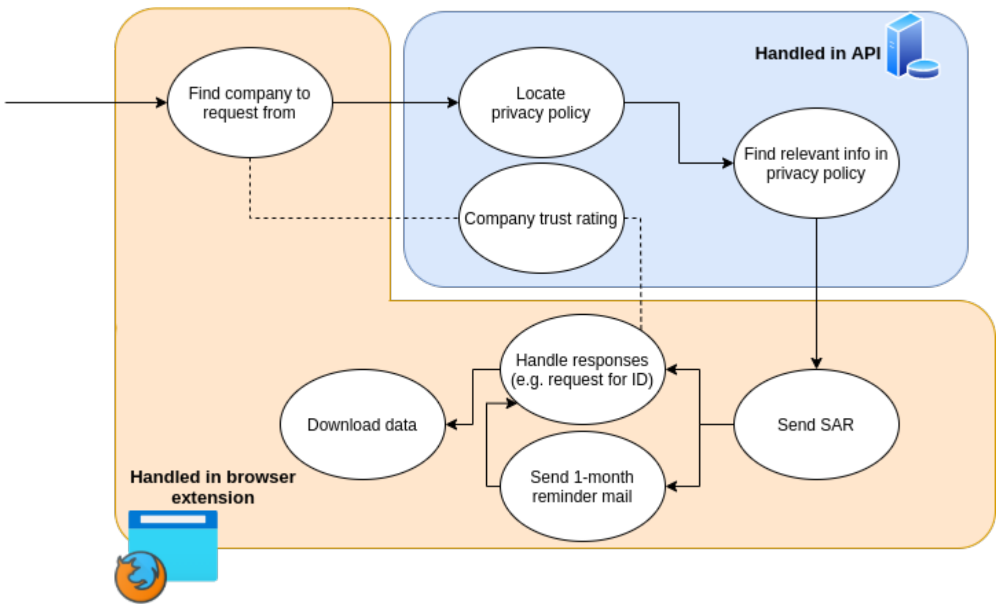

Handled in API
Find company to request from
Locate privacy policy
Find relevant info in privacy policy
Company trust rating
Handle responses (e.g. request for ID)
Download data
Send SAR
Send 1-month reminder mail
Handled in browser extension

Figure 5.1: High level overview of the SAR process.

Chapter 6

Function of the application

The following subsections discuss how each step, as laid out in Section 5, was or could be implemented. Section 6.1 discusses how the privacy policy can be searched starting from a URL that, for example, was found in an e-mail sent to the user. This is also evaluated in Subsection 6.1.3. Section 6.2 describes how information, such as the e-mail address to contact, is extracted from the policy once it was found. The trust rating is discussed in Section 6.3. Lastly, Section 6.4 first discusses what previous work found to be common in the e-mail correspondence (6.4.1). After that it will be discussed how partially automated correspondence could be implemented (6.4.2).

6.1 Locating the privacy policy

The first step of the process aims to locate the privacy policy of a website starting from a generic link to the website. This generic link could in the simplest case be the homepage. This process has been split into two main components. On one hand there is the webpage crawler that produces a stream of candidate webpages/URLs and on the other is the classifier that will determine the likelihood of the webpage being a privacy policy.

6.1.1 Crawling the webpage

There exists some literature on this subject in the context of creating a privacy policy corpus. Zimmeck et al. [23] created a corpus of Android app privacy policies. The major upside of creating a corpus from mobile apps is that the Play Store page usually has a direct link to the privacy policy. However, it was found to not always be the case. Some

URLs linked to a home page or a landing page with links for different countries. For this reason a limited crawl is performed to download a set of candidate pages. The exact details of this limited crawl were not specified, the assumption can be made that this is a recursive crawl to a certain depth.

Kumar et al. [24] also created a corpus of privacy policies to use as input in an opt-out choice machine learning classifier. The top 500 on the U.S. Alexa list in 2018[1] was used as starting list for this corpus. Every URL on the homepage is then also stored to create the list of candidate webpages. Both previously mentioned papers use a browser instance (headless Firefox [23] and Selenium[2] with Geckodriver[3] [24]) to download the pages instead of, for example, a simple curl or wget fetch. The reasoning for this is that the browser instance will fetch and execute the JavaScript code that is attached to the webpage HTML. This allows (most) dynamically loaded content to be instantiated. This is in contrast to, for example, wget which does not contain a JavaScript engine, thus the JavaScript won't be executed.

While the approaches of Zimmeck et al. [23] and Kumar et al. [24] certainly work, we would argue there are some things that could be improved. Additionally, there also exists a feature of the average webpage that can be exploited. To shortly reiterate, the application that is being built receives a URL extracted from, in this case, an e-mail. The aim is to find the privacy policy page starting from this URL. One main issue is that the amount of URLs on a webpage can be very large. Secondly, there exist some websites that, as mentioned earlier, do not directly link to the privacy policy but rather have a landing page or some short summary before the actually full-text privacy policy. This means the crawler might have to go two or three layers deep in a "tree" of linked pages. Combining these two issues means there is an almost exponential growth of candidate pages as each linked page can also have a big list of URLs. This leads to the conclusion that some shortcuts are needed to more quickly hone in on more likely privacy policy candidates. There is a feature of webpages that will be useful to achieve this. Since the introduction of privacy related legislation, close to all websites have included a URL to their privacy policy, privacy landing or privacy summary page on their homepage. Most often this is a link in the footer or navigation bar of the webpage.

[1]https://www.alexa.com/topsites, last accessed 10 june 2021
[2]https://www.selenium.dev, last accessed on 11 june 2021
[3]https://github.com/mozilla/geckodriver, accessed 28 may 2021

To exploit this feature, the implementation used in the API of the application employs a priority queue. The URLs found on a webpage are given a priority based on the context of the URL. The highest priority is that of URLs that contain the word “privacy” and are located within the footer of the webpage. The second-highest priority is for URLs containing “privacy” in the general body of the page, which includes any navigation bars. This distinction of priority was made to shortcut the situation where the website in question contains, for example, a blogpost about privacy. The privacy URL in the footer will more often be the privacy policy over that in the body of the webpage. All other URLs are given a lower general priority. The general flow of the crawler is thus as follows. The flow can also be seen in figure 6.1. The priority queue is created with the input URL as first entry. Each entry in the queue is then opened in a Selenium instance with the chromium driver. Selenium is used for the same reasons as mentioned in the other work of [23]. The webpage source is checked as to whether or not it is as privacy policy (see Section 6.1.2 for details). If the page is not a privacy policy, the URLs are extracted and inserted into the queue according to the priorities mentioned earlier. If the current page is a privacy policy the main processing loop can be halted since we should have the correct page.

There still remain some edge cases that need to be handled to achieve more accurate results. Firstly, the domain of the webpage should also be taken into consideration before processing it further. The need for this exception can be best illustrated using an example. A webpage can contain a link to some other page on a vastly different domain. If by chance the different domain happens to contain a link to a privacy policy and it gets processed before the target domain, this privacy policy on the wrong domain would then be chosen as the result of the crawling operation. To circumvent this situation, URLs will only be added to the queue if the domain from which it was gathered is the same as the base domain with which the queue was initialised. Simply discarding any URL on a different domain would be problematic since there exist situations where the privacy policy of a website is hosted by, for example, the parent company’s website.

The second edge case could occur in the situation where there are multiple privacy policy webpages. This is most prominent in websites with a user base in the United States. The state of California has also introduced some legislation regarding privacy (namely the CCPA or California Consumer Privacy Act). This leads to the situation where some websites have a different privacy policy for California residents and international users. If the California policy is processed before the international version, the Californian version is simply returned as the result. The first step in solving this edge case is to simply check all URLs with a high priority after the main loop is halted due to a result being found. These high priority URLs are those with "privacy" in their name. In this manner there is a (almost always) short list of URLs that are probably a privacy policy of one form or another (e.g. a Californian and an international policy). The second step is then to differentiate between these policies and decide which is the correct one. To achieve the most accurate result, it was decided that a top 3 should be collected. This top 3 list would then be presented to the end-user to have them decide upon the most correct policy.

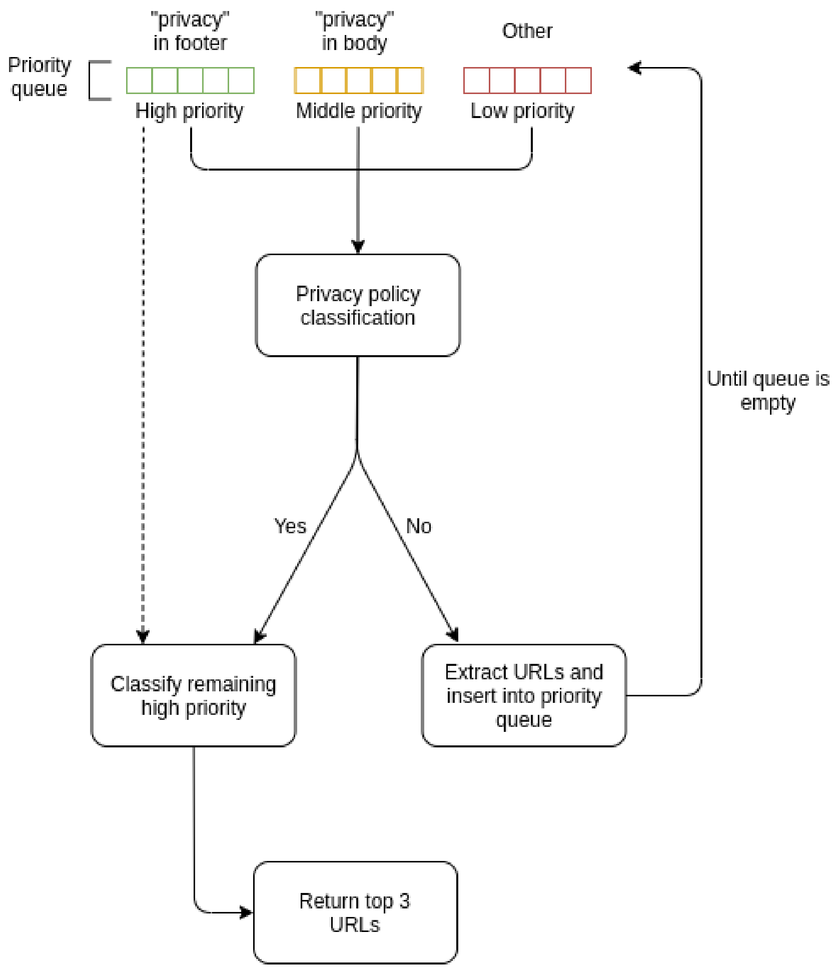
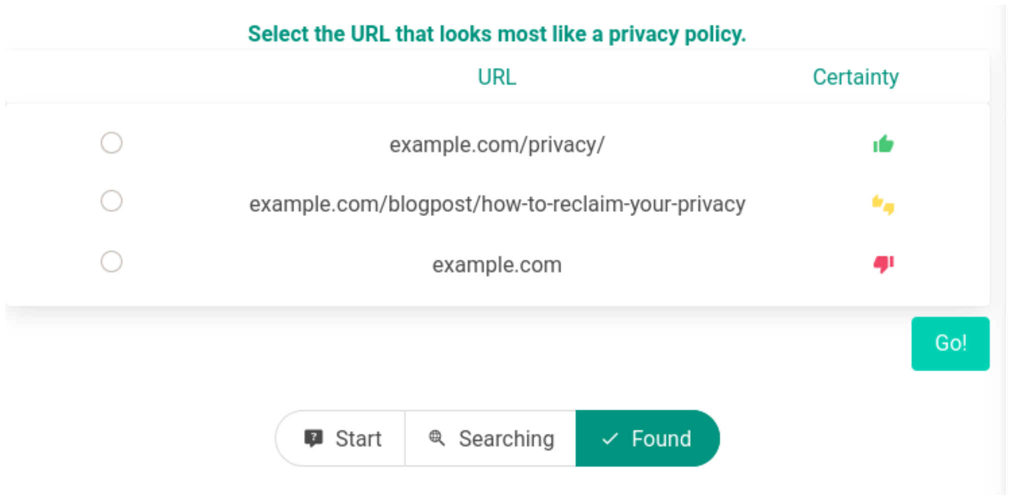


Figure 6.1: High-level crawler flow diagram

This top 3 list shifts, without a doubt, more responsibility on the user. This does not mean the user is left to their own devices, however. To assist the user in selecting the most probable privacy policy URL, a certainty rating is given to each option. This can be seen in figure 6.2. Depending on the score given to the URL a different icon is presented (the source of this score is discussed in Section 6.1.2). If there is a high probability the page is a privacy policy (P>75%) a green thumbs-up is shown. A low probability (P<50%) is shown by a red thumbs-down. Intermediary scores (75%>P>50%) are shown with combined thumbs and a yellow colour.


Select the URL that looks most like a privacy policy.

	URL	Certainty
○	example.com/privacy/	👍
○	example.com/blogpost/how-to-reclaim-your-privacy	👍👎
○	example.com	👎

Figure 6.2: Presentation of the top 3 URL candidates

6.1.2 webpage classification

In the crawling process described in Section 6.1.1 a classification of privacy policies versus pages that are not privacy policies needs to be made. There are multiple ways to achieve such classification. The theoretically easiest manner of achieving this would be to manually create rules for which to check a document. To give an example, if the aim was to classify e-mails about delivery dates of packages, one could add rules that search for words like: “delivery”, “shipping” or “tomorrow”. The main advantage of this type of system would be that a human would be able to comprehend the decisions being made. Changes or updates to specific rules can be easily made and errors can be fixed based on knowledge of the subject. There are of course some disadvantages too. The major one being that expert knowledge is required to create these rules. The creation of these rules takes a lot of time due to the analysis and testing when the classification is concerned with a complex or wide topic. Lastly, this method would not scale well. If there would be need for more rules or more classes, the changing or moving around of rules could have unknown effects on the interaction between these rules.

The second option is then to look at automated ways to create classifiers. There are some machine learning algorithms available with which text classifiers can be created. Three popular algorithms are logistic regression, Support Vector Machine (SVM) and naïve bayes classifier. Logistic regression will be used in the following sections. Some details as to how logistic regression works were discussed in the background section, Section 4. One feature of the logistic regression is the probability that results from the classification. This probability is useful for the top 3 list that is presented to the user.

Feature extraction for webpages

To start classifying, the webpages need to be processed in a manner that would enable the creation of a feature vector. A webpage is in essence an HTML document with a layer of CSS applied on top. While this CSS is important for the browser to present the HTML elegantly, the content is completely captured within the HTML text. For this reason the HTML can, mostly, be handled as if it were a natural language text document.

Before the text is transformed into a feature vector, it needs to be cleaned and standardised to make the later process more streamlined. One of the more obvious steps when working with HTML documents is to remove any and all HTML tags. Since HTML tags are very prominent, they could throw off the importance of other words. The HTML tags would introduce noise within the feature vectors. Besides that, all letters are converted to lowercase, webpage quirks such as "&amp;" and any punctuation are removed. More interesting changes to the text are the expansion of contractions, removal of stop words and lemmatisation. In English text, some word constructions are combined into shorter versions where some letters are omitted. Examples are "hasn't" being a contraction of "has not" and "they're" being a contraction of "they are". By expanding these contractions or, in other words, replacing the contractions by their non-contracted counterpart the full set of words in a document is reduced by combining words that have the same meaning. This reduces the amount of worthless info within the data of the documents. The removal of stop words has the same use as that of removing the HTML tags, reducing the amount of noise in the data. While stop words such as "always", "but" or "if" are obviously useful in conveying thoughts in text, this is not necessarily the case when looking at the text on a subject basis. The stop words have no effect on the high-level subject of a document since these words are used in basically every context or subject. Lastly is lemmatisation, in most natural languages some words have inflected forms. An example of an inflection is conjugations of verbs. Another example is plural forms of words ("wolves" to "wolf").

Now that the data is cleaned, the text can be transformed into a feature vector. Two widely used manners of feature extraction for text document classification are bag-of-words (BoW) and TF-IDF [22]. Bag-of-words is the simplest of both options. In essence, a set of all words in the documents is created. This could be seen as a list containing each word or token in the text once. The bag-of-words is then a vector with an element for each token containing a count of how many times the word appears in the document. The count or frequency of a word or term in a document can be represented as $tf(t, d)$, with t the term and d the document. This vector of counts can then be passed into the machine learning algorithm or classifier as the input. There are, however, some more nuances that can be applied. Within a corpus of text documents, some words are much more common than others (even if the stop words have been removed). Words such as the conjugations of "to be" simply are much more common. These common words add very little discriminatory data for the classifier, since they are common in every document. To reduce the count or influence of these common words or to increase the influence of uncommon words, TF-IDF can be used. TF-IDF stands for "Term Frequency-Inverse Document Frequency". As the name implies, the document frequency is taken into account in the feature vectors created by TF-IDF.

$$tf\text{-}idf(t, d) = tf(t, d) \times idf(t, d)$$

$tf(t, d)$ is the same term frequency count as in bag-of-words. The $idf(t, d)$ term is the inverse document frequency. This is calculated as follows [25]:

$$idf(t, d) = log\frac{n_d}{df(t)} + 1$$

Where n_d is the total number of documents and $df(t)$ is the number of documents containing the term. The addition of 1 is to prevent terms that are in every single document from having an *idf* of 0 and as effect a *tf-idf* of 0. It is easiest to get an intuitive understanding of the purpose of the *idf* by looking at an example. Let's say there are a total of 100 documents and a term t_1 appears in 10 documents. The tf would then be multiplied by 11, causing the term to more easily have an effect on the classification process. Inversely, a word appearing in 90 documents would receive an *idf* of 1.004

Machine learning algorithms: in practice

To create a classifier using supervised machine learning, a dataset of privacy policies and non-privacy policies is needed. Luckily, there exist some previous work, such as within the usable privacy project, which have made their data available[4]. To train the classifier, the dataset by Bannihatti et al. [24] was used. This dataset was, at the time of writing, the most recent and was preferable since the dataset is based upon the Alexa top 500[5] instead of, for example, mobile app policies which some of the alternative datasets focus on. Some manual filtering was needed since the dataset contained some non-privacy policy entries (e.g. a warranty page). Secondly, Selenium was used to get the matching homepages to each of the privacy policies. This resulted in a dataset of 366 homepages and 366 privacy policies. Lastly, to cover for the edge case of EULA's, around 75 EULAs were manually searched and added.

These pages were parsed using Mercury Parser [26], which removes unnecessary data such as navigation headers from the HTML pages. See Section 6.2.1 for detailed info on this subject. The pages were then pre-processed to create a feature vector, as was explained in the previous section. The lemmatization functionality and stop word corpus of the Natural Language ToolKit (NLTK) python module [27] were used for the relevant parts of the pre-processing. There exist many implementations to train a logistic regression classifier. One such implementation is sci-kit [28]. Additionally, sci-kit has implementations for the TF-IDF vectorization.

These implementations of TF-IDF and logistic regression, such as many machine learning algorithm, have multiple parameters to tune. These parameters are, for example, the regularization parameter or the type of regularization. By tuning these parameters a better or worse classifier can be found. Finding the optimal set of parameters can be achieved in multiple ways. The most simple way to tune the parameters is exhaustive grid search. With exhaustive grid search, the list of options is defined for each parameter and all combinations are tried. The best set of parameters will logically produce the best classifier. Other techniques include random search [29], which tries to evade wasting computation time on useless parameters by randomisation, or more advanced techniques such as the particle swarm [30]. Due to the relatively limited size and simple nature of the webpage classifier, exhaustive grid search was used.

[4]https://www.usableprivacy.org/data, accessed 18 feb. 2021
[5]https://www.alexa.com/topsites, last accessed 10 june 2021

One aspect left out in the search for the best parameters, is how is to decide which classifier is the best. As with many things within machine learning, there are many metrics by which this can be decided. For classifiers, a first metric one could think of would be accuracy. Dividing the number of correct classifications by the total number of entries. The issue with simply looking at the accuracy is that it might not express the specific type of error that is being made. Let's say, for example, a classifier is being made that does a first, fast screening for a certain very infectious virus. One could ask the question as to whether false positives or false negatives are a bigger issue. In the case of the virus, many false positives would probably be less of an issue than many false negatives. This with the train of thought that a false positive could get fixed with a more accurate (be it more expensive) test while a false negative could cause the patient to unintentionally infect many people since they think they are safe. Figure 6.3 shows these error types in a confusion matrix.

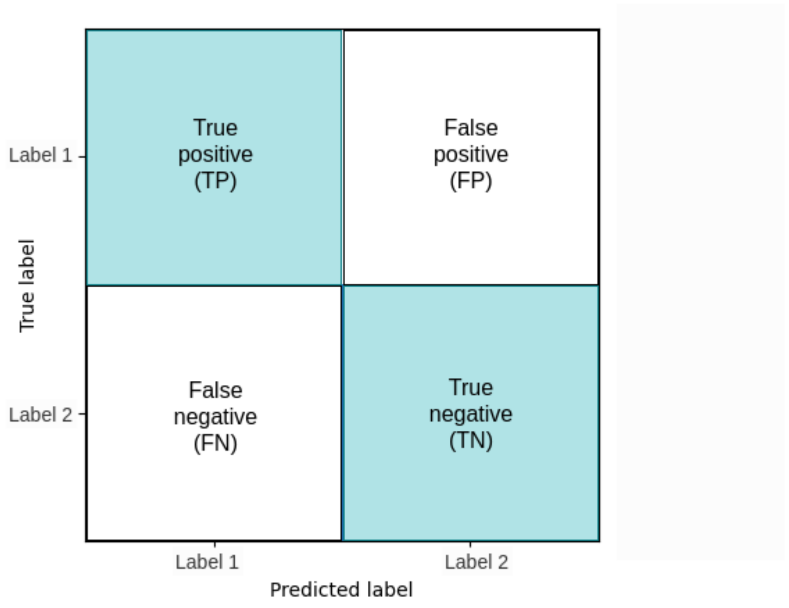


Figure 6.3: Confusion matrix error types

Using these error types, there are multiple combinations possible to create metrics by which a classifier can be judged. First, though, accuracy can be calculated from this via: $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$. When taking the example about the virus from earlier, it could be interesting to see from the number of people who actually have

the virus, how many were correctly detected. This metric is called the recall and is calculated by $Recall = \frac{TP}{TP + FN}$. A counterpart to recall is precision, which calculates the proportion of positives that were correct. $Precision = \frac{TP}{TP + FP}$
These two metrics, precision and recall, on their own also have their pitfalls. To give an example, if the classifier would decide to only mark 1 case (from the test set) as positive, the precision would be 100%. Similarly, if the classifier would always classify a person as having the virus, recall would also be 100% since there are no false negatives. Each metric has a use in some specific cases. However, if the classifier does not really lend itself to either of these metrics, another metric could be useful. The F1-score is a metric that takes an in-between solution of precision and recall. This is also called the harmonic mean between recall and precision.

$$F1\_score = 2 * \frac{precision * recall}{precision + recall} = \frac{TP}{TP + 0.5 * (FP + FN)}$$

Sasaki [31] gave a great example to show the advantage of a harmonic mean over the standard arithmetic mean for classification issues. Imagine a fingerprint recognition system with a precision and recall of 1.0 and 0.2 respectively. In this example, the system would work terribly since 80% of the fingerprint would wrongly be marked negative. The arithmetic mean of these values is 0.6. The harmonic mean or F1-score would be: $2 * \frac{1 * 0.2}{1 + 0.2} = 1/3 = (0.3333...)$. When trying to measure the performance of the system, the harmonic mean conveys the shortcoming more than the arithmetic mean.

Evaluation of the classifier

When training and evaluating a classifier, it is normal practice to split the full dataset into parts where, for example, 70% will be used to train the classifier and the remaining 30% is used to test it after training. Figure 6.4 shows the confusion matrix of one such test. The F1-score of this particular run (using privacy policy as the positive) is 0.9814 or 98.14%. This, however, shows a small distortion due to it being a single test. Since the split of the dataset is random, the test dataset can contain a lot of easy samples. To account for this a k-fold cross-validation is used. In this validation, the dataset is split into k chunks or folds and a fresh model is trained on all but one of these chunks with the remaining one being used as a test set. The k metrics are then simply averaged to get a mean performance of the model. For the privacy policy classifier, with $k = 5$, the resulting mean is 0.9705 or 97.05%.

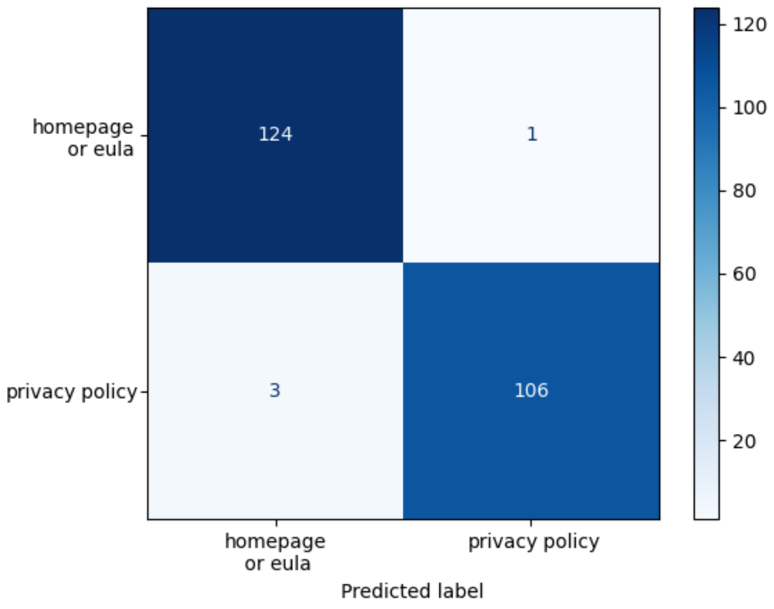


Figure 6.4: Confusion matrix result of the trained classifier

As a final note, during the introduction of this section, three popular classifiers were mentioned (Logistic regression, SVM and naïve bayes). All three could be tested to choose the best performing one However, since this classifier is only one step in the process and the result with logistic regression is generally good, The other algorithms will not be expanded upon.

6.1.3 Evaluation

Now the aim is to evaluate the effectiveness of the crawler and classifier when put together, to locate the privacy policy of a random site. This was achieved by randomly picking 150 websites from the top 500 Alexa sites[6]. These were then one by one fed into the crawler and the results were manually checked as to whether the result was correct. If the result was incorrect, it was also checked why the application failed. A summarisation of the results can be found in table 6.1. The following goes into more detail as to the reasons behind the results.

Of the 150 websites that were tested, 128 returned the correct result. The remaining 22 had some widespread reasons as to failing. Four of the 22 fails were due to websites automatically redirecting to the webpage in a different language. Since the test was run

[6]https://www.alexa.com/topsites, last accessed 10 june 2021

Table 6.1: Evaluation results of the classifier and crawler combined

Type	Amount	% of all
Success	128	85.3
Fail	22	14.7
• Language redir.	4	2.7
• Connection refused	4	2.7
• async JS	4	2.7
• CAPTCHA	4	2.7
• Application fail	6	4.0
– Mercury Parser	3	2.0
– Classifier	2	1.3
– Crawler	1	0.7

from Belgium, the websites in question redirected to either a Dutch or French version. This is not an unsolvable problem, the main issue is that the classifier has been trained on privacy policies written in English. The fix could thus be to add classifiers that were trained on other languages. Four webpages, at the time of the test, either partially or completely refused connection. One specific case was due to the website not allowing EU visitors. It can be assumed that this is done to simply evade the requirements of GDPR. A second case within this group refused only some parts of the webpage being loaded. The webpage would load the HTML, CSS and some images but then the remaining parts were refused. These remaining parts included the footer which contained the privacy policy link. The precise reason as to why some parts of the site refused connection and some did not is unknown. However, it seems this was a temporary issue as at a later date this issue did no longer present itself on the site in question. The last two cases refused connection completely.

For four of the websites, part of the content was loaded in separately via JavaScript. Two of these websites only loaded the link of the privacy policy after a button was pressed or the footer containing the link is added only after scrolling down to the bottom of the page. The third and fourth case of this group loaded the content of the privacy policy asynchronously after loading the page. This last case could be solved by introducing a wait time after each page load. This would, however, introduce a hit to the performance since this wait would happen for each and every page that is loaded. A trade-off could be made here for performance versus catching these few cases.

A last set of cases where the cause was some particularity of the webpage all revolved around CAPTCHAs. These cases made up four of the 22 unsuccessful ones. One particular case seemingly only triggered the requirement for a CAPTCHA whenever the crawler ran in headless browser mode. For the cases where CAPTCHA got triggered in non-headless mode (3 cases), the execution was paused and the CAPTCHA manually completed. This led to successful discovery of the privacy policy in all but one of these three cases. This one case where it was unsuccessful seems to have implemented the CAPTCHA incorrectly and would never exit the CAPTCHA page, even upon completion of the CAPTCHA. If a CAPTCHA were to be triggered in a real situation (a user searching, for example), the CAPTCHA would block the crawler from effectively finding the privacy policy.

The remaining 6 unsuccessful cases were due to failures in the application. Each subportion of the classification system are to blame for the failure of detecting some cases. Three cases were unsuccessful due to Mercury Parser [26] failing to extract anything meaningful from the privacy policy page. Which in turn caused the classification to classify incorrectly. As a sidenote, one of these cases also encountered a CAPTCHA but even after manually passing this, the search still failed in the manner described.

The classifier misclassified two homepages as being privacy policies. This caused the process to be cut short, before any links could be crawled. The last remaining case was due to the crawler portion of the process. In this case the webpage did not include a link directly containing "privacy". This then means that more general pages are searched before the actual privacy policy is found and classified. By chance one of these general pages contained links to the privacy policy of every single partner company. It is certainly commendable for the website owner to be so thorough in their website, however this did hinder the crawler as it is currently implemented. In the end, the crawler did encounter and correctly classify the actual privacy policy but the privacy policy of every partner company was also included in the result. This could, by chance, push the actual privacy policy out of the top 3. A fix to this error could be to rate pages in the same domain (artificially) a bit higher than policies off-site. This of course also has downsides such as that if the top 3 contains, for example, a technical blog post and an actual off-site policy, the blog post would be higher in the top 3 than the policy.

6.2 Extracting data from the privacy policy

Once the privacy policy link is found, the aim would be to extract useful data from the document. Most important is the contact information of the company in question. This is often an e-mail address or a URL to a web-form. More basic information (such as the relevant section of the policy) is also used and sent to the user.

6.2.1 Extracting the textbody

Modern webpages contain much more than simply the main content (e.g. the text of a news article). Often a template or boilerplate is used into which the content is inserted. This boilerplate contains, for example, the navigation bar, the logo of the website or a footer. D. Gibson et al. found that in 2005 almost 40%-50% of webpages was boilerplate [32]. The percentage also seemed to grow by around 6% each year.

As might be expected, this boilerplate content is not interesting for the purposes of the extension. This is true for both the classification of webpages described in Section 6.1.2 or the data extraction described in the following subsection. For this reason it could be interesting to extract the main content. Which in the context of a privacy policy would be the body of text. There exist multiple ways to possibly achieve this, two of which are used within the application and will be discussed in more detail. The first of which was already in mentioned in Section 6.1.2, namely Mercury Parser [26].

Mercury Parser is a web parser developed by Postlight. This project was originally an API but was open-sourced in 2019. The original goal was to automatically generate AMP pages. While there is no academic literature on the inner workings of the application, the open-source nature allows for a decent enough insight. The framework allows for custom parsers to be written, such that the content can be extracted more accurately. There is, however, also a generic content extractor. The generic extractor searches for the most probably HTML node based on a score given to these nodes. Only HTML tags such as p, div and spans are considered for this scoring. The basic scoring mechanism used counts the amount of commas and the amount of groups of 50 characters in the inner text of an HTML node. This score is also partially bubbled up to the parent and grand-parent node so the combined score of multiple child-nodes would make the parent the highest scored node. Some score is also added according to some pre-defined tag names. The node with the highest score should then be the main content of the webpage. Mercury Parser was used to create the dataset by Bannihatii et al. [24], for this reason it was also used in the classifier from Section 6.1.2. For the purposes of the following sections, however, Mercury's output was found to be a bit too inconsistent and it mostly keeps the HTML tree of the original page. Secondly, the Mercury Parser was found to return empty documents in some cases as mentioned in the evaluation of the crawler in Section 6.1.3.

For the reasons mentioned, another library with similar purposes was used to remove boilerplate code from the web documents before extracting relevant data. To this purpose the boilerpipe library by Kohlschütter et al. [33] was used. Using a decision tree, boilerpipe categorises blocks of text in an HTML document as boilerplate or content. These blocks are firstly created by simply separating the different HTML tags. Secondly, the blocks are merged based on a text-density metric. The text-density is simply the number of words in a string divided by the number of lines it fills after word-wrapping in a fixed width column. The blocks are merged until the difference between the blocks is too large. These merged blocks can then be sent through classification. One point of interest when classifying these text-blocks is the features that are used. For example, the features of the classifier used in Section 6.1.2 are not going to be useful for this situation. TF-IDF is deeply intertwined with the exact content of a document and the boilerplate classifier should be as topic agnostic as possible. For this reason more shallow text features were used. While many features were tested in the work by Kohlschütter et al. [33], the final version settled on 2 different explicit feature types, namely link density and the text density mentioned earlier. The former is the amount of words or tokens in a string are surrounded by an "a" tag divided by the total amount of tokens. There is however another property of HTML text documents that was used. Depending on the granularity of the sectioning of the text-blocks, "content" blocks are more often followed by "content" blocks than by "boilerplate" blocks. Similarly, in most situations the content is surrounded by boilerplate rather that boilerplate surrounded by content. To exploit this property the link and text density of the previous and next block are also used to classify blocks for a total of 6 features. These features are then used in a decision tree learner. The result is a probably most simply explained as a series of boolean tests such as $link\_density < 0.6$ to lead to some conclusion of class. The blocks which were marked as content are then simply appended to obtain the result.

6.2.2 Section selection

Now that the policy has been transformed into a relatively clean state, relevant data can start being extracted from the policy. The first step to finding the relevant data is finding the correct section or paragraph of the policy text. To achieve this, the text obviously has to be split into logically different sections or paragraphs. A first thought could be to simply split the policy document on HTML heading tags (e.g. H2, H1). This, however, makes the assumption that website owners will format their policy pages using these tags and it also, more erroneously, assumes that the logical blocks created using these tags are the same logical blocks that a human would split the text into. It could very well be that the site owner used a single H2-tag in the middle of the document. The resulting chunks would be rather purposeless in finding the correct section.

For the reason described, a more complex solution can be used. One such solution was created by M. Hearst in the TextTiling algorithm [34]. TextTiling is a technique for automatically subdividing texts into multi-paragraph blocks that represent passages or subtopics. The algorithm has an implementation in NLTK [27] which was used for the application. A main observation, upon which TextTiling is based, posits that as a topic changes in expository text, new terms will be introduced and some terms will go out of use. TextTiling thus works via calculating a score for each possible boundary based on the introduced terms and segmenting the text based on these scores.

Starting from the text, token-sequences are created. In essence token-sequences are pseudo-sentences, the text is split into sentences of a pre-defined number of tokens. This allows for more accurate comparison between different token-sequences since the similarity between two normal sentences and between a short and a long sentence can not that easily be compared. To find the boundaries of sections, a sliding window of blocks is used. A block is a group of token-sequences. Figure 6.5 shows the sliding window in context, the size of the sliding window can be set to whatever is needed. The gap score is a metric that will later be used to decide whether or not there is a topic boundary. It can most easily be interpreted as a similarity score between the two blocks. The score is calculated as follows:

$$gap\_score(i) = \frac{\sum_t count(b_1, t) * count(b_2, t)}{\sum_t count(b_1, t)^2 * \sum_t count(b_2, t)^2}$$

Where t is a specific token and b_1 and b_2 are the previous and next block. $count(...)$ is simply the count of how often a certain token is present in that block.

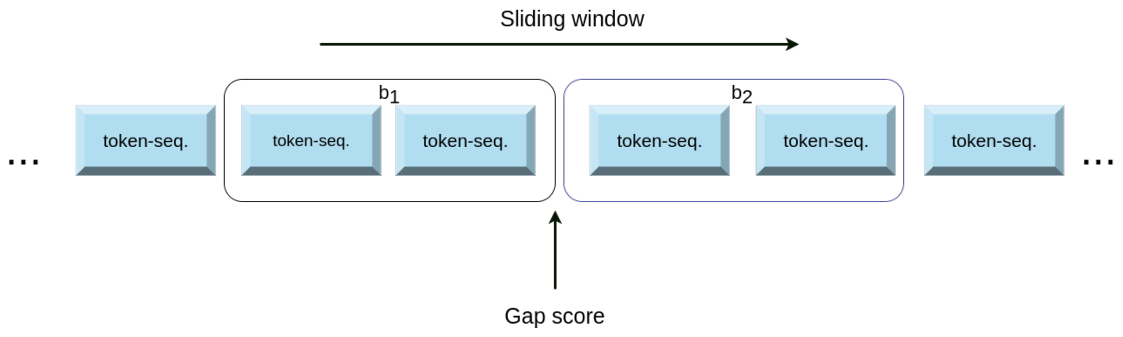


Figure 6.5: Sliding window of the TextTiling algorithm visualised

After finding the gap score for each of the blocks, the boundaries need to be decided upon. The gap scores can be plotted as seen on figure 6.6. Since the gap score represents a similarity between the different blocks, a sudden drop in similarity means that the subject has changed. This can be most clearly be seen in the graph of figure 6.6(a). The drop in similarity can be quantified by $(y_{a1} - y_{a2}) + (y_{a3} - y_{a2})$. Deeper valleys will gain a higher score than shallow valleys. This score is called the depth score.

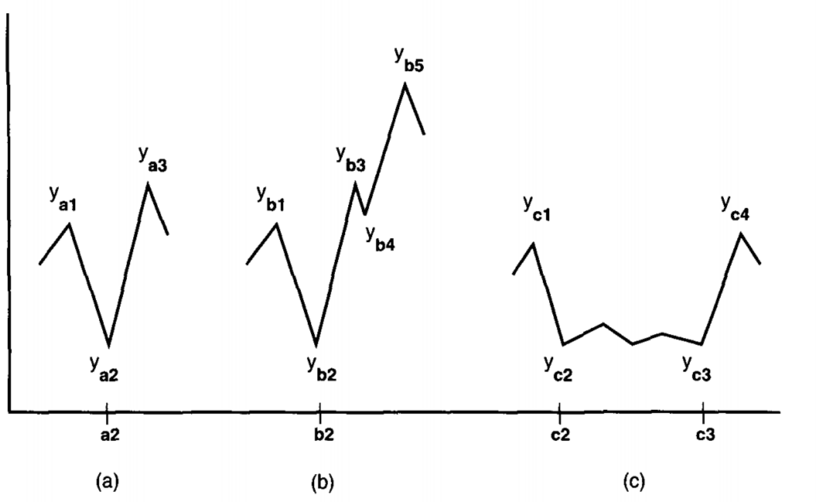


Figure 6.6: A sketch illustrating the computation of depth scores in three different situations. The x-axis indicates token sequence gap number and the y-axis indicates gap score [34]

There are some caveats, however. In figure 6.6(b), should y_{b4} also be marked as possible boundary? Small dents could incorrectly be marked as boundary candidate regardless of whether or not it should be. For this reason the gap scores are smoothed so small dents have less influence. The last case, figure 6.6(c), shows a valley of dissimilar blocks. In this case the algorithm is forced to make a slightly arbitrary choice. M. Hearst mentions that more data could possibly be used to solve this edge case [34]. When the smoothed gap scores have been transformed to the depth scores, boundaries can be chosen. The depth scores are sorted and based on a cut-off value, the top boundaries are used as section splits. The cut-off value is based on the mean and standard deviation of the depth scores.

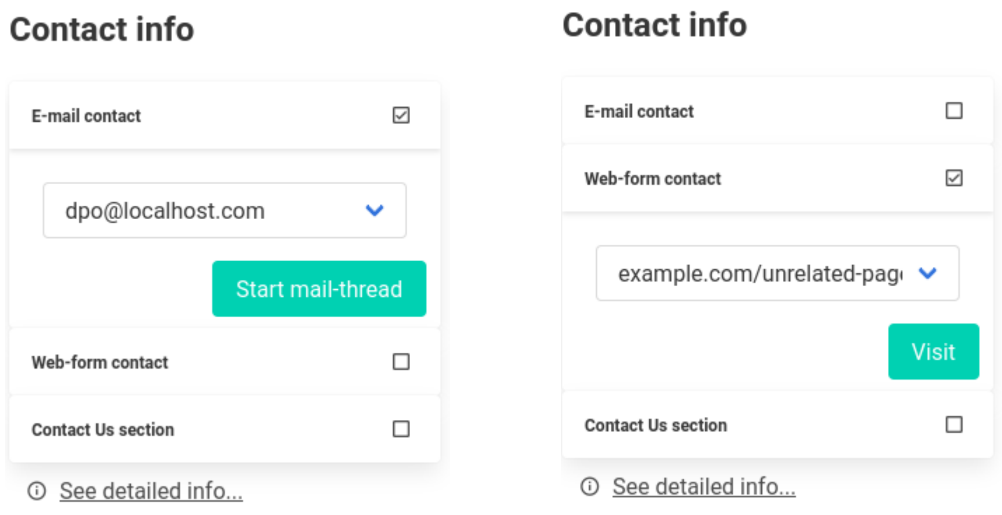
The algorithm returns a set of boundaries with which the original text can be split. The remaining issue is then, how is the most relevant section decided? One solution would be to look at machine learning again, such as was done in Section 6.1.2. However, sections of a text are, for one, much shorter than full webpages. Secondly, the content type or style of the document is already known, since it should be a privacy policy. This would mean the set of possible values or tokens are much more limited than for a wide collection of webpages. The argument can be made that a form of machine learning could be overkill when attempting to identify a section that is about the subject access request. The vocabulary concerning subject access requests or similar GDPR rights such as right to be forgotten is different enough, so a selection can be made based on the words present in the sections. One example would be that "resident of Europe" is often mentioned in these sections. In this manner, a set of search terms or regular expressions can be gathered empirically with which sections can be scored based on the presence of those terms. Listing 6.1 shows some example search terms. The 'r' prefix denotes search terms which are regular expressions. The amount of search terms that are present within each section are counted. Sections that are more relevant to the right of access, should logically receive a higher score.

```
'GDPR'  r'europ(ean|ese|e)?'  r'data protection( officer)?'
'EEA'  'resident'  'exercise'  'data access request'
'subject access request'  r'access to (your )?(personal )?data'
```

Listing 6.1: Pseudocode examples of search terms used to find "right of access" sections

Similar to many natural language systems, the TextTiling and the section selection which were discussed are not perfect. There will always exist edge cases in which a certain text will throw off the result. For this reason, similarly to the crawler and classifier from Section 6.1, the top 3 results are all used in the next step and also sent to the user for informative purposes.

To find the methods of contact, these being either an e-mail address or a URL to a form, they are simply extracted from the top 3 sections using regular expression and an HTML parser for the e-mail address and URL respectively. The results of all the above are then sent to the front-end browser extension for use by the user.

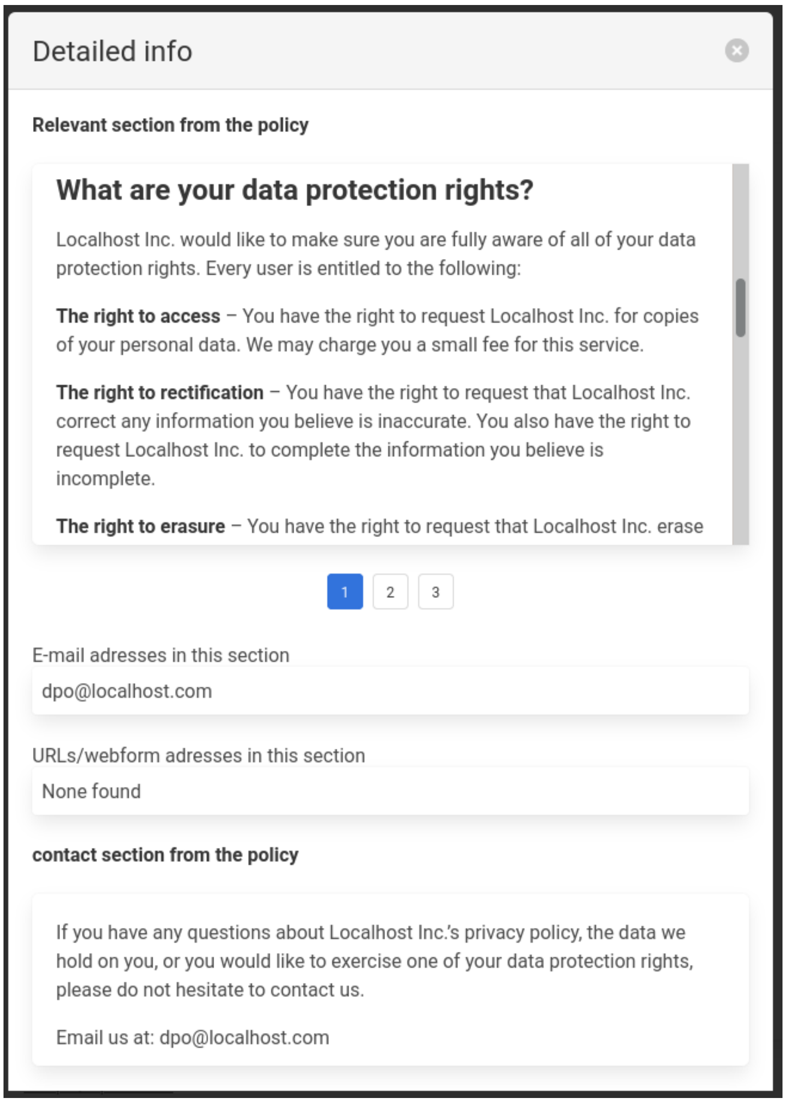


(a) (b)

Figure 6.7: Access request contact info as presented to the user.

Figure 6.7 shows the presentation of the contact info to the user. The information is split up into an accordion element based on the type of contact point. To, be it marginally, assist the user in deciding which is the most probable correct contact point, additional sorting is done to have the most relevant contact info presented first. The first layer of sorting is which drop-down opens by default. Contact info in the higher ranked section has preference over those in lower ranked sections. The second layer is sorting for the content of the actual contact point. The logic is that e-mail addresses containing a word such as “DPO” (short for data protection officer) are more often relevant than an e-mail address containing none of such words. Lastly, info extracted from the contact section in the privacy policy is also collected and sent to the user. This was implemented due to some privacy policies referencing the “contact us” section when instructing the reader on how to contact the company.

One point of discussion with the presentation so far is that it is presented as a complete black box. The user has very little idea where these e-mail addresses or URLs originate from. However, there is also the concern that showing too much contextual information at first sight would confuse novice users or people very unfamiliar with privacy policy terminology. The approach of the "See detailed info..." button on figure 6.7 was thus taken. Clicking this button opens an overlay which contains an overview of the sections which were deemed most relevant and the contact info that was extracted from them. This presentation can be seen on figure 6.8. This approach allows users who are interested in the source of the information to interpret it freely while not overwhelming novice users.



Detailed info

Relevant section from the policy

What are your data protection rights?

Localhost Inc. would like to make sure you are fully aware of all of your data protection rights. Every user is entitled to the following:

The right to access – You have the right to request Localhost Inc. for copies of your personal data. We may charge you a small fee for this service.

The right to rectification – You have the right to request that Localhost Inc. correct any information you believe is inaccurate. You also have the right to request Localhost Inc. to complete the information you believe is incomplete.

The right to erasure – You have the right to request that Localhost Inc. erase

E-mail adresses in this section

dpo@localhost.com

URLs/webform adresses in this section

None found

contact section from the policy

If you have any questions about Localhost Inc.'s privacy policy, the data we hold on you, or you would like to exercise one of your data protection rights, please do not hesitate to contact us.

Email us at: dpo@localhost.com

Figure 6.8: Detailed info overview of the extracted sections

6.3 Trust rating

As mentioned in Section 5, a trust rating could assist users in making decisions. This decision could include what data to send upon request from the data controller. ID cards in particular are sensitive data which controllers may request of a user. As this information could be used for nefarious purposes by illegitimate data controllers, such as impersonation, it is of some importance that the user does not send too much data. On the other hand, in some cases this data is actually needed to do proper authentication. An example for this situation would be that of banks. One should hope their bank is a trustworthy entity, so sending a scan of their ID would be reasonably safe.

The aim is to quantify this trust. One main issue with attempting to quantify a level of trust for companies is that this is rather subjective. Companies that only exist within a single country could from one perspective seem more trustworthy than an international company but another perspective could be that the smaller company possibly invested less into security measures. Similarly, mid-sized international companies should, logically and hopefully, invest more in security but their international profile certainly makes them more interesting targets for data leaks. A reassurance, however, is that for the very big companies the GDPR policies and securities should be better managed. We would argue that the big companies have more eyes, from researchers and the like, controlling for mismanaged policies and guidelines. In this sense, the two extremes of trustworthiness can be reasonably defined. These being the top companies on the, generally, more trustworthy side with GDPR-related data and phishing or fake websites can be put on the other, far from trustworthy side.

Filtering these extremes out is the first step of the process. The top companies can be filtered out by looking at the Alexa top sites[7]. This lists the most visited websites on the internet and this generally means these are the biggest companies. The untrustworthy side of phishing or fake sites are slightly more complicated. One approach to filter out nefarious or fake data controllers would be to look at the actual e-mail that was sent. This is, however, less interesting and possibly not really useful in the current context. Since the browser extension relies on the APIs of e-mail services such as Gmail, it can be reasonably assumed that the e-mail services would generally do a much better job at filtering out questionable e-mails. The amount of data to which these services have access is simply much larger. A possibly more productive approach would be to look at the actual URL that was found and would be used as an entry point for the crawler.

[7]https://www.alexa.com/topsites, last accessed 10 june 2021

The aim is then to find phishing websites (to filter out the untrustworthy extreme) based on a URL. There exist multiple ways of achieving this. A first way would be automated methods of detection. Dunlop et al. [35] have a certainly interesting approach to finding, as they call it, "zero-day" phishing websites. By capturing a picture of the webpage, text can be extracted using optical character recognition. The extracted text is then fed into a search engine and thus leveraging the PageRank of Google. If the results match the URL given to start with, the page is probably not phishing. This technique rests on both the accuracy of the PageRank and the assumption that phishing pages are generally short-lived and do not quickly rise up the PageRank results. Li et al. [36] have a more standard approach to this issue. After gathering features from the URL and webpage, the feature vectors are passed through a multi-layered machine learning system. Mohammad et al. [37] used similar features but applied a dataset to calculate reasonable borders for which the feature is considered "phishy". Some examples of such features are the length of the domain name, since phishing URLs tend to have long names, and using unusual symbols in the URL such as the "@" or "$\sim$". The "@" symbol in particular could be abused since browsers ignore anything before the "@" as the actual address usually follows the symbol. Other, more self-explanatory, features are the use of HTTPS over HTTP or the URL being an IP address. Some features also rely on the content of the webpage, such as the amount of links that go to a different domain or certain JavaScript functionalities that are often abused.

Another method of filtering phishing sites is using existing and updated databases of webpages. One example of such database is PhishTank[8]. The website is an offshoot of OpenDNS[9], which is part of Cisco, and offers a community-ran database of phishing URLs. Users submit and vote on candidate phishing URLs. PhishTank database is publicly available for use. The database is, for the purposes of the trust rating, a more feasible and simple solution. Since this situation of a phishing site is only an extreme, focusing on catching every single "zero-day" phishing site might be heavy-handed. This does not mean that the features introduced in [36] and [37] are of no use. Some features can be used to give some gradation between the two extremes of phishing and top companies.

Thus far, only the extremes of trust have been filtered out. There obviously exist some graduation between these two extremes. Trust related work exists in some contexts different from the current. One such area is for marketing related topics [38] but they

[8]https://www.phishtank.com, accessed 15 apr. 2021

[9]https://www.opendns.com, accessed 08 apr. 2021

often rely on visual aspects from the end-user perspective. This is probably not the type of trust that is aimed for in the current context. More computer science related works often talk about the trust between users or websites with regard to the information that is posted [39, 40]. These often require a set of multiple users or sites to calculate the expertise and trust between. Within the current context the only real information the trust can be based on is the e-mail and the extracted URL. Other characteristics will have to be used to, to some degree, decide whether a website seems trustworthy.

Some basic features from [37] on which the gradation can be partially achieved include: the missing of HTTPS, the presence of unusual symbols, the presence of many subdomains (detected via "."), the presence of suspicious words (e.g. "secure", "login") and the length of the domain. The last of which uses the length (75 characters) defined in [37] as boundary. The feature of the domain name being an IP address, is assumed to be the same as a phishing site. No business with acceptable security standards would use a plain IP address. An additional feature is that of the top-level domain. There is a reasonable list that can be compiled of top-level domains that, generally, could point towards untrustworthy practices. Many or most companies use top-level domains such as ".com" or their country code such as ".be". However, there exist top-level domains that are free, such as ".tk" and ".ml". If a website is not willing to spend money on a proper top-level domain, the question is whether or not they are willing to spend on security. While it is definitely not a deal-breaker, it is something that would pull the overall trust down.

One more features from which a certain shift in the amount of trust can be gathered would be past security (mis-)management by the company in question. While certainly up for debate, we would argue that if a company or organisation has, in the past, experienced a data breach or leak, their trustworthiness goes down. Using a news API such as the one provided by Usearch[10] allow for the searching of news articles matching a set of search terms. In this case those search terms would include the name of the company and terms such as "data", "leak" and "breach". If any articles matching these terms are returned, the trust rating gets reduced. As mentioned before, this approach could actually be disputed. Some could argue that since the company has come under scrutiny for this data leak, the company should have upgraded their security and be more vigilant. There are arguments to be made for and against both perspectives.

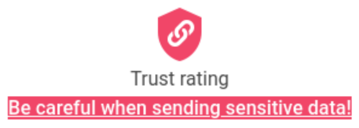
There also exist some other APIs that provide a "trust rating". Most prominent are those available with some anti-virus software. These include but are not limited to:

[10]https://usearch.com/news-api, accessed 09 apr. 2021

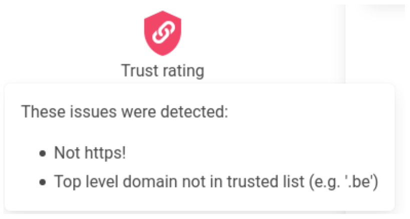
Norton Safe Search[11], McAfee WebAdvisor[12] or F-Secure Browsing Protection[13]. The effectiveness of these may be called into question, however. McAfee WebAdvisor even had a lawsuit filed against them over misleading and incorrect ratings [41]. A similar trust ratings also exist in the form of a free service and optional browser extension. Web of Trust[14] is a service basing the trust ratings on user reviews and existing blacklists. The browser extension has also received deserved criticism for selling user data [42]. Besides the issue of data selling, basing website trust on user reviews opens the platform to abuse by vindictive users. When looking at less commercial APIs for website trust rating, some options include: BuiltWith Trust API[15], Domain ReputationAPI by WhoisXMLAPI[16] and Web Risk by Google[17]. One major issue with these APIs is, again, the question of whether or not they employ the same meaning of trust as is intended in the situation of this thesis. For example, the Google Web Risk API, at time of writing, offers three threat types for which can be searched. These types are malware, unwanted software and social engineering. The latter definitely could align with our purposes (mostly with the phishing extreme) but the former two are not immediately relevant for our purposes. A second major issue is the question of the effectiveness of most of the above APIs. Some empirical testing with known phishing or untrustworthy URLs showed some APIs returning safe ratings. Due to all mentioned reasons, it was decided to, initially, not apply any of these APIs for the trust rating of the subject access request targets.

A last thing to note is that, ultimately, the trust rating is a subjective measure. One person could trust certain sites more based on some metrics than others. For this reason the trust rating is given using a colour coding system. If the URL is found to be from either a top company or no negative features are found, it is shown as a green shield. If the URL is found to be a phishing site or multiple negative features are found, a red shield is shown. Situations in between the former two are shown using a yellow shield. For full transparency, all negative features are listed in a list which is shown on hover over the shield. This allows the user to make their own, more precise, decisions based on the characteristics shared with them.

[11]https://search.norton.com, accessed 09 apr. 2021
[12]https://www.mcafee.com/en-us/safe-browser/mcafee-webadvisor.html, accessed 09 apr. 2021
[13]https://help.f-secure.com/product.html#home/safe-windows/latest/en, accessed 09 apr. 2021
[14]https://www.mywot.com/, accessed 09 apr. 2021
[15]https://api.builtwith.com/trust-api, accessed 09 apr. 2021
[16]https://domain-reputation.whoisxmlapi.com/api, accessed 09 apr. 2021
[17]https://cloud.google.com/web-risk, accessed 09 apr. 2021



(a)



These issues were detected:

- Not https!
- Top level domain not in trusted list (e.g. '.be')

(b)

Figure 6.9: The shield icon showing the trust rating and the reasons associated with it.

6.4 E-mail interaction: future work

At this point in the process, if all has gone well, the correct contact information to which to send a subject access request should have been found. What remains now is sending the initial mail (if the point of contact is an e-mail address) and replying to the following correspondence. It could be interesting to, to some degree, assist the user in creating the response mail. For the initial access request mail, a simple template mail can be used. This is, however, not the case for the correspondence that follows it. Data controllers sometimes, for example, ask for an ID scan of the user. There is some existing work on the authentication and communication between data controller and data subject. This work will first be examined to get an idea of what might occur in the correspondence.

6.4.1 Survey of existing work regarding SAR correspondence

Kröger et al. performed a longitudinal study on the type of responses that are received in three iterations [43]. With the iterations taking place before, close to and after the GDPR enforcement date. Occurring in November 2015, March 2018 and August 2019 respectively. The access requests were performed on mobile applications for both iOS and Android operating systems. The response rate to the initial GDPR mail averaged 77.7%. In the third iteration of the research a reminder mail was sent which boosted the reply rate by 6%, from 74% to 80%. The responses arrived rather quickly as 80.3% arrived within five days. While the response rate and speed of response are relatively high, the quality of responses in the study by [43] certainly paints a bleak image. From the set of apps that responded only 19%, 65% and 55% fulfilled the subject access request for the three iterations respectively. The increase from iteration 1 to iteration 2 is assumed to be attributed to both the introduction of the GDPR and the tone of the access request. In the second and third iteration of the study Kröger et al. wrote more formal mails, including references to the actual data protection laws and mentioning that the responsible authorities would be notified if necessary. This effect of the tone of the mail is also supported by the work done by Herrman et al. [44] before the introduction of GDPR.

While perhaps less relevant, a quite interesting effect that [43] observed, is that some companies explicitly noted the lack of means or time to comply to the legal duties laid out by GDPR. A few companies even quit offering a login function or discontinued the entire app. One mayor issue, at least for the context of this thesis, is the introduction of a language barrier in the correspondence. The study found ten language barriers in the form of responding in other languages, switching languages mid-correspondence, faulty English or responses translated using Google Translate. Also relevant are the results regarding identification of the user. In the group of data controllers that sent a copy of personal data, most did not ask for identity verification. Coming down to 84%, 85% and 76% for each of the three iterations respectively. One major factor for this is, in contrast to works discussed later, is the fact that the inquiries were sent from the same e-mail address as the account on the platform in question. While this is certainly spoofable, it is a minimal form of authentication. Only a minority of data controllers required additional identity-related information. In iteration one 13% requested the birthdate, address or customer number. This dropped to 2% and 0% in the following iterations respectively. This was replaced by requests for copies of ID cards and driving licences at 5% for iteration two and three while these requests were absent in the first iteration. As the author rightfully notes, these methods of authentication often request more data from the user than the data controller initially had. Particularly ID cards are a questionable form of authentication in situations where the data controller has no means or knowledge of the users real identity. An example of this would be apps where you sign up with nothing more than an e-mail address, user- or nickname and password. Once authorised if needed, the study received the requested data via e-mail at 86.7% on average. In iteration two, 9% sent the data via postal mail and in the last iteration, a jump was seen in the number of data controllers who deliver data through the app itself (from 7% in round two to 19%). When looking at the file formats, in most cases it was delivered in PDF, HTML, DOCX, CSV etc. or as plain text in the e-mail body. However, the content of these files are not always that clear. The authors found that the data was unintelligible due to formatting errors or obscure data labels in almost one in four cases (24%) in the third iteration. More worryingly, in five cases the account of the user was deleted in response to the subject access request, without explicit consent.

Urban et al. [45] performed a study on subject access requests in online advertising. In this study two iterations were performed, one month and four months after the GDPR enforcement date, June and September 2018 respectively. The angle of online advertising certainly offers some new challenges when it comes to authentication of the user. Online advertising mostly functions based on cookie IDs. This means that there is no form of authentication via, for example, the e-mail address which was the case in the mobile app study. Additionally, the information the data controller holds over a user has a high chance of being more detailed than a single mobile application since the data spans many websites and most importantly which websites were visited. Interestingly enough, seven data controllers across the two iterations requested an affidavit confirming the cookie ID is of the user in question. An affidavit is a document signed under oath as proof of its veracity. Eleven data controllers asked for a copy of an ID card. Even tough the same situation applies where the data controller has no idea of the identity of the user (or at least shouldn't). There were also situations where an online form was available where the cookie ID could be simply entered and partial (non-sensitive) data was shown. However, these forms allow for adversaries to fake cookie IDs to gain access to this data. The affidavit is a decent solution for this situation even tough it is probably inconvenient for the end-user. In total 54% and 64% finished the process in the two iteration respectively. Similarly to the previous work, most shared data was in standard file formats as prescribed by the GDPR.

The work of Boniface et al. [46] takes a specific focus on the authentication mechanisms used in communication between a data subject and data controller. This study was performed in October 2018. As mentioned before, there exist some issues in this authentication process. The data subject does not want to leak more information than necessary, but the data controller also needs to correctly authenticate the user requesting access. Boniface et al. [46] defined three main issues that can occur in this authentication process: "breach of data" where data can be accessed by unauthorised users, "privacy invasion" where the data controller abuses their power to gain more data and denial of access. Additionally, the possible threats from these issues were also defined: "impersonation", "incorrect disclosure", "abusive identity check" and ' "impossibility of authentication". The first two of which are a "data breach" issue where an adversary impersonates someone to gain access or the data controller sends data of the wrong person respectively. These dangers or threats were firstly matched against the recommendations by different Data Protection Authorities (DPA). These recommendations most often describe the rights of the data subject are. Only 17 of the 28 studied recommendations provide explanations on how to actually achieve this. Several provide a template for the access request. None such guidelines or explanations were found for the role of data controller. For authentication specifically, five DPA require a copy of government ID, one of which (Germany) suggests blurring unnecessary information. Austria's DPA gives a template containing an "identity" field where either a customer number or a government ID can be given. Three DPA recommended applying minimisation of shared data.

Boniface et al. also evaluated the authentication practices of the top 50 Alexa websites. These sites were combined if they belonged to the same (parent-)company, resulting in a final 27 sites. Seven of these websites requested a copy of a national ID card or other government issued documents. Most times this request for ID is supported by the need to check for eligibility of the request. In other words, to check if the user in question is resident of the European Union. One specific site gave no such justification and this is considered an abusive identity check by the author. Similarly to Urban et al. [45], the paper [46] also looked at subject access requests for third party trackers. The authors attempted to exercise the right of access for 30 third party tracker domains (7 of which were grouped due to belonging to the same company). Thirteen requests were denied or technically not possible. Two of these companies quoted that the right of access should not adversely affect others and that the rights do not apply if the organisation is unable to effectively identify the data subject. Eight companies requested additional data, such as an ID and full name, besides the online identifier or cookie ID. Five of these eight companies request a signed form to validate that the user is the owner. Interestingly, one specific company asked for a witness to also sign the form and send their ID. Which is obviously an abusive identity check. Luckily, four of all the domains did have a system based purely on the cookie ID and a verification step as confirmation of ownership of that ID. This, of course, might also open up the possibility for bad actors to spoof their cookie ID.

Lastly, there also exist previous work abusing the, perhaps too weak, authentication methods employed by data controllers. Di Martino et al. [47] attempted to achieve unauthorised subject access requests by impersonating a target individual. Out of the 55 companies studied, 14 only offered access requests via a dedicated webpage behind a login page. These are, in the context of the authentication methods, generally safe from abuse via impersonation. Similarly, companies that employed a login authorisation even when exercising right of access via e-mail are safe from abuse. The remaining 41 companies asked for various additional information of the user to verify their identity. The requested information might, at first sight, seem more detailed than might be needed. It should be noted, however, that this work revolves around attempting to convince a data controller that the person sending e-mails from a similar address is the person who has an account on their platform or website with the original e-mail address. Only 12 of the remaining 41 enforced the policy that subject access requests should come from the same e-mail address as with which the account was made. Five companies allowed other credentials if the subject had no access to the original e-mail account. This was mostly related to account specific information such as recently bought items, which the author considered as a decent safety measure since an adversary usually does not have this information. Other credentials companies have requested are national ID card (13 out of 55), home address (5) and as mentioned earlier, various data such as last visit (11). Interestingly enough, two companies decided to call the user on a phone number they had on file. In the end the author found 12 of the 55 organisations vulnerable to access requests by third parties. Three companies had a "link leakage" where data was leaked of, for example, users with a similar or identical name. Di Martino et al. [47] concluded that the most powerful authentication system would be automated processes using known login credentials. If this is not possible, a strict policy on same-address access requests should suffice as well.

6.4.2 Generating response mails

The initial goal was to also implement and test an approach as to assisting the user in responding to the follow-up mails. There exists work that aims to automatically generate response mails for other contexts (such as tech support mails). One major factor in these approaches is the need for an existing set of mails to base the answers on. A dataset of access request response mails is something that, to our knowledge, is not freely available. For this reason, the approaches that could be taken if this were to be implemented are rather limited and even then it would be a speculative implementation. Machine learning approaches, for example, would not be possible since there is nothing to learn from and "rule-based" approaches are based on the guesses of the creator without data to base the rules on. The following explanation is thus a look at some ways the automated responses could be generated. The actual implementation and testing is left to future work which would probably go paired with the creation of an access request mail dataset.

As mentioned, there exists some work regarding automated response mails. Sneiders [48] divided these works up into three main categories: text categorisation with machine learning, statistical text similarity and text pattern matching. The angle that is taken for most of these approaches is either a specific type of situation (e.g. internal company mails being matched to "task", "proposal to meet" or "promise") or frequently asked questions support ticketing. The latter of the two is most relevant for the assisted e-mail answering in the context of subject access requests. When looking at the existing works presented in the previous subsection, the most prominent information that a company can ask about, in terms of authentication, is probably the ID card but also the name or birthdate of the user, for example. We would argue these requests for info are, in essence, a form of frequently asked questions. There is a limited set of most common requests (e.g. authorise via ID card) and matching these requests is a set response (e.g. attaching ID card to the response). Attempting to cover any and all cases would be difficult since these requests can be rather unpredictable due to the open-ended nature. Account specific details such as in [47], for example, are difficult to pre-plan for since this can cover a very wide range of details. Completely automating everything would also be difficult since the authentication sometimes requires explicit user intervention. Examples of this would be the affidavits requested in [45] or the telephone call from [47].

The machine learning techniques, while probably possible, are not great for the situation of the browser extension. It would either require the classification to happen in the browser or the mails to be sent to a back-end server. The latter is a privacy issue since from the outset it was the aim to do as much as possible of the process in the browser due to a privacy concern. Besides that, the machine learning techniques mostly focus on mail categorisation. Text similarity approaches, such as in the work by Lapalme et al. [49] and the addition by Alfalahi et al. [50], use previously sent question-response mails to decide upon a response for the incoming mail. The context for [49] is professional investors responding to questions via e-mail. The system searches through the mail history and finds the most similar queries and their matching response. One idea applied in this is that the professionals often use a more focused vocabulary in responses to describe situations, while less knowledgeable users might not use precise language in their questions. When comparing the new question to the known questions in the question-answer set, this imprecise language could lessen similarity. For this reason, the co-occurrence of terms in the question and answers are calculated. Words that appear together more often in the question and answer can then be used to improve results. These co-occurrences were then used by [50] to produce "shadow answers". By replacing question terms by their co-occurring synonyms, the query should provide more relevant results.

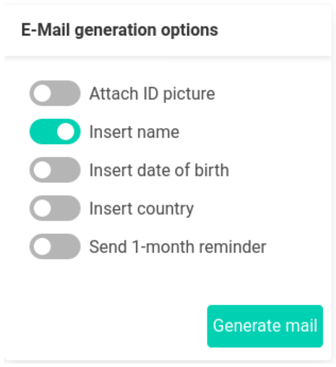
The last approach to automated e-mail responses is the text pattern matching. Sneiders [51] uses a regex-like set of patterns to decide upon an answer to send. The question mail is split into paragraphs and each paragraph is then matched against the list of patterns. If the paragraph matches the required pattern and does not match the forbidden pattern, the corresponding answer can be used. Table shows 6.2 shows some features of the text similarity and text pattern approaches as defined by [48]. We would argue that text patterns would probably be the simplest, most efficiënt choice for the case of subject access requests. Its drawbacks of domain dependency or diversity in answers are not that big of an issue since it is a focused domain and the value of diversity in answers might not be great. The main issue is, of course, language dependency. However, while the technique might not be language dependent, the collection of access requests definitely is. Text similarity will also not work for a language without a dataset to support that language. Lastly, there could be an issue with storing a full dataset of question-answer pairs in the browser extension. Which approach is objectively best, only future work could decide.

Table 6.2: Features of the e-mail answering approaches [48]

	Text similarity	Text patterns
Correctness of answer	Medium	Good
Nuances in the query text	Poor	Good
Diversity in answers	Unlimited	Small
Domain dependency	Little	Significant
Same domain, different language	Easy	Difficult

Some last notes on the topic of answering e-mails will be made regarding preprocessing of the e-mails. In the previously mentioned technique of text patterns, for example, an issue might form if the mail that is received is not written in full sentences. A pattern that could form is that of the data controller listing the requirements. If it is an HTML-formatted e-mail, this would be achieved using “li” and “ul” tags. A possible solution to alleviate this issue could be the list aggregation step used by [10]. The list aggregation combines the different items in a list with the preceding intro (or vice versa depending on the item length). In this manner, the listed items become, more or less, full sentences on which the detection can be performed. The concept can become more clear via an example, say a data controller answers with “To confirm your identity, please send us your:” followed by a listing of items. Aggregating the list would result in multiple sentences such as “To confirm your identity, please send us your: ID card”. The latter is much more logical to match answers against.

In the current implementation, the users has been given some switches for elements to include and a button to generate a mail based on some combined template parts. By selecting some elements (such as date of birth, for example) and clicking the generate button, an example mail containing this information is generated for the user.


E-Mail generation options

- Attach ID picture: off
- Insert name: on
- Insert date of birth: off
- Insert country: off
- Send 1-month reminder: off

Figure 6.10: Mail generation options

Chapter 7

Experimental results

To round out the main portion, some final evaluations were performed. In this section, the aim is to once again involve users. Previous evaluations were more based on theoretical success with a random selection of top websites. Additionally, in the process as was developed, the user was sometimes assumed to intervene in the decision-making such as selecting the most likely privacy policy URL. Both of these aspects should be tested with real users. This will be done in two parts, initially a usability test is performed to have users run through some cases and the intervention was assessed. Secondly, a list of websites that real users submit will be tested as was done in Section 6.1.3. Another option would have been to have users test the extension using their actual mailbox. The decision was made against this option since, on one hand, asking participants to open up their mailbox for a test might make them more hesitant to participate. On the other hand, there is a lot of uncertainty as to the quality of test that could be performed. It would be an unknown whether or not there would be any interesting companies to test on. The user intervention aspect would also be more difficult to make claims on, since there is no consistent situation between the different participants. In future work, once the extension is a bit more developed (e.g. assisted e-mail correspondence added), an option could be to do a "diary study" where participants track their interactions with the extension over time as they perform actual access requests.

7.1 Usability test

Due to the active pandemic, it was decided to limit participants to acquaintances and colleague students. In normal conditions the test would be performed in a known location and a laptop would be provided with necessary installations prepared. Since the test had to be performed online, the participants were asked to (temporarily) install the browser extension. The choice of acquaintances and colleague students was made, in part, due to this need for installation of (to them) unknown software by the participant. Ten participants were contacted via e-mail and allowed to select a timeslot in which they wished to participate. According to Nielsen et al. seven users should be sufficient for small projects while 15 is optimal for medium to large projects [52].

This selection of participants, obviously, introduces a bias towards people more knowledgeable about computer usage and computer science. Out of the ten participants, seven either currently study in computer science or IT. This group is split over students doing their bachelor's, master's and PhD degree. The remaining three participants do not work or study in computer related fields. However, two participants within this group are rather familiar with computer usage.

The test was split into three main parts. After some introductory questions to gain some knowledge about the participant (e.g. their area of study or work), the participants were given a series of tasks or situations to perform using the browser extension. The participants were asked to think aloud while performing the task so their thought process would be clear. The last part of the usability test consisted of the participant giving their feedback and some questions to address some specific points. The video-calls in which the usability test was taken were recorded after receiving consent by the participant.

The participants were asked whether or not they had heard about "right of access" as a concept and if so, how they would describe it. From the 10 participants, four were spot on with their knowledge of "right of access". Of the remaining participants: two had a vague but not complete idea of what this entails, two had heard the term somewhere and two had never heard the term before or at least had no recollection of where they heard the term. If they were unfamiliar with the right, it was explained or expanded upon in the vague cases. Six participants claimed to have requested their data in the past. Most of these users said this was done in the case of Facebook. Some were among the participants who only had heard of the term in the past. It could be assumed that these users simply thought this was a feature of Facebook as opposed to a right of European citizens.

None of the participants outright stated to be very involved with their privacy in daily life online. Four participants reported to always try and select minimal cookies or to minimize tracking. Two participants said it was something they held in the back of their mind while online and the remaining four said they either don't really think about it or simply didn't care about the data they send. Lastly, all but one participant claimed to have actively worked with browser extensions in the past.

The hands-on part of the usability test contained three main scenarios. In each scenario, the participant was asked to perform a subject access request for a company. Some contextual story was added to the scenario so it was not a completely repetitive experience for the participant. In all cases, if there was e-mail correspondence in the scenario, the mails were received and responded to by the moderator. This means that the e-mail correspondence is not necessarily realistic when it comes to the content. The searching of the privacy policy and the correct contact information or mail address, however, was done on real websites in two of the three scenarios. The choice to control the e-mail correspondence is mostly due to time restrictions but also gave the ability to force an interaction based on the trust rating. This low trust interaction was done with the third, completely fake, company.

The first scenario was basic, as to give the participants an introduction to the application. The URL choice (figure 6.2) resulted in two options, one with positive certainty and one with negative certainty. The contact selection gave a single option (e-mail address). This single address result might have been detrimental to later results but this will be discussed in more detail later. The e-mail correspondence in this scenario is simply the request followed by the company asking for the name and date of birth of the user. The user should then be able to answer using the e-mail generation switches. The second scenario gave the users two positive certainty and one negative certainty URLs to select from. The contact options were also more elaborate: The default drop-down (see figure 6.7) that opens is the web-form section. These URLs point to some information on an EU site. The e-mail addresses that are found are in the form of "support@example.com" and "privacy@example.com". The user should in this case select the correct e-mail address, preferably using the "detailed info" window. The third, and last, scenario shows the user that the company has a low trust rating, as they have had a data breach. In this scenario, the user is asked to send a scan of their ID card by the DPO. The intended interaction in this scenario would, obviously, be for the user to be at the least hesitant in sending their ID.

The first scenario did not immediately show any grave mismatches between the intended way and the way users interacted. Multiple participants noted that they did not know how the process, as conceptualised in their minds, translated to the application. However, since the process up until the sending of the mail has no diverging paths (in the current implementation), there is effectively only one button the user can click to go to the next step. This is visible in figures 10.2 through 10.4 from the appendix. If this project were to be an actual public product, it could be interesting to introduce some alternative paths such as manually entering the privacy policy link. The only point at which the user has complete freedom is during the e-mail sending process (figure 7.1). Since, in scenario one, there are only two options for the URL, no participants were unsure about which one to pick. Once on the e-mail request screen (figure 7.1), some participants noted uncertainty about the meaning of web-forms or the difference between e-mail contact and contact us section. This is a theme that persisted over the entire application and in general feedback. Most participants that are less familiar with computer usage or jargon noted confusion about the words and terms used in certain parts of the application.

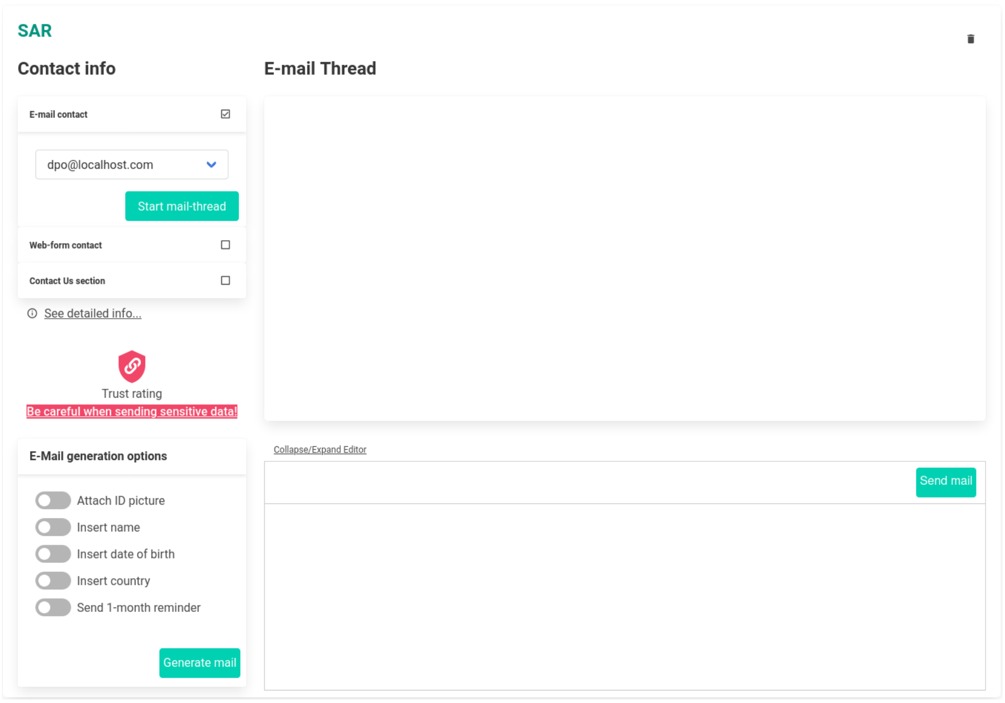


Figure 7.1: Screenshot of the full SAR overview page

There was only a single option for e-mail address in scenario one. An issue many participants ran into, was that users expected the starting template to also be updated by the e-mail generation options. This was not the way it was implemented in the current version but should be a simple enough change. When responding to the first scenario's request for name and date of birth by the DPO, all but one participant that answered used the e-mail generation options. Another user skimmed over the DPO's mail rather quickly and assumed that it told them the only information the company had was name and date of birth. The moderator could have intervened in that situation but, in general, intervention was avoided unless some error occurred or the proceeding of the test was seriously impeded by a certain interpretation or choice.

The second scenario posed some more issues. Almost all participants selected the correct privacy policy URL from the two positive certainties. Since one positive certainty privacy policy was by another company completely, participants familiar with computers definitely can make the distinction when faced with this choice. The participant that was the least familiar with computer usage seemed to get overwhelmed with the choices. This led to the participant switching between all options multiple times in sequence and to then select the negative certainty option. This ties in with the aloud thinking that some other participants shared. They mentioned that there could be more information on which to base their decision. While the certainty rating certainly assured them of their decision, a manual visit to the page could assist them in making the decision. In the current implementation this was not easily possible due to the URL that is shown simply being text and not an anchor tag. A particularly interesting suggestion a participant gave was to integrate a screenshot of the page. More often than not, privacy policies have some distinct visuals that a person who has seen them before could recognise. While some websites definitely have more unique or unusual visuals, many (if not most) privacy policies are a central block of text.

Once the correct URL was selected, the user had to choose the correct contact from the options (top left on figure 7.1). As mentioned, the user was immediately faced with the application selecting the incorrect type of contact option, namely web-forms. Besides the web-form, the "contact us" drop-down contained both an e-mail address starting with "support@" and "privacy@". The e-mail address drop-down contained only the "privacy@" option. One participant visited the URL from the web-form drop-down and interpreted a physical address on the webpage as the place to which to send a letter for their access request. In hindsight this conclusion does not seem very far-fetched to people that are not very familiar with the whole access request process. This participant also happens to be the one least familiar with computers etc. It is understandable that the amount of information from an incorrect detected web-form is overwhelming. The other participants all selected the correct e-mail address ("privacy@") but did so in different ways or with different motivations. Only four of the remaining nine participants really noticed the "support@" option in the "contact us" drop-down. The choice for all four of these participants to go with was split evenly between an intuition and simply choosing the e-mail contact drop-down. Similarly, another four participants chose the e-mail drop-down simply because this was the case in the first scenario. This is probably a fault in the set-up of the usability test. Introducing the users via the case of only having the e-mail drop-down as an option might have caused participants to simply choose this by default in the following scenarios. Only one participant used the detailed info window (figure 6.8) to make the decision in selecting the "privacy@" e-mail address. Participants who used the e-mail address (either by intuition or by default) but did not use the detailed info window were asked to visit the detailed info window during the feedback part. This will be discussed later.

The third and last scenario's main focus is the participant's interaction in context of the trust rating. Two participants sent the scan upon request of the DPO but both shared in their thinking that they did it simply for the purpose of the exercise or because there was no alternative they knew. Both said afterwards that in a real scenario they would not send this information. All other participants did not send the ID scan. Multiple of them, however, mentioned that they did not know what to do instead. Only a single participant suggested asking the DPO for alternative ways of authorisation. It is not very surprising that participants would be unsure about what to do in this situation. An extra confirmation when sending your ID if the trust rating is low would be possible as it currently stands. We would argue that adding a switch in the sense of "request alternative authorisation" would only add to the confusion of users in the standard scenarios (good trust rating). It is also not certain if this switch would be clear to users and would be used in the correct situations. Pre-emptive detection could fill this specific issue. If it is detected that the response is asking for sensitive data, a warning window could appear with the correct response to this situation. This detection would be part of the same detection as done by the generated response mails (Section 6.4.2) if they were to be implemented.

At the end of the usability test, the user was given the chance to give feedback with some questions by the moderator to guide them. Some of these have already been mentioned in the previous paragraphs. A few participants noted that the question of having to identify a privacy policy based on the URL is relatively strange. The certainty thumbs assist in this decision but without a broader context it could be difficult to know it for certain. This is where the earlier mentioned suggestion of a screenshot was also brought up. Another unclear aspect for some participants was the reasons behind the trust rating. Some of the listed reasons have little to no meaning to a layman. To give an example: the Alexa top 500 means very little to people who have never heard of it before, so it could be both a positive or negative thing without further knowledge. A balance needs to be made between clarity and verbosity. An explanation that is too verbose could also be detrimental to the understanding by the user.

The detailed info window was often skipped over by the participants. During the feedback, most participants were asked to revisit this window to explain their interpretation. Only three of the participants understood what the multiple tabs meant (1, 2, 3 on figure 6.8). Some more context as to the function of buttons is definitely required. However, a solution that would solve this and more is an onboarding walkthrough when the user first opens the application. This would solve multiple of the problems that were mentioned by the participants. Some examples would include: the meaning of the multiple tabs, the meaning of the different drop-downs or simply an overview of the process as a whole. This onboarding could also explain more technical information, such as the context of what right of access is exactly. This might also assist in the jargon issue. While buttons etc. are relatively limited in their size, limiting the amount of information in them, a short explanation in the onboarding would probably make the jargon clear for many users.

Besides the unclear things mentioned before. The response was mostly positive from the participants. The participants that are more familiar with right of access and computers in general responded more positively. Participants less familiar had more difficulty due to the aforementioned issues and uncertainty of the flow as a whole when starting out. Due to the biased participant set, it is difficult to make large-scale conclusions. With some caution, it is not far-fetched to say that, as currently implemented, layman users would most probably have difficulty using the application without some extra context and explanation. However, We would argue that laymen might not necessarily be the optimal target audience. Users that already have an interest in keeping tabs on their privacy might gain more from an automated tool such as this.

7.2 User submitted URLs

Users were asked to submit 3 URLs. It was explicitly requested that users do not submit "popular" sites such as Google and Facebook. Since the Alexa top list was used for the previous random sample test, running the test on the same sites would not be very insightful.

Table 7.1: Evaluation results using user submitted URLs

Type	Amount	% of all
Success	22	73.3
Fail	8	26.6
• CAPTCHA	1	3.3
• Very short policy	2	6.6
• No policy	1	3.3
• Application fail	4	13.3
– Mercury Parser	2	6.6
– Classifier	2	6.6

A total of 30 URLs were collected for this test. The results are relatively comparable to those of the previous test. The results are visible in table 7.1. Of al tested URLs, 22 were found successfully. Some of the remaining 8 encountered similar issues to those previous tests. In one case, another CAPTCHA was triggered by the crawler. After manually intervening on this CAPTCHA, the privacy policy was correctly classified. Four of the eight fail cases were due to the application making a mistake. Two of these cases were due to Mercury Parser failing to extract meaningful or only little of the privacy policy content. The other two cases were due to the classifier classifying the homepage as a privacy policy. The remaining 3 cases, possibly, show an interesting effect of the websites being less popular in general. For these 3 cases, the privacy policy was either very short or missing completely. However, one of these cases was a website which generally does not collect a lot of information. The data collection (according to the privacy policy) is limited to things such as the IP address of the visitor for logging purposes. It is very understandable that the privacy policy would be short in that case. The other two websites, however, were both online stores. The webstore that at least had a short text about the privacy policy only mention the information that is and is not stored when processing the purchases. The last case, the webstore with no privacy policy, does bizarrely make mention of the privacy policy in the terms and conditions. However, no link is provided in these terms. Upon closer inspection, it was noticed that the site was made using the Shopify platform. It turns out that Shopify offers a free terms and conditions creation tool, which fills in the relevant data into an existing template. This template, of course, makes mention of the privacy policy. Comparing the text of the offered template and the terms and conditions of the webstore found an exact match. The most puzzling thing is that besides the terms and conditions generator, Shopify also offers a privacy policy generator. The reason behind not simply adding this too, is unknown. That the crawler and classifier failed to correctly locate the privacy policy is not extremely unexpected if there is very little or no privacy policy to be discovered. This might, however, be an issue that could be encountered for some small or unknown sites. If a website has a small user base, they might not have an extensive privacy policy if they bother to add one at all.

Chapter 8

Discussion

This section will collect some shorter topics that either do not really fit in the main discussion or some more general ideas that could be interesting to add in the future.

8.1 Third-party trackers and FLoC

A thesis on the topic of privacy would not be complete without, at the least, mentioning third-party tracking. Contrary to what one might think, GDPR had little, if any, effect on the amount of third-party tracking parties. Sørensen et al. [53] found a small decrease in trackers but it was not certain if this was caused by the introduction of the legislation. Requesting any and all data a third-party tracker has on the user is definitely possible. However, as [45] showed, the process is more difficult than with a website for example. The issue of trackers is that together with this more difficult process, trackers can gather a lot of data about a user. Definitely combined with the system of cookie syncing, which allows a tracker to track a user over multiple sites. This is also not limited to websites either, [43] found that 90.4% of mobile apps that were checked contained at least one tracker.

It could be interesting to incorporate third-party access requests into the flow of the extension. When a user requests their data for a certain company or site, the option could be given to automatically contact the third-party trackers present on the site simultaneously. The natural way to achieve this would be to extract this info from the cookie policy, which can be a separate page or sometimes added as a section to the privacy policy. This might not be needed, however, WhoTracks.me [54] has collected the trackers of 1.5 billion page loads performed by real users. This database could thus be used to query the current site to which the current request is being sent and have a list of trackers as a result. A, possibly, bigger issue is the verification issue. For generic website cases the verification is mostly limited to information the user entered manually at some point (e.g. e-mail address, username, date of birth) and thus can be easily reproduced or handled automatically to some degree. Since third-party trackers usually do not have this kind of information, more uncommon techniques such as the affidavit from [45] are used. Automating this would obviously be more difficult.

All this works under the assumption that third-party cookie trackers are the main way of tracking, of course. Google Chrome (and other browsers before it) has decided to faze out the support for third-party cookies [55]. In its place Google has introduced an alternative of tracking. Federated Learning of Cohorts (FLoC) is, at time of writing, being trialled in Chrome browsers. FLoC, as Google claims, should be a privacy preserving mechanism while still allowing for targeted advertisement [56]. Described briefly, FLoC uses the browser history to (in the browser) generate an identifier which should group similar browser histories. Unsurprisingly, not everyone agrees with the description of FLoC as “privacy preserving”. Brave and Vivaldi browsers made explicit statements against this technology [57, 58]. Concerns were raised (among other things) about the fact that the complete browser history is being used, instead of only the websites to which the trackers usually has access. FLoC is still a developing standard and its uncertain if this technology will actually take off. That means it’s difficult to say what this would mean for the right of access or privacy rights in general.

8.2 Changing e-mail domain

During the work on this thesis, the question was raised as to the consistency of the e-mail correspondence. Some companies could, for example, receive the request on one e-mail address domain and answer these from another. If the browser extension were to attempt to track the correspondence in this manner, the response would go undetected. For the purposes of acquiring a sense of how common this is, a small scale subject access request test was carried out. Ten different companies were contacted with request for access to all data of the data subject. The companies were randomly sampled from marketing mails in a user’s inbox. Broader results beyond the domain issue will be discussed. If a company did not respond by the 23rd calendar day after the initial request, a reminder mail was sent.

The ten companies are split into following categories: four software/API, two online stores, two video game publishers, one music publisher and one mobile application. While this sample size is rather limited, the results were far from uneventful. Three of the four software/API companies answered the access request successfully. Two of these did not require a reminder for the one-month deadline. The company that did need a reminder answered with a very detailed answer within 12 hours of the reminder, going as far as to include a slideset on the topic of their marketing scoring system. All of these answered from the same domain and to the same e-mail thread as the initial request. The correspondence with the last software/API company was more complicated and will be discussed later.

The first of the two online stores responded with an automated response mail the same day as the request. The actual response was sent the next day from the same e-mail address as to which the initial request was sent. The other online store, however, did not respond at all, even after sending a reminder. At the time of writing, no response has been received and the more lenient deadline of 30 workdays since the request has elapsed. Regular marketing mails were still received by this company. Both video game related companies answered the request within the deadline and from the same e-mail address.

This leaves three companies that have not been discussed. This includes the mobile application, the music company and the last software/API company. All three of these companies deleted or suggested deleting the data, instead of sending a copy of it. The mobile application deleted and anonymised the data of the data subject without any confirmation. After sending a clarification of the initial request (specifying that a copy was requested and not a deletion), the company responded with an apology. The music company required the reminder mail before a first answer was sent. This answer included a single line: “Your information has been deleted”. No proper closing to the mail nor a full stop to the sentence was included. At the time of writing, no response has been received to the clarification as to the meaning of the initial request. The claim that the data has been deleted is doubtful as well, since four separate marketing mails have been received since the alleged deletion. Both the mobile application and the music company communicated from the same e-mail address. The deletion of a data subjects information when they ask for access is not a unique experience. Kröger et al. [43] similarly had five data controllers delete their data.

The remaining software/API company is the only one that did not respond from the same e-mail address. The target for this subject access request was one specific software product. The e-mail address that was linked in the privacy policy of this product went to the domain of the company that acquired the product some years ago. The first response that was received contained some vague information about different software product owned by the same parent company. This response did originate from the same e-mail address as to which the initial mail was sent. All further correspondence was received from a different e-mail address domain. A response was made stating that this first response did not complete the requested subject access request as, in part, not all data for which they were the controller was included. No response was received until a reminder for the deadline was sent. The same day, a form was received which needed to be filled in if the data subject wished to delete their account data. Once again, a mail was sent explaining the actual intent of the initial request, this time also including the intended target for the access request. To this, the support representative responded that the intended target is outside their scope and would have to be transferred. The support representative also mentioned that, according to their internal system, the intended target did look at or at the least received the initial request.

As this series of events was certainly intriguing, a mail containing some questions were sent. Firstly, the question was asked as to why a form for deletion was sent out. This with the intention of figuring out whether the initial template was unclear. Allegedly, this was a miscommunication between different representatives. The current representative thought the data subject wanted to “continue with the GDPR process and exercise the right to delete your information from our system”. Secondly, it was asked as to why the initial request ended up with this particular product instead of the intended target or another. The answer to this was that the first representative found no mention of an account other than the product that ended up answering. This is, however, demonstrably false since the account on the intended target product still exists and can be logged in to.

In conclusion, only one data controller of this small scale test sent a response from a completely different e-mail address. Additionally, the author of [47] was able to share that they did not encounter any switching domain names. This edge case of a switching domain name could still pose an issue if more companies have setups similar to that of the last software/API company. A possible solution to this could be to include some sort of ID at the end of the initial mail. If the data controller were to answer from another domain, the answer could be linked back to the initial request. This, however, works only under the assumption that the data controller includes the initial mail in the answer. When looking at the small scale test, one company did not include the actual request. Another possible issue with this approach is the fact that, to find these IDs, every single mail needs to be scanned. Practically, this isn't an issue since any mail interaction is currently done on the client-side, but it could be a privacy concern for users. A combination of e-mail based, thread based and ID based matching should catch most cases.

8.3 A standardised alternative

Some of the processes used in the extension as it stands now, are obviously a best effort approach. The most natural example would be the machine learning used in Section 6.1.2. Making the process guaranteed to be exact would also remove the need for user intervention, since this would account for any error in the process. The most straightforward way to achieve this would be for the company in question to tell, in some way, where the privacy policy is and what the contact information for different things are. An example of a similar idea was implemented with the P3P [12] standard. As has been mentioned in the related work, P3P had issues taking off due to there being little to no incentive for companies to implement it. Multiple websites that did implement the standard simply did so to evade the cookie bans by Internet Explorer [59]. This was achieved by simply misrepresenting the policy or using made-up tokens such as "AMZN" or "HONK" [59].

So, unsurprisingly, companies are not very interested in spending time and manpower on implementing a standard that (if implemented in the intended way) has little benefit to them. If the idea is to replace the best-effort parts of the extension by some standardised, the effort-benefit balance has to be worth for a company. We would argue this is best accomplishable when keeping the standardised component as simple as possible. A possibility for this would be an HTTP header which contains the URL to the privacy policy among other information. Listing 8.1 shows a concept of what this could look like. The directives "policy" and "roa" should be self-explanatory, represent the URL to the privacy policy of the website and the mail address to which right of access mails can be sent. The value matching the "right of" directives could practically be either e-mail addresses or URLs for web-forms. The "rop/e" directive could be used in cases where the same URL or e-mail address is used for two or more rights (in this example: portability and erasure). The last directive "ro*" would be used in the case where all rights use the same URL or e-mail address.

```
GDPRRight: policy=<url>; roa=<mail>; rop/e=<url>; ro*=<mail>
```

Listing 8.1: Rough example of HTTP header

Implementing this header should be rather low-effort for companies to add. The benefit would mostly be time saved on people contacting a generic support address instead of the correct department, URL or DPO. It should be noted, however, that this benefit only starts to show if the HTTP header is somehow connected back to the user in some manner. An obvious way would of course be the extension from this thesis but generic browser support to more quickly access the privacy policy, for example, could be a step in the right direction. Since this would be a positive incentive instead of a negative incentive such as Internet Explorer implemented with P3P, the idea (or a similar idea) might be received a bit better.

Chapter 9

Conclusion

In this thesis, a proof-of-concept browser extension was developed to (partially) automate the process of exercising the right of access. The intention was to give users an easier, more usable way to send subject access requests. While large scale conclusions are difficult to make due to the biased participant sample, some steps in the right direction have probably been made. Most steps have been made in the way of skipping most of the searching for the privacy policy and the contact info within. This search is not perfect and can still be improved by, for example, switching out the Mercury Parser by a more stable alternative. Some edge cases, such as the CAPTCHA's, will always exist. However, (semi-)automated in most cases is still far better than having to search every privacy policy manually. The actual correspondence assistance as is currently implemented is fairly basic. We would argue, however, that the initial template, the trust rating and the basic generation options do take some of the barriers away that might keep a user from exercising this right. This barrier might most often be laziness but it is a barrier nonetheless. This would definitely be strengthened if the automated mail generation and "low trust" suggestions were to be implemented.

From the usability test can be concluded (with some caution) that as it currently stands, the extension as is currently implemented would not fare well with broader audiences. As mentioned in the relevant section, the (initial) audience for such a tool might not be the computer layman. Attempting to create a tool that everyone can use is a lofty ideal but certainly a tall order. Besides that, there is the question of whether or not a layman is really that interested in knowing what data a company has on them. Even if they were to get as far as to receive their data, this data might often be meaningless to them or the structure might simply be incomprehensible. If we assume that it happens to be a comprehensible data structure and the layman notices that the company is collecting data they shouldn't, does the layman even know what to do in response to that? Does the layman know who to contact in case of abuse by a company?

All this to say that, in essence, while the tool assists with the process of exercising ones right to access, there is many more non-user friendly aspects outside but connected to the privacy rights as a whole, at least for the layman. An interesting thought could be that if this tool gives the non-layman more inclination to exercise their right more often, the companies could introduce more streamlined manners in which to do so. In a sense, increased demand might push companies to simplify the process. It is, of course, only conjecture as to what the effects would be if the extension were to be an actual product.

As a solution to the issue of subject access requests, this proof of concept might be a decent approach. That is, for if the situation would stay the same as it currently exists. The most optimal solution would, of course, to be for companies to meet users half-way in making this process more user-friendly. Whether or not this in the actual interest of (data-heavy) companies is definitely up for debate. Additionally, if some more standardised way for subject access requests were to be created, the smaller companies might not have the time or money to implement it.

self-reflection

The subject of this thesis has been a bit unusual. I would think that most master theses are usually one particular technique or protocol, for example. This thesis due to its nature required to switch between objective for each step in the subject access process. Most steps were related to natural language processing to some degree but, for example, the crawling section had a machine learning related part while the data extraction had a text segmenting focus. On one hand, I personally enjoyed the multifaceted nature but, on the other hand, this sadly led to the fact that some subjects are not as deeply explored as they could be. One particular example is that only a single machine learning algorithm was explored. However, since it's a proof of concept, a good enough solution could suffice and the absolute best solution is maybe not always needed.

When working on this thesis, the main flow was decided by the steps as defined in Figure 5.1. In the research and development this was mostly the order that was followed too. This is partially what led to the inability to implement the automated correspondence generation. If, perhaps, I looked at the other steps in the process earlier it might have been possible to collect a little bit of data to base the correspondence generation on. It almost certainly wouldn't have been a good enough basis to make really good detection but it would probably have been better than the current nothing. However, it is unknown how difficult (or easy) the data collection is. Since it is entirely unpredictable what a company might answer, the whole dataset might have been correspondence without any request for additional info. For example, in the ten request mails that were sent out no company asked additional data for the SAR flow.

A last thing of note would be that, in hindsight, the level of detail with the user survey and usability test could have been better. For the usability test in particular, the preparation to ask follow-up or multi-layer questions wasn't there. Some details have probably been lost due to the open-ended nature of multiple questions.

All in all, I am relatively happy with the result but as always there is more that could have been done, in hindsight. The result might not have been the level of automation that one could have hoped for but it was interesting as a proof of concept in and of itself.

Bibliography

[1] Council of European Comission, “Regulation (eu) 2016/679,” 2018-05-25. `https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN`.

[2] European Parliament and Council , “Directive 95/46/ec,” 1995. `https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML`.

[3] W. S. Blackmer, “Gdpr: Getting ready for the new eu general data protection regulation,” May 2016.

[4] J. A. Obar and A. Oeldorf-Hirsch, “The biggest lie on the internet: ignoring the privacy policies and terms of service policies of social networking services,” *Information, Communication & Society*, vol. 23, p. 128–147, Jan 2020.

[5] A. M. McDonald and L. F. Cranor, “The cost of reading privacy policies,” *Isjlp*, vol. 4, p. 543, 2008.

[6] R. Proctor, A. Ali, and K.-P. Vu, “Examining usability of web privacy policies,” *Int. J. Hum. Comput. Interaction*, vol. 24, p. 307–328, Mar 2008.

[7] W. B. Tesfay, P. Hofmann, T. Nakamura, S. Kiyomoto, and J. Serna, “PrivacyGuide: Towards an Implementation of the EU GDPR on Internet Privacy Policy Evaluation,” in *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, IWSPA ’18, (New York, NY, USA), pp. 15–21, Association for Computing Machinery, Mar. 2018.

[8] R. N. Zaeem, R. L. German, and K. S. Barber, “PrivacyCheck: Automatic Summarization of Privacy Policies Using Data Mining,” *ACM Trans. Internet Technol.*, vol. 18, pp. 53:1–53:18, Aug. 2018.

[9] S. Zimmeck and S. M. Bellovin, "Privee: An architecture for automatically analyzing web privacy policies," in *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, (USA), p. 1–16, USENIX Association, 2014.

[10] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, K. G. Shin, and K. Aberer, "Polisis: Automated analysis and presentation of privacy policies using deep learning," in *27th USENIX Security Symposium (USENIX Security 18)*, (Baltimore, MD), pp. 531–548, USENIX Association, Aug. 2018.

[11] Disconnect, "Disconnect privacy icons."

[12] L. Cranor, B. Dobbs, S. Egelman, G. Hogben, J. Humphrey, M. Langheinrich, M. Marchiori, M. Presler-Marchall, J. Reagle, M. Schunter, and et al., "The platform for privacy preferences 1.1 (p3p1.1) specification," Nov 2019.

[13] P. G. Leon, L. F. Cranor, A. M. McDonald, and R. McGuire, "Token attempt: The misrepresentation of website privacy policies through the misuse of p3p compact policy tokens," in *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*, WPES '10, (New York, NY, USA), p. 93–104, Association for Computing Machinery, 2010.

[14] I. S. Rubinstein, "Privacy and regulatory innovation: Moving beyond voluntary codes," *ISJLP*, vol. 6, p. 355, 2010.

[15] L. F. Cranor, P. Guduru, and M. Arjula, "privacybird: User interfaces for privacy agents," *ACM Trans. Comput.-Hum. Interact.*, vol. 13, pp. 135–178, June 2006.

[16] R. W. Reeder, P. G. Kelley, A. M. McDonald, and L. F. Cranor, "A user study of the expandable grid applied to p3p privacy policy visualization," in *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*, WPES '08, (New York, NY, USA), p. 45–54, Association for Computing Machinery, 2008.

[17] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, K. Bacon, K. How, and H. Strong, "Expandable grids for visualizing and authoring computer security policies," in *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*, p. 1473, ACM Press, 2008.

[18] P. G. Kelley, "Designing a privacy label: assisting consumer understanding of online privacy practices," in *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, p. 3347–3352, Association for Computing Machinery, Apr 2009.

[19] P. G. Kelley, L. Cesca, J. Bresee, and L. F. Cranor, "Standardizing privacy notices: an online study of the nutrition label approach," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, p. 1573–1582, Association for Computing Machinery, Apr 2010.

[20] I. Salian, "Supervize me: What's the difference between supervised, unsupervised, semi-supervised and reinforcement learning?," Aug 2018.

[21] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009.

[22] S. Raschka, *Python machine learning: unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics.* Community experience distilled, Packt Publishing open source, 2016.

[23] S. Zimmeck, P. Story, D. Smullen, A. Ravichander, Z. Wang, J. Reidenberg, N. Russell, and N. Sadeh, "MAPS: Scaling Privacy Compliance Analysis to a Million Apps," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, pp. 66–86, July 2019.

[24] V. Bannihatti Kumar, R. Iyengar, N. Nisal, Y. Feng, H. Habib, P. Story, S. Cherivirala, M. Hagan, L. Cranor, S. Wilson, and et al., *Finding a Choice in a Haystack: Automatic Extraction of Opt-Out Statements from Privacy Policy Text*, p. 1943–1954. Association for Computing Machinery, Apr 2020.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "sklearn.feature_extraction.text.tfidftransformer."

[26] Postlight, "mercury-parser." `https://github.com/postlight/mercury-parser`, 2019.

[27] S. Bird, E. Klein, and E. Loper, "Natural language processing with python, analyzing text with the natural language toolkit," *Language Resources and Evaluation*, vol. 44, no. 4, p. 421–424, 2010.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine

learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *Journal of machine learning research*, vol. 13, no. 2, 2012.

[30] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002.

[31] Y. Sasaki, "The truth of the f-measure.," 2007.

[32] D. Gibson, K. Punera, and A. Tomkins, "The volume and evolution of web page templates," in *Special Interest Tracks and Posters of the 14th International Conference on World Wide Web*, WWW '05, (New York, NY, USA), p. 830–839, Association for Computing Machinery, 2005.

[33] C. Kohlschütter, P. Fankhauser, and W. Nejdl, "Boilerplate detection using shallow text features," in *Proceedings of the third ACM international conference on Web search and data mining - WSDM '10*, p. 441, ACM Press, 2010.

[34] M. A. Hearst, "Text tiling: Segmenting text into multi-paragraph subtopic passages," *Computational Linguistics*, vol. 23, no. 1, p. 33–64, 1997.

[35] M. Dunlop, S. Groat, and D. Shelly, "Goldphish: Using images for content-based phishing analysis," in *2010 Fifth International Conference on Internet Monitoring and Protection*, p. 123–128, May 2010.

[36] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, "A stacking model using url and html features for phishing webpage detection," *Future Generation Computer Systems*, vol. 94, p. 27–39, May 2019.

[37] R. M. Mohammad, F. Thabtah, and L. McCluskey, "Intelligent rule-based phishing websites classification," *IET Information Security*, vol. 8, no. 3, p. 153–160, 2014.

[38] C. L. Corritore, B. Kracher, and S. Wiedenbeck, "On-line trust: concepts, evolving themes, a model," *International Journal of Human-Computer Studies*, vol. 58, p. 737–758, Jun 2003.

[39] Y. A. Kim and R. Phalak, "A trust prediction framework in rating-based experience sharing social networks without a web of trust," *Information Sciences*, vol. 191, p. 128–145, May 2012.

[40] Q. Liu, B. Xiang, N. J. Yuan, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, "An influence propagation view of pagerank," *ACM Trans. Knowl. Discov. Data*, vol. 11, Mar. 2017.

[41] L. Constantin, "Mcafee sued over siteadvisor warning," Aug 2008.

[42] Spiegel, "Beliebte browser-erweiterung spioniert offenbar nutzer aus," *Spiegel*, Nov 2016.

[43] J. L. Kröger, J. Lindemann, and D. Herrmann, "How do app vendors respond to subject access requests? A longitudinal privacy study on iOS and Android Apps," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, ARES '20, (New York, NY, USA), pp. 1–10, Association for Computing Machinery, Aug. 2020.

[44] D. Herrmann and J. Lindemann, "Obtaining personal data and asking for erasure: Do app vendors and website owners honour your privacy rights?," *arXiv:1602.01804 [cs]*, Apr 2016. arXiv: 1602.01804.

[45] T. Urban, D. Tatang, M. Degeling, T. Holz, and N. Pohlmann, "A Study on Subject Data Access in Online Advertising After the GDPR," in *Data Privacy Management, Cryptocurrencies and Blockchain Technology* (C. Pérez-Solà, G. Navarro-Arribas, A. Biryukov, and J. Garcia-Alfaro, eds.), Lecture Notes in Computer Science, (Cham), pp. 61–79, Springer International Publishing, 2019.

[46] C. Boniface, I. Fouad, N. Bielova, C. Lauradoux, and C. Santos, "Security Analysis of Subject Access Request Procedures," in *Privacy Technologies and Policy* (M. Naldi, G. F. Italiano, K. Rannenberg, M. Medina, and A. Bourka, eds.), Lecture Notes in Computer Science, (Cham), pp. 182–209, Springer International Publishing, 2019.

[47] M. Di Martino, P. Robyns, W. Weyts, P. Quax, W. Lamotte, and K. Andries, "Personal information leakage by abusing the gdpr "right of access"," in *Proceedings of the Fifteenth USENIX Conference on Usable Privacy and Security*, SOUPS'19, (USA), p. 371–386, USENIX Association, 2019.

[48] E. Sneiders, *Review of the Main Approaches to Automated Email Answering*, vol. 444 of *Advances in Intelligent Systems and Computing*, p. 135–144. Springer International Publishing, 2016.

[49] G. Lapalme and L. Kosseim, "Mercure: Towards an automatic e-mail follow-up system.," *IEEE Computational Intelligence Bulletin*, vol. 2, p. 14–18, 2003.

[50] A. Alfalahi, G. Eriksson, and E. Sneiders, "Shadow answers as an intermediary in email answer retrieval," in *Experimental IR Meets Multilinguality, Multimodality, and Interaction* (J. Mothe, J. Savoy, J. Kamps, K. Pinel-Sauvagnat, G. Jones, E. San Juan, L. Capellato, and N. Ferro, eds.), p. 209–214, Springer International Publishing, 2015.

[51] E. Sneiders, "Automated email answering by text pattern matching," in *Advances in Natural Language Processing* (H. Loftsson, E. Rögnvaldsson, and S. Helgadóttir, eds.), Lecture Notes in Computer Science, p. 381–392, Springer, 2010.

[52] J. Nielsen and T. K. Landauer, "A mathematical model of the finding of usability problems," in *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pp. 206–213, 1993.

[53] J. Sørensen and S. Kosta, "Before and After GDPR: The Changes in Third Party Presence at Public and Private European Websites," in *The World Wide Web Conference*, WWW '19, (New York, NY, USA), pp. 1590–1600, Association for Computing Machinery, May 2019.

[54] A. Karaj, S. Macbeth, R. Berson, and J. M. Pujol, "Whotracks.me: Monitoring the online tracking landscape at scale," *CoRR*, vol. abs/1804.08959, 2018.

[55] D. Temkin, "Charting a course towards a more privacy-first web," Mar 2021.

[56] S. Dutton, "What is federated learning of cohorts (floc)?," Mar 2021.

[57] P. Snyder and B. Eich, "Why brave disables floc," Apr 2021.

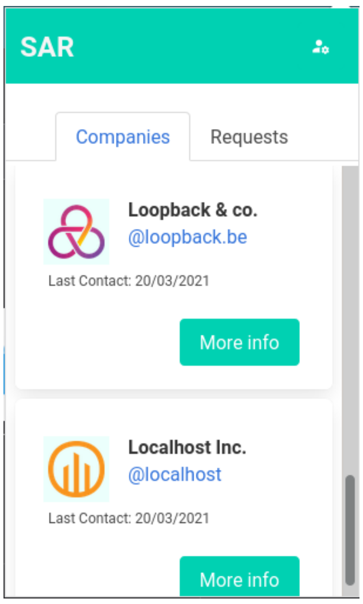
[58] J. von Tetzchner, "No, google! vivaldi users will not get floc'ed.," Apr 2021.

[59] L. F. Cranor, "Necessary but not sufficient: Standardized mechanisms for privacy notice and choice," *J. on Telecomm. & High Tech. L.*, vol. 10, p. 273, 2012.

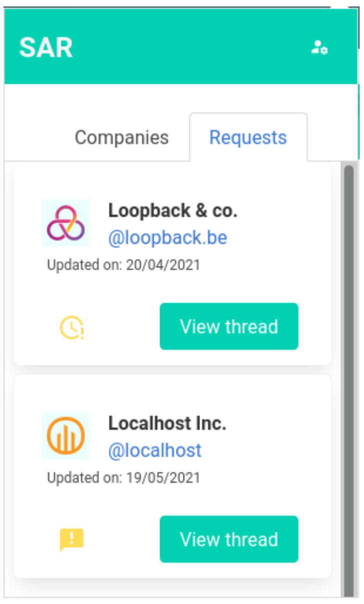
Chapter 10

APPENDIX

10.1 screenshots

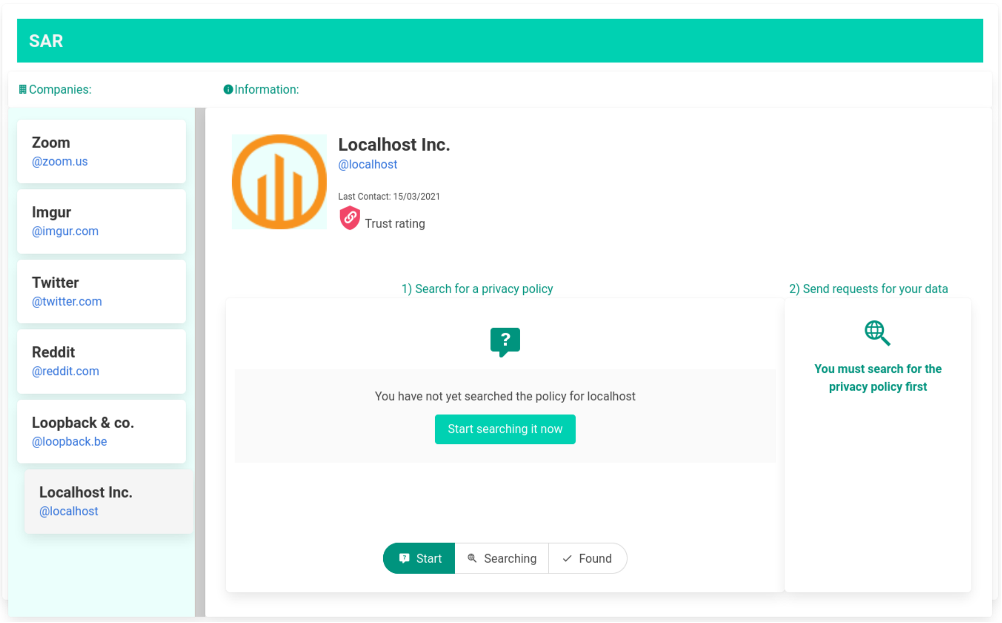


(a)



(b)

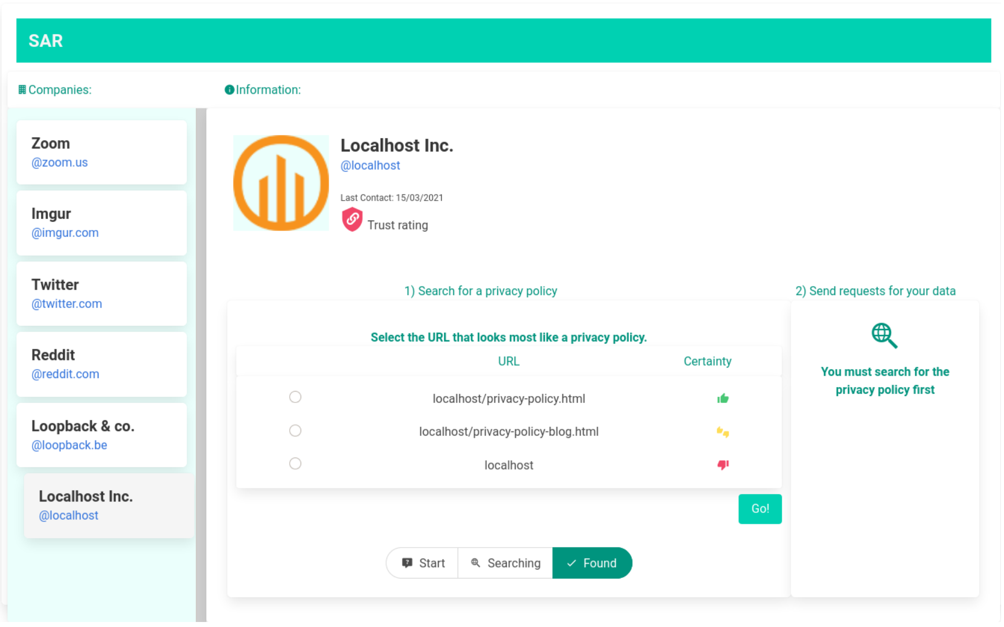
Figure 10.1



You have not yet searched the policy for localhost

You must search for the privacy policy first

Figure 10.2



Select the URL that looks most like a privacy policy.

	URL	Certainty
○	localhost/privacy-policy.html	👍
○	localhost/privacy-policy-blog.html	👋
○	localhost	👎

You must search for the privacy policy first

Figure 10.3

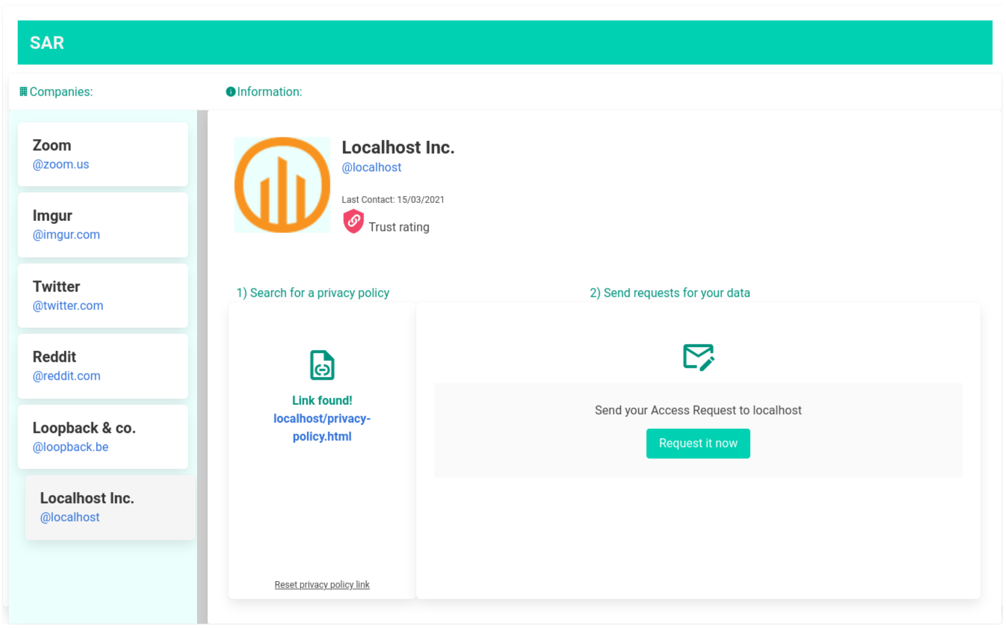


Figure 10.4

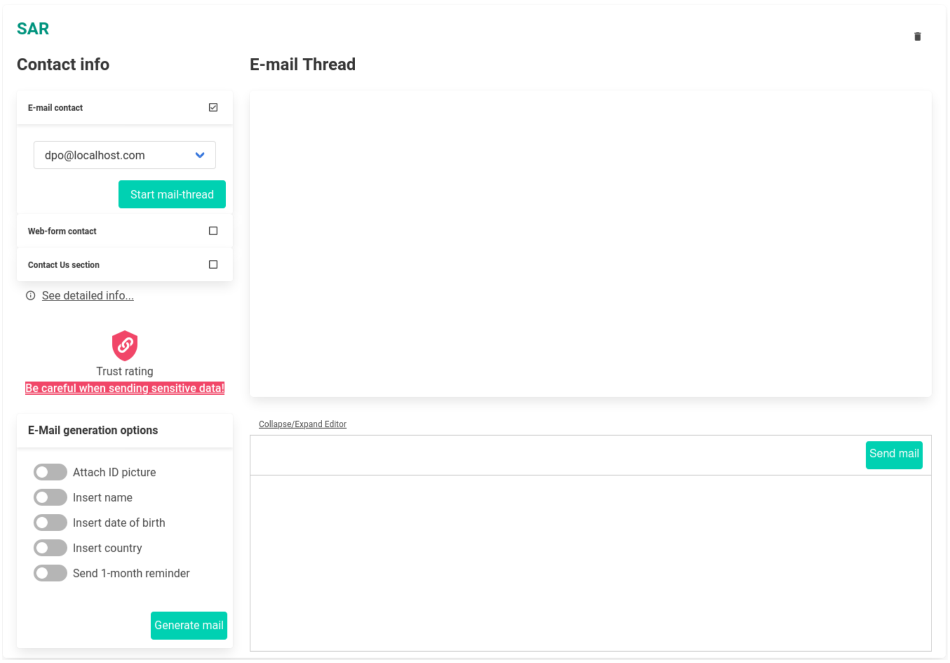
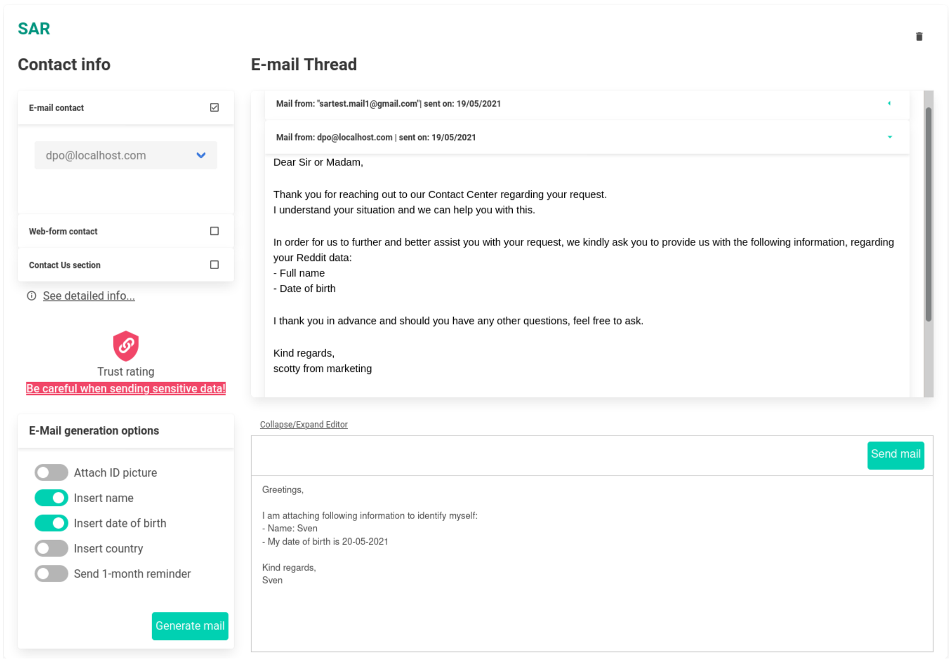


Figure 10.5



SAR

Contact info

E-mail contact ☑

dpo@localhost.com

Web-form contact ☐

Contact Us section ☐

See detailed info...

Trust rating

Be careful when sending sensitive data!

E-Mail generation options

- ☐ Attach ID picture
- ✓ Insert name
- ✓ Insert date of birth
- ☐ Insert country
- ☐ Send 1-month reminder

Generate mail

E-mail Thread

Mail from: "sartest.mail1@gmail.com"| sent on: 19/05/2021

Mail from: dpo@localhost.com | sent on: 19/05/2021

Dear Sir or Madam,

Thank you for reaching out to our Contact Center regarding your request.
I understand your situation and we can help you with this.

In order for us to further and better assist you with your request, we kindly ask you to provide us with the following information, regarding your Reddit data:
- Full name
- Date of birth

I thank you in advance and should you have any other questions, feel free to ask.

Kind regards,
scotty from marketing

Collapse/Expand Editor

Send mail

Greetings,

I am attaching following information to identify myself:
- Name: Sven
- My date of birth is 20-05-2021

Kind regards,
Sven

Figure 10.6

10.2 Nederlandse samenvatting / Dutch summary

10.1 Introductie

In 2016 werd de “General Data protection Regulation” (GDPR) geïntroduceerd [1]. De alternatieve Nederlandse naam hiervoor is de “Algemene verordening gegevensbescherming” (AVG). Het doel was, onder andere, om de privacy wetgeving binnen Europa te verenigen. De wetgeving werd pas ingevoerd in mei 2018.

Één specifiek recht binnen deze wetgeving is het “Recht op inzage”. Dit recht is gedefinieerd in artikel 15 van de GDPR [1]. Wat dit recht in essentie inhoudt is dat een gebruiker van een website of een klant van een bedrijf toegang moet kunnen krijgen tot alle data en informatie dat het bedrijf over hen bezit. In artikel 12(3) wordt vastgelegd dat het bedrijf of verwerkingsverantwoordelijke één maand de tijd heeft om te antwoorden op een aanvraag van de gebruiker. Een mogelijk belangrijk deel van dit proces is het verifiëren van de identiteit van de gebruiker. De persoon die zijn data aanvraagt moet kunnen aantonen dat het effectief zijn data is. Wat dit in essentie betekend is dat een verwerkingsverantwoordelijke extra informatie kan vragen bij een aanvraag.

Dit recht is natuurlijk iets positief voor de inwoners van de Europese Unie maar een recht dat niet gebruikt wordt is niet echt nuttig. De meest gebruikte manier om de rechten te communiceren tussen de verwerkingsverantwoordelijke en de gebruiker is de privacy policy van bijvoorbeeld de relevante website. Nu blijkt echter dat zeer weinig mensen de privacy policies lezen en zelfs in 90% van de gevallen dat ze dat toch doen, deze fout begrijpen [4]. Dit is toch niet de schuld van de gemiddelde gebruiker. McDonald et al. berekenden dat indien een gebruiker elke privacy policy zou lezen, elke gebruiker 244 uur per jaar eraan zou spenderen [5]. Zelfs als gebruikers deze tijd zouden spenderen aan het lezen van policies, zouden veel het niet begrijpen. Proctor et al. ontdekten dat mensen zonder diploma hoger onderwijs de privacy policies meestal niet begrijpen [6].

Deze thesis zal zich focussen op het automatiseren van het recht op inzage. Met andere woorden, er zal getracht worden het proces van recht op inzage te versimpelen voor de eindgebruiker. Dit zal gedaan worden via een proof-of-concept browser extensie. De volgende secties focussen zich op de verschillende stappen van het geautomatiseerde proces. Sectie 10.2 overloopt de high level stappen die daarna in Sectie 10.3 stuksgewijs worden behandeld. Sectie 10.4 behandelt de usability test en bijhorende resultaten. Ten laatste bespreekt Sectie 10.5 de conclusie.

10.2 Structuur van de applicatie

Het doel is, samengevat, om de gebruiker te assisteren bij het uitvoeren van zijn recht op inzage. De effectieve aanvraag van dit recht noemt een “Subject Access Request” (SAR). Het high level proces van zo een SAR is zichtbaar in Figuur 10.7. Een logische eerste stap is natuurlijk dat de gebruiker kan kiezen naar welk bedrijf de SAR wordt gestuurd. Het contactpunt waarnaar deze SAR gestuurd kan worden, staat meestal vermeld in de privacy policy. Eenmaal de gebruiker een bedrijf heeft gekozen kan er naar deze privacy policy gezocht worden. Uit de privacy policy kan dan de nodige informatie gehaald worden zodat de SAR effectief verzonden kan worden. Deze informatie is meestal een e-mail adres of een URL. Figuur 10.7 toont de situatie waar een e-mail adres wordt gebruikt. Eenmaal de initiële aanvraag mail is verzonden, zijn er enkele mogelijke volgende situaties. Enerzijds is er de deadline van een maand, de verwerkingsverantwoordelijke moet een antwoord sturen voor deze deadline. Anderzijds kan de verwerkingsverantwoordelijke ook voor extra informatie vragen om de gebruiker te identificeren. Ten laatste is er natuurlijk de stap waar de gebruiker hun data ontvangt.

Een ander deel van het proces is het vermelden van een “trust rating” of met andere woorden, een soort aantoning van hoeveel vertrouwen er gelegd kan worden in het bedrijf. Deze trust rating is het meest relevant voor de autorisatie en identificatie stap. Er is de mogelijkheid dat een verwerkingsverantwoordelijke om meer data vraagt dan nodig. Indien deze verantwoordelijke eigenlijk weinig vertrouwen verdient, kan dit een probleem vormen. Een gebruiker wil bijvoorbeeld geen scan van zijn identiteitskaart sturen naar een bedrijf dat niet valt te vertrouwen.

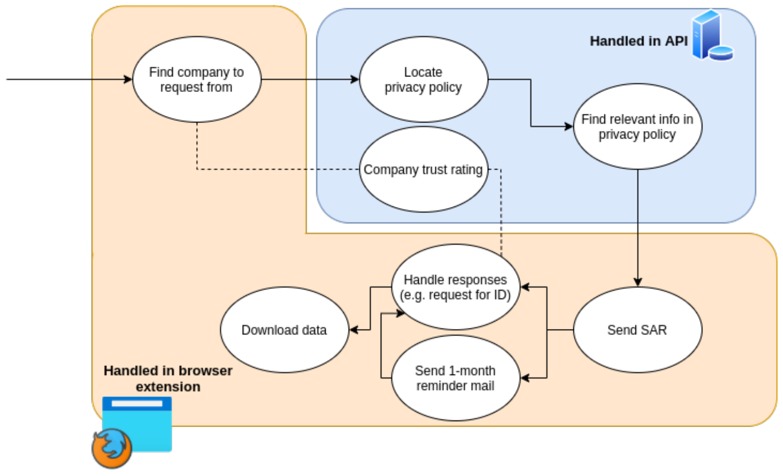

Handled in API
Find company to request from
Locate privacy policy
Find relevant info in privacy policy
Company trust rating
Handle responses (e.g. request for ID)
Download data
Send SAR
Send 1-month reminder mail
Handled in browser extension

Figure 10.7: High level overzicht van het SAR proces.

10.3 Werking van de applicatie

De volgende subsecties duiken dieper op de verschillende stappen van het proces.

10.3.1 De privacy policy vinden

Het doel is op dit moment dus om vanaf een URL te beginnen en dan de privacy policy te vinden. Deze start-URL kan simpelweg de homepage van de website in kwestie zijn. In dit zoeken naar de privacy policy zijn er twee componenten. Enerzijds is er de crawler dat zal kandidaat URLs zal vinden. Anderzijds is er de classifier dat deze kandidaat URLs zal evalueren of classificeren om de kans te bepalen of het al dan niet een privacy policy is.

Een hoofd probleem bij het crawlen van webpagina's is dat het aantal URLs hierop vrij groot kan zijn. Elke URL langsgaan is niet bepaald efficiënt. Sommige websites voorzien ook een extra stap tussen de homepage en effectieve privacy policy. Dit kan voorkomen in situaties waar de website privacy policies voorziet in verschillende talen of voor verschillende regionen. De crawler zal in deze situaties dus, bijvoorbeeld, twee of drie lagen diep moeten zoeken in de "boom" van gelinkte webpagina's. Er moeten dus bepaalde afkortingen gevonden worden zodat het crawlen niet eeuw doorgaat.

Een bepaalde eigenschap van moderne webpagina's is dat de URL naar de privacy policy of de landing page van de privacy policy vaak in de footer of header vermeldt staat. Om deze eigenschap te gebruiken zal er een prioriteit toegekend worden aan de URLs dat gevonden worden. De hoogste prioriteit gaat natuurlijk naar de URLs dat gevonden worden in de footer van de webpagina met het woord "privacy" in de naam. De tweede hoogste prioriteit wordt toegekend aan de URLs dat zich in de body van de webpagina bevinden met het woord "privacy" in de naam. Dit onderscheid wordt gemaakt om de situatie van een privacy gerelateerde blogpost te vermijden. De privacy gerelateerde URL in de footer is met meer waarschijnlijkheid de correcte URL. Alle URLs dat niet in de twee eerder vermelde groepen vallen krijgen de laagste prioriteit. Figuur 10.8 toont de high-level functionaliteit van de crawler. De URLs worden geopend in een Selenium browser instantie. Dit laat de website toe om JavaScript uit te voeren.

Een edge case waar rekening mee gehouden moet worden is het voorkomen dat sommige websites meerdere privacy policies hebben. Sommige websites hebben, bijvoorbeeld, een aparte privacy policy voor Europese gebruikers en Amerikaanse gebruikers. De oplossing hiervoor is ten eerste om alle URLs met hoge prioriteit ook te verwerken wanneer de eerste werd gevonden. Het resultaat is dan een (meestal) korte lijst van een aantal URLs en de bijhorende kans dat het een privacy policy is. Er moet op een manier onderscheid worden gemaakt tussen deze opties. Om de meest accurate keuze hier te maken wordt deze lijst gepresenteerd aan de gebruiker. De gebruiker moet dan uit de top 3 kiezen welke dat de privacy policy is. Om de gebruiker hier bij te assisteren wordt de kans ook getoond via een duim-systeem. Een groene duim betekent een hoge kans dat het een privacy policy is, een gele duim betekent dat er twijfel is en een rode duim betekent dat de kans klein is dat het een privacy policy is.

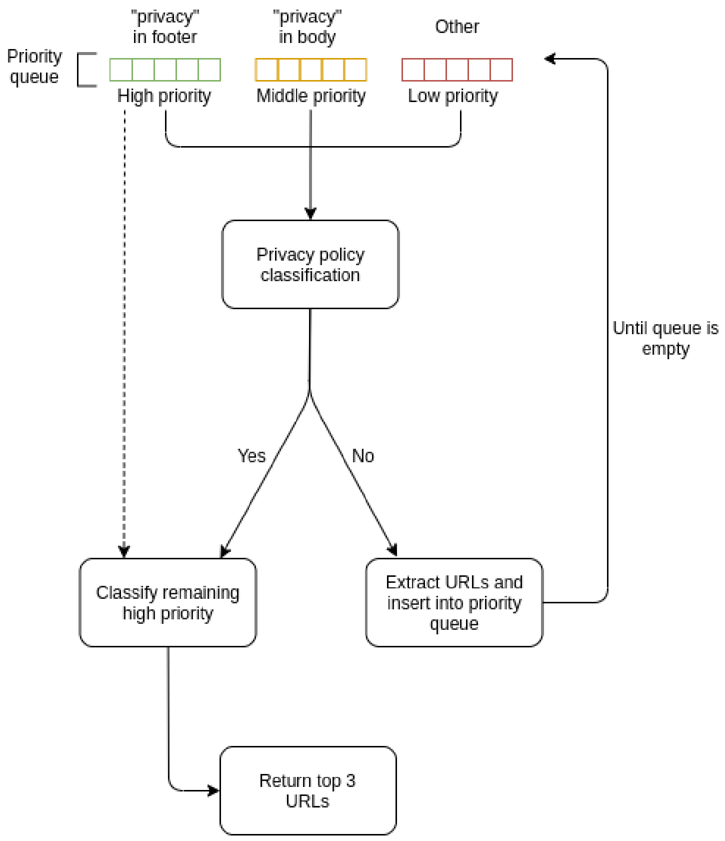


Figure 10.8: High-level crawler diagram

In de voorgaande uitleg werd er gebruik gemaakt van de classificatie en de bijhorende “kans dat het een privacy policy is”. Om deze te bereiken wordt er gebruik gemaakt van machine learning. De HTML van elke pagina dat wordt ingeladen door de Selenium instantie wordt eerst verwerkt eer dit geclassificeerd kan worden. Dit houdt in essentie in dat de tekst omgezet wordt tot een vorm waarmee berekeningen gedaan kunnen worden. Een essentieel onderdeel hieraan is de TF-IDF of “Term Frequency-Inverse Document

Table 10.1: Evaluatie van de classifier en crawler samen

Type	Amount	% of all
Succes	128	85.3
Incorrect	22	14.7
• Taal redir.	4	2.7
• Connectie geweigerd	4	2.7
• async JS	4	2.7
• CAPTCHA	4	2.7
• Applicatie fail	6	4.0
– Mercury Parser	3	2.0
– Classifier	2	1.3
– Crawler	1	0.7

Frequency". De term frequency is simpelweg het aantal keer dat een bepaalde term of woord voorkomt in een bepaald document (de tekst van de privacy policy). De inverse document frequency is een extra term dat zorgt dat er rekening wordt gehouden met de voorkomst van de termen in alle document. $idf(t,d) = log\frac{n_d}{df(t)} + 1$ met n_d het totaal aantal documenten en $df(t)$ het aantal documenten waar de term t in voor komt.

Deze TF-IDF geeft dus een numerieke waarde voor elk woord dat voorkomt in de privacy policy. Deze lijst van waardes of feature vector kan dan aan de logistic regression machine learning classifier gegeven worden. Logistic regression heeft als kenmerk dat de resulterende waarde, in essentie, een kans is dat de input tot een bepaalde klasse behoort (wat in dit geval "privacy policy" is).

Indien we deze classifier alleenstaand evalueren, werd dit gedaan op basis van een k-fold cross-validation. De privacy policy classifier met een $k = 5$ resulteerde in een gemiddelde F1-score van 97.05%. De evaluatie van de crawler en classifier samen werd gedaan door middel van een random sample uit de top 500 Alexa lijst. De resultaten staan samengevat in tabel 10.1. Problemen werden gevonden in het geval van een automatische redirect op basis van taal, CAPTCHAs, asynchrone JavaScript en websites die niet werken. In zes van de gevallen werd een probleem gevonden bij de effectieve implementatie.

10.3.2 Data uit de privacy policy halen

Eenmaal de privacy policy is gevonden door middel van de crawler en de keuze van de gebruiker. Het doel is dan natuurlijk om informatie uit deze privacy policies te halen. De belangrijkste informatie dat hier uit gehaald kan worden is het e-mailadres of de web-form URL waarop de subject access request kan uitgevoerd worden. Hier zijn twee stappen voor nodig. Eerst en vooral willen we de effectieve tekst van de privacy policy uit het HTML-document halen. Websites bevatten vaak boilerplate code die de navigatie balk, footer etc. voorzien in de website. Deze zijn niet van belang voor de informatie dat uit de policy moet worden gehaald.

Om deze boilerplate code te verwijderen uit het HTML-document werd er gebruik gemaakt van de boilerpipe library [33]. In een eerste stap van de text-extractie worden verschillende HTML-tags samengevoegd tot blokken op een text-density meting. De text-density is simpelweg het aantal woorden in de string gedeeld door het aantal lijnen dat de string vult als deze in een kolom met vaste breedte wordt geword-wrapped. De blokken worden samengevoegd tot het verschil tussen de blokken te groot is. Eens de blokken samengevoegd zijn, kunnen ze geclassificeerd worden tot boilerplate of hoofdtekst. Hiervoor wordt de text-density opnieuw gebruikt maar ook de link-density. De link-density wordt berekend door het aantal woorden in een anchor tag te delen door het totaal aantal woorden in de string. Op basis van deze metingen maar ook de waardes van de volgende en vorige blok wordt de klasse van de huidige blok bepaalt. De naburige metingen worden ook in rekening gehouden wegens het feit dat een overgang van boilerplate naar hoofdinhoud veel minder voorkomt dan, bijvoorbeeld, een blok hoofdinhoud dat volgt op een blok van dezelfde klasse.

Uit de resulterende privacy policy tekst moet er nu data gehaald worden. Een eerste stap daarin is het vinden van de sectie in de tekst dat het recht tot inzage behandelt. De aanpak hiervoor is het opdelen van de tekst in samenhangende delen en hieruit dan de juiste te selecteren. Om de tekst op te delen wordt er gebruik gemaakt van het TextTiling algoritme [34]. Dit algoritme deelt een tekst op in multi-paragraaf blokken die subonderwerpen representeren. Het hoofdconcept bij het TextTiling algoritme is de introductie van nieuwe termen. In een doorlopende tekst zullen er, wanneer er van onderwerp veranderd wordt, nieuwe woorden geïntroduceerd worden. De similariteit tussen twee blokken kan berekend worden op basis van de termen dat er in voorkomen. Indien er een sterke verlaging in deze similariteit gevonden worden, is er een goede kans dat dit een overgang is tussen verschillende subonderwerpen.

Uit deze lijst van samenhangende secties moet de sectie over recht tot inzage gevonden worden. Hiervoor kunnen opnieuw ingewikkelde oplossingen voor gevonden worden zoals machine learning maar een simpele aanpak volstaat waarschijnlijk ook. Het jargon dat gebruikt wordt omtrent "right of access" is vrij uniek. Een voor de hand liggend voorbeeld is natuurlijk "right of access" zelf, maar ook meer generieke termen zoals "exercise (your right)" komen vaker voor. Op basis van de voorkomst van zulke termen kunnen de meest relevante secties gekozen worden. Uit deze meest relevante secties kan dan met regex het e-mailadres of URLs gehaald worden.

10.3.3 Trust rating

In het proces om het recht tot inzage uit te voeren, kan het voorkomen dat een gegevensverantwoordelijke vraagt voor extra informatie om, bijvoorbeeld, de gebruiker te authenticeren. Hierbij kan gevraagd worden voor, bijvoorbeeld, een scan van de identiteitskaart. Dit is natuurlijk niet iets wat naar zomaar iedereen gestuurd moet worden. Het idee is dus om de gebruiker een bepaald idee te geven van in welke mate het bedrijf te vertrouwen valt.

Een groot probleem hiermee is dat deze waarde zeer subjectief is. Verschillende mensen kunnen verschillende factoren als een positieve factor zien terwijl anderen het als negatief interpreteren. Een voorbeeld hiervan is middelgrote internationale bedrijven. Deze investeren waarschijnlijk meer in hun security, maar zijn ook interessante doelwitten voor hackers. Een eerste aanknopingspunt voor het bepalen van een niveau van vertrouwen voor een bedrijf is twee extremen besluiten. Een verre van te vertrouwen website is een phishing website. Hiervoor wordt gebruik gemaakt van een bestaande phishing website database genaamd PhishTank[1]. Er is ook de mogelijkheid om bepaalde automatische systemen te gebruiken voor phishing detectie maar deze zijn waarschijnlijk te uitgebreid voor het doel dat bereikt moet worden. De te vertrouwen uiteinde berust op de gedachtegang dat de meest populaire websites veel meer investeren in security en het correct opvolgen van de GDPR-wetgeving. Om deze te detecteren kan er gebruik gemaakt worden van de Alexa top websites lijst.

De gradaties tussen deze twee extremen moeten natuurlijk ook bepaald worden. Hierbij is een eerste insteek dat bepaalde factoren het vertrouwen gaan verlagen. Een factor dat voor zichzelf spreekt is de afwezigheid van HTTPS. Ook de aanwezigheid van verdachte woorden ("secury", "login") of de aanwezigheid van veel subdomeinen kunnen gebruikt worden. Daarnaast bestaan er ook top-level domain names die gratis te gebruiken zijn

[1]https://www.phishtank.com

(".tk", ".ml"). Bedrijven die niet investeren in een betaalde top-level domain investeren waarschijnlijk ook niet veel in security.

Een laatste factor is de security management van een specifiek bedrijf in het verleden. Indien een bedrijf in het verleden een datalek heeft gehad, gaat het algemene vertrouwen waarschijnlijk omlaag. Met behulp van een nieuws-artikel API kan er gezocht worden naar artikels die de naam van het bedrijf vermelden samen met termen zoals "data leak" of "breach". Indien deze gevonden wordt, kan de trust rating verlaagd worden.

10.3.4 e-mailcorrespondentie: toekomstig werk

Initieel werd er gedacht om de correspondentie tussen de gegevensverantwoordelijke en de gebruiker ook in bepaalde mate te automatiseren. In de zin dat er op basis van het antwoord automatisch er al een bepaalde mail gegenereerd wordt. Een probleem hierbij is dat meeste technieken voor automatische mail generatie berusten op de aanwezigheid van een dataset om de detectie op te baseren. Voor zover we weten bestaat er geen dataset van SAR-correspondentie die vrij beschikbaar is. Om deze reden wordt deze stap gelaten aan toekomst werk. Er zal kort besproken worden op welke manier dit mogelijk zou zijn.

De aanpak dat volgens ons het beste zou zijn voor de situatie van een browser extension zijn tekst patronen [51]. De tekst wordt opgesplitst in paragrafen en elke paragraaf wordt gematcht met de lijst van tekst patronen. Indien het tekstpatroon matched en het "verboden" patroon niet matched, kan het vooraf gedefinieerd antwoord gebruikt worden om een mail te genereren. De grootste drawback hieraan is de afhankelijkheid van de taal. In de huidige implementatie wordt de gebruiker een aantal switches getoond waarmee ze een gewenste mail kunnen laten genereren.

10.4 Experimentele resultaten

Als evaluatie van de complete browser extension werd er een usability test uitgevoerd. Hiervoor werden tien participanten gezocht en gecontacteerd via mail waarin ze een zelf-gekozen moment voor de test konden nemen. De participantengroep is vrij biased naar mensen die bekend zijn met computers en informatica. Van de tien deelnemers zijn er zeven die momenteel informatica studeren. De drie overblijvende studeren geen informatica maar twee hiervan zijn vrij regelmatige computer gebruikers.

Er werden drie scenario's voorgelegd aan de participanten. Eerst en vooral een normale situatie om het algemene idee van de extension vertrouwt te maken. In het tweede scenario zal de gebruiker actief keuzes moeten maken uit de top 3 policy lijst en uit de contactopties. Het laatste scenario focust op de trust rating. Het idee in deze laatste situatie is dat de applicatie vermeld dat het bedrijf niet te vertrouwen is door een data leak in het verleden. De gegevensverantwoordelijke in deze situatie zal de participant vragen voor een kopie van zijn of haar identiteitskaart.

In scenario één werden er geen grote problemen gevonden. Een bepaalde opmerking dat door meerdere mensen werd aangehaald, is dat er een barrière is over het jargon dat wordt gebruikt. In scenario twee ontstonden er meer problemen. De participant die het meest onbekend is met computers ondervond problemen bij het selecteren van de correcte policy URL. Het kan aangenomen worden dat de keuze overweldigend was. Gebruikers die niet bekend zijn met privacy policies en URL-structuur begrijpen er weinig van. Het duimpjes-systeem assisteert de gebruiker misschien niet genoeg bij het maken van de keuze. Een participant maakte de suggestie om screenshots toe te voegen.

Eenmaal de correcte URL was gekozen, moest de participant ook de juiste contactoptie selecteren. Hierbij werd duidelijk dat er misschien een fout werd gemaakt bij de opzet van de opdracht. In het eerste scenario was er maar één optie, een e-mail adres. Hierdoor zijn er meerdere participanten die uit gewoonte simpelweg de e-mail contact optie namen. Één participant maakte gebruik van de "detailed info" pop-up scherm. De andere participanten maakten hun keuze uit gewoonte of op basis van een intuïtie.

In het laatste scenario was de meest voorkomende feedback dat er geen alternatief werd aangereikt voor de situatie waar ze de scan van de identiteitskaart niet willen versturen. Dit is natuurlijk deels te verklaren door het tekort aan automatische correspondentie. Één participant suggereerde om een andere manier van autorisatie te vragen aan de gegevensverantwoordelijke.

10.5 Conclusie

De applicatie dat werd ontwikkeld, heeft zich vooral toegespitst op het overslaan van de privacy policy in het SAR proces. Die manier waarop dit gedaan wordt is niet perfect. Er zijn enkele edge cases, maar deze zullen altijd bestaan. CAPTCHAs specifiek zijn moeilijk om rond te raken. De effectieve assistentie bij de correspondentie is zeer simpel, maar het is beter dan niets.

Van onder andere de usability test kan afgeleid worden dat de browser extension, zoals het momenteel is geïmplementeerd, niet erg bruikbaar is voor de normale, dagdagelijkse gebruiker. We argumenteren dat de beste doelgroep voor een dergelijke applicatie in elk geval de gebruikers zijn die al bekend zijn met gelijkaardige onderwerpen. Het versturen en vervolledigen van een subject access request is één stap in een groter proces. Het is niet vergezocht om te zeggen dat de dagdagelijkse gebruiker zeker moeite gaat hebben met het begrijpen met de data die ze ontvangen. Indien ze deze begrijpen en, bijvoorbeeld, merken dat de gegevensverantwoordelijke te veel data verzamelen, weet de leek wie of wat ze moeten contacteren om dit aan te geven? Een positief aspect is dat indien non-leken een dergelijke tool gebruiken om meer SARs te sturen, dat bedrijven misschien een versimpelde manier introduceren.