Henry Hosseini*, Martin Degeling, Christine Utz, and Thomas Hupperich*

Unifying Privacy Policy Detection

Abstract: Privacy policies have become a focal point of privacy research. With their goal to reflect the privacy practices of a website, service, or app, they are often the starting point for researchers who analyze the accuracy of claimed data practices, user understanding of practices, or control mechanisms for users. Due to vast differences in structure, presentation, and content, it is often challenging to extract privacy policies from online resources like websites for analysis. In the past, researchers have relied on scrapers tailored to the specific analysis or task, which complicates comparing results across different studies.

To unify future research in this field, we developed a toolchain to process website privacy policies and prepare them for research purposes. The core part of this chain is a detector module for English and German, using natural language processing and machine learning to automatically determine whether given texts are privacy or cookie policies. We leverage multiple existing data sets to refine our approach, evaluate it on a recently published longitudinal corpus, and show that it contains a number of misclassified documents. We believe that unifying data preparation for the analysis of privacy policies can help make different studies more comparable and is a step towards more thorough analyses. In addition, we provide insights into common pitfalls that may lead to invalid analyses.

Keywords: privacy policy, data handling, policy detector, natural language processing

DOI 10.2478/popets-2021-0081
Received 2021-02-28; revised 2021-06-15; accepted 2021-06-16.

***Corresponding Author: Henry Hosseini:** University of Münster & Ruhr University Bochum, E-mail: henry.hosseini@wi.uni-muenster.de
Martin Degeling: Ruhr University Bochum, E-mail: martin.degeling@ruhr-uni-bochum.de
Christine Utz: Ruhr University Bochum, E-mail: christine.utz@ruhr-uni-bochum.de
***Corresponding Author: Thomas Hupperich:** University of Münster, E-mail: thomas.hupperich@wi.uni-muenster.de

1 Introduction

Protecting personal data in times of pervasive online data collection raises challenges for website operators, visitors, and legislators. Such pervasive technologies have increasingly raised ethical concerns by privacy researchers [1], leading regulators to create guidelines like the Fair Information Practice Principles [2] and pass new legislation, including the European Union's General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA). These laws aim to provide individuals with legal means to gain some control over their personal data. One central principle to achieve this is transparency by informing people about a company's or service's data processing practices and individuals' rights regarding the use of their personal data, such as opting out of certain types of data collection. As for web technologies, the established methods of informing are privacy notices such as privacy policies and cookie banners [3].

Privacy and cookie policies have received increasing attention by researchers over the last decade, with investigated aspects including content analysis [4], summarization and key phrase analysis [5–7], readability measurements [8], annotation of privacy practices [9, 10], training machine learning and deep learning models [11], and the recent discovery of potential security and privacy concerns [12–14].

The majority of these studies are based on a few initial but crucial steps for privacy policy retrieval and extraction:

1) identification of potential privacy and cookie policies on websites,
2) extraction of relevant text from scraped HTML and PDF files,
3) detection of the language of the extracted text,
4) distinguishing privacy policies from other texts (non-privacy policies),
5) and storage of the plain text along with associated metadata.

In this paper, we show that if each of these steps is not performed carefully, entire studies based on the resulting corpus of privacy policies could lack correctness and completeness: If a website's privacy policy, cookie policy, or both are not detected correctly in step 1), the re-

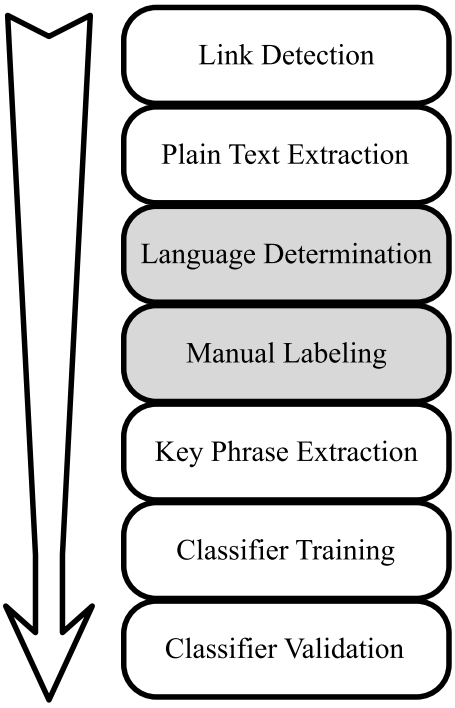

Fig. 1. Overview of the toolchain. The gray components are optional, depending on the processed data sets.

sulting corpus can contain irrelevant documents such as (chocolate) cookie recipes, news articles about privacy topics, and error messages. Additionally, certain privacy policies might be missing from the corpus because they could not be detected. In step 2), if a privacy policy is not correctly extracted from HTML code, either critical information is lost or the resulting plain text contains noise such as headings or HTML tags. In step 3), tasks involving natural language processing, such as searching for specific phrases or topic modeling, depend on the correct detection of the privacy policy's language. In step 4), correctly distinguishing privacy and cookie policies from other texts requires labeled data from which useful features can be crafted to train machine learning or deep learning classifiers.

To aid future studies on privacy policies, we introduce a new toolchain to perform these essential steps, based on best practices we identified. Our toolchain consists of multilingual heuristics for privacy policy link detection, text-from-HTML and text-from-PDF converters, an ensemble of language detection libraries, well-established key phrase extractors, and trained machine learning classifiers for English- and German-language privacy and cookie policies. For each step, we identify best practices. In summary, we make the following contributions to the field of privacy policy analysis:

1. We compare approaches to find privacy policies on websites, as well as different text-from-HTML extractors, which we evaluate on privacy policies. Out of seven text extraction libraries, Boilerpipe with the NumWordsRules setting performs best.
2. We present machine learning classifiers trained with the most important key phrases extracted from privacy policy texts. They achieve a balanced accuracy of 99.1 % and 99.6 % for policies in English and German, respectively.
3. While the large majority of previous research focused on English privacy policies, our toolchain provides a means to process non-English privacy policies, fostering cross-language research.

Figure 1 shows the high-level structure of our toolchain. In Section 2, we provide an overview of the approaches in previous work to detect and extract privacy policies. For each component in our toolchain, Section 3 describes our approach and Section 4 evaluates its performance. Finally, in Sections 6 and 7, we discuss our findings, provide ideas for future research, and conclude this work.

2 Related Work

Privacy policies have been widely studied for different aspects, from their prevalence [15, 16] to their content to allow for automatic extraction of important information. The focus has been mostly on the privacy and cookie policies of websites and mobile apps. Table 1 provides an overview of previous research and shows the various methods used to preprocess privacy policies. The majority of studies used their own sets of tools to collect and preprocess privacy policies from websites. Our work draws on and compares different approaches used in the past for text extraction, language detection, classification, and evaluation.

Liu et al. [17] addressed the problem of extracting privacy policy text and hired three Amazon Mechanical Turk workers to manually find privacy policies on websites and extract their text, leading to 1,010 unique privacy policies with manually collected metadata. Boldt and Rekanar [23] also manually extracted the plain texts of the 100 privacy policies in their corpus.

Libert [20] applied the library Readability.js to extract the policy text and titles from websites loaded in Chrome or PhantomJS. Policies were sanity-checked for the presence of the terms "privacy" or "cookie." Spot checks for containing only policy text led to an accuracy of 100 %. Ramadorai et al. [22] analyzed a compre-

Table 1. Comparison of related work

Study	Year	#	Extraction	Lan.	Classifier	Comment
Liu et al. [17]	2014	1,010	Manual/MTurk	EN	-	
Yu et al. [18]	2016	1,197	Beautiful Soup (BS)	EN	Links from app store	
Fabian et al. [8]	2017	49,036	Boilerpipe	EN	Comparison of three	Decision Tree 91 % F1
Gopinah et al. [19]	2018	152	ASDUS	EN	Manual	
Libert [20]	2018	184,897	Readability.js	EN	Contain "cookies" or "privacy"	
Fukushima et al. [21]	2018	32	Manual	JP	Manual	
Zaeem et al. [6]	2018	400	Manual	EN	Google Prediction API	
Harkous et al. [11]	2018	~130,000	segmenter	EN	-	
Ramadorai et al. [22]	2019	4,078	-	EN	Contain "privacy"	F1 85 % to 87 %
Boldt and Rekanar [23]	2019	100	Manual	EN	Comparison of 15 classifiers	Naïve Bayes was best
Degeling et al. [15]	2019	5,091	Boilerpipe	Multi	-	
Zimmeck et al. [24]	2019	~500,000	-	EN	Links from app store	99 % F1 on test set
Sarne et al. [25]	2019	4,982	Goose	EN	Links from app store	
Hosseini et al. [26]	2020	100	Beautiful Soup	EN	Links from app store	
Kumar et al. [27]	2020	236	Mercury Parser & BS	EN	Same as [24]	
Linden et al [28]	2020	6,278	Boilerpipe & BS	EN	based on [29]	
Srinath et al. [16]	2020	1,005,781	Dragnet	EN	Random Forest	F1 97 %
Amos et al. [30]	2020	1,071,488	Readability.js	EN	Own classifier	Random Forest 95 % F1

hensive set of privacy policies from US companies for different aspects such as quality, readability, and ease of access. They applied automated Google searches and web crawling techniques to scrape privacy policies and kept a text only if it contained the word "privacy."

Yu et al. [18] used the Python library Beautiful Soup [31] for text extraction, followed by removing non-ASCII symbols. Their corpus only included English privacy policies, but we could not find any information about language detection. The work of Hosseini et al. [26] applied a nearly identical approach.

Kumar et al. [27] applied a text-from-HTML pipeline that used the Postlight Mercury Parser API [32] to identify the website's main content, followed by Beautiful Soup and the lxml [33] parser to construct the website's Document Object Model (DOM) tree to extract the plain text segments. A similar approach was taken by Sathyendra et al. [34]. The (English) privacy policy classification was performed on raw HTML with a logistic regression classifier – the same as used by Zimmeck et al. [24], who report a 99.0 % accuracy and an F1-score of 99.2 %. Manual inspection was performed to remove false positives and non-English privacy policies.

Linden et al. [28] analyzed GDPR-related changes in 6,278 English privacy policies using syntactic text features and a user study. They identified privacy policy candidates using regular expressions to search for specific keywords. Language identification was conducted using LangID [35], and body text extraction was performed with Boilerpipe [36] and Beautiful Soup. Privacy policies were classified with a one-layer convolutional neural network based on Kim's [29] work on sentence classification. The classifier was trained on 1,600 web pages and achieved 99.09 % accuracy for 400 web pages in the test set. The training set was the ACL/COLING Corpus of 1,010 privacy policies collected by Ramanath et al. [37], while the test set consisted of a selection of unrelated texts whose title or URL did not contain keywords associated with privacy policies. Ramanath et al. [37] applied an unsupervised hidden Markov model to align similar segments of privacy policies. They hired Amazon Mechanical Turk workers to manually collect their set of privacy policies.

Wilson et al. [9] created a manually annotated privacy policy corpus. They manually verified each privacy policy for being written in the English language. Sarne et al. [25] performed unsupervised topic extraction from a set of 4,982 privacy policies from the Google Play App Store and, after manual review by an expert, compared their resulting topics to the work of Wilson et al. [9], searching for new topics that could be observed due to new privacy regulations such as the GDPR. They applied the Goose library [38] to convert the scraped documents to text, followed by langdetect [39] to filter all non-English privacy policies. Zaeem et al. [6] created the *PrivacyCheck* Google Chrome extension to automatically summarize privacy policies using a keyword-based approach. A classifier was trained using the English-only Google Prediction API to distinguish between privacy policies and non-privacy policies.

Several projects have addressed summarization to present long privacy policies in a condensed way. Audich et al. [7] evaluated five key phrase extraction algorithms on 21 privacy policies against manually extracted key phrases, achieving a maximum F1 score of 27 %. They conclude that key phrase extraction algorithms evaluating a single document perform better than those operating on the entire corpus. In a follow-up study [10], they trained a supervised key phrase extraction model using the KEA algorithm, which outperforms unsupervised key phrase extraction methods. Tomuro et al. [5] used keywords identified by a human domain expert for ensemble learning to identify the most important sentences in a privacy policy.

Fabian et al. [8] studied the readability of English privacy policies and designed a crawler using Boilerpipe for text extraction. They compared three machine learning classification algorithms to determine whether an extracted text is a privacy policy. The best approach turned out to be a decision tree with an F1 score of 90.8 %. A language check was performed to keep only English texts, but no information on its performance was provided. Tesfay et al. [40] created a privacy policy summarization tool in Java based on the GDPR. They also used Boilerpipe to extract the main content of privacy policies.

Harkous et al. [11] applied a segmenter to remove irrelevant HTML elements from privacy policy pages and achieved a 99.08 % coverage compared to manually extracted policies from the 200 most popular websites according to the global Alexa.com ranking. Degeling et al. [15] used a similar approach and identified privacy policies on websites as HTML links containing specific phrases in 46 languages.

Srinath et al. [16] created a searchable corpus of web privacy policies. They used LangID for language detection and the Dragnet [41] Python package for content extraction. To distinguish between privacy policies and non-privacy policies, they labeled 1,000 documents and trained a random forest classifier using word-based and URL-based features, achieving an F1 score of 97 %. To summarize privacy policies, RAKE [42] and TextRank [43] were applied.

Gopinath et al. [19] created *ASDUS*, a Java tool for robust separation of section titles and text in web documents, irrespective of the HTML structure. ASDUS achieved precision and recall of 82 % and 98 %, respectively, based on a comparison with 100 manually annotated policies. The authors applied the Java HTML parser jsoup [44] to extract tuples of text and its XPath.

Comparing this related work, Table 1 illustrates an existing focus on the English language, although privacy policies are equally important outside the Anglosphere. Moreover, only a limited number of studies have evaluated whether the analyzed texts actually represent privacy policies. Especially studies that focused on automated content analysis often used privacy policies of mobile apps, which are easy to locate due to the uniform listings in app stores containing a link to the app's privacy policy. Approaches to identify privacy policies on websites vary and include searching for links containing specific words, looking for specific URLs (e. g., `/privacy`), or using a search engine. Various papers only mention that the policies were scraped/collected and extracted/sanitized, without reporting on performance or providing further details [45–49].

3 Approach

In the following, we describe the privacy policy corpora we used in this work and each component of our toolchain for the preprocessing of privacy policies.

3.1 Privacy Policy Corpora

Related work has created different corpora of privacy policies, mostly in the English language. As a first step towards the multi-language analysis of privacy policies, this work also covers policies in the German language. For both languages, separately labeled data is required to train classifiers for privacy policy detection. Depending on the state of the corpus, it can be necessary to manually label texts as privacy policies, cookie policies, or neither. To avoid redundancies in the labeling process, we partially leveraged sanitized English corpora from previous work and a subset of the multilingual data set compiled by Degeling et al. [15]. In the following, we describe each of these instrumented data sets.

GDPR-2019: Degeling et al. [15] collected a multilingual data set of privacy policies to find evidence for GDPR-related changes. Between April 2016 and November 2018, they automatically visited the top 500 most popular domains for 28 European countries according to the Alexa website ranking. The sites were scraped for pages that were presumed to be privacy or cookie policies if links to them contained specific keywords of a multilingual list. The raw data set consists of 127,328 scraped web pages presumed to be privacy

or cookie policies. After data cleaning, which included discarding duplicates and searching for common web error messages, it comprises 81,617 privacy policies from 9,461 different URLs and 7,812 domains. Since the texts in the data set had not been labeled as privacy/cookie policies or other texts, we manually labeled parts of the German and English subsets of this corpus as described in Section 3.4.

Rogue-Top-100: This corpus consists of 167 English privacy policies from both legitimate and "rogue" companies collected in 2016 [23]. The 100 legitimate privacy policies belong to the websites of the companies on the Fortune Global 500 list. 67 privacy policies were collected from the websites of rogue companies as classified by SpywareGuide.com. We included this data set to generalize our classifiers beyond the usually well-written privacy policies of large companies.

APP-350 contains 350 annotated English privacy policies of Android apps [24]. We chose to include this corpus to generalize our approach to the privacy policies of mobile apps, which are increasingly studied in privacy policy research to reflect that Internet usage is shifting away from desktop computers and browsers towards mobile devices and designated apps [50].

OPP-115 is a corpus of 115 privacy policies from English-language websites, manually annotated for data and privacy practices [9]. The websites for this corpus were selected from 15 categories such as shopping, business, and news. We include this corpus for its variety and detail of annotation.

Princeton-2020: This corpus contains over 1 million privacy policies from 1990 up to 2019, collected from the Internet Archive's Wayback Machine [30]. The decision whether a text is a privacy policy was not based on manual labeling but on the output of a trained machine learning classifier. Therefore, it cannot be considered a gold standard, and only a comparison of the assigned labels can be performed. As this corpus also contains non-privacy policies, it provides us with the necessary data to evaluate our toolchain, so we did not use it to train our classifiers but only to evaluate and compare.

Privacy news articles: We include two corpora of news articles about privacy topics to enhance our toolchain's ability to distinguish between privacy/cookie policies and texts that use privacy-related terminology but are not privacy/cookie policies:

- The privacy incident database by Murukannaiah et al. [51] contains 408 English news articles about privacy incidents. We managed to scrape 386 of these articles since 22 were not available anymore.
- As we are not aware of any comparable German corpus, we collected 112 privacy-related articles from popular German (tech) news websites, mainly Heise online, WinFuture and Bayerischer Rundfunk, published between October 2018 and January 2021.

3.2 Text Extraction

For precise extraction of clean privacy policy texts from their HTML pages, we identified plain-text-from-HTML extractors used in previous research. Other output options such as Markdown could be beneficial to identify headers and titles and output formatted text. We tested the following extractors:

1. Boilerpipe [36]
2. HTML2Text [52]
3. Inscriptis [53]
4. Readability.js [54]
5. Goose3 [38]
6. Beautiful Soup [31]
7. Dragnet [41]

The ideal extractor is language-independent for a large variety of applications. We still included the only language-aware extractor on this list, Goose3, because it was used in previous work. During the evaluation described in Section 4.1, we noticed that Beautiful Soup's output with any engine often contained script tags and CSS code. We still kept it in our set of text extractors because it is language-independent and widely used in the literature.

In addition, we include a text-from-PDF extractor, PyMuPDF [55], to cover the rare case in which a website provides its privacy policy only in PDF format.

3.3 Language Determination

Privacy policies can pose challenges for automatic language detection. For instance, some websites provide their privacy policy in multiple languages on a single web page, an example of which is shown in Figure 2. Such texts need special handling as natural language processing libraries are language-dependent and would output incorrect results, e. g., in stemming or removing stop words.

Multilingual language detection may require high effort and language expertise if done manually, with difficulty further increased by figurative speech and common words across languages. Thus, for large corpora, like GDPR-2019 and Princeton-2020, manual language detection is not feasible. To still ensure the highest possible accuracy in automated language detection, we instrumented an ensemble of the following eight popular

open-source libraries for language detection and performed a majority voting on their output:

1. Apache Tika [56]
2. Compact Language Detector v2 (CLD2) [57]
3. Compact Language Detector v3 (CLD3) [58]
4. Guess-language [59]
5. LangID [35]
6. Langdetect [39]
7. Textacy [60]
8. fastText [61, 62]

The trained model of fastText is able to detect 176 languages. Training data includes Wikipedia articles, Tatoeba [63], which is a collection of sentences and their translations, and SETimes [64], which consists of news articles in English. Textacy includes a trained model with a macro and micro F1 score of 96 % to identify 140 languages. Besides Tatoeba and Wikipedia, we used other training texts such as news and journalistic articles [65, 66], the Universal Declaration of Human Rights, and a collection of Tweets in 70 languages [67] to train this model. Langdetect applies naïve Bayesian filters and character n-gram language profiles created using Wikipedia to detect 53 languages with a precision of over 99 %. LangID is able to detect 97 languages and features a model insensitive to domain-specific features such as HTML/XML code. It was trained using several sources including JRC-Acquis [68], a multilingual aligned parallel corpus with more than 20 official European languages, the ClueWeb09 data set [69] consisting of over 1 billion web pages in 10 languages, Wikipedia articles, the Reuters RCV2 [70] corpus consisting of 487,000 Reuters news stories in 13 languages, and Debian i18n [71], the software internationalization project of Debian Linux. Guess-language utilizes character trigrams to detect 60 languages. According to its documentation, and in contrast to LangID, HTML/XML tags and scripts lead to incorrect results. Google's CLD3 includes a neural network that utilizes a trained model to detect 107 languages. Character unigrams, bigrams, and trigrams serve as the input of the neural network model. CLD2 is the previous version of CLD3, capable of detecting 83 languages and receiving plain text or HTML/XML as input. The design target is web pages of at least 200 characters. Internally, it uses a naïve Bayes classifier, whose input can be either unigrams or quadgrams. The training data included at least 100 million web pages. Both CLD2 and CLD3 are capable of detecting multilingual text. CLD2 outputs the indices of each detected text span in a specific language, which solves the problem of segmenting multilingual privacy policies.

Some of the examined libraries also output the confidence scores of the returned language(s). Due to the different metrics used, we did not incorporate them into our toolchain.

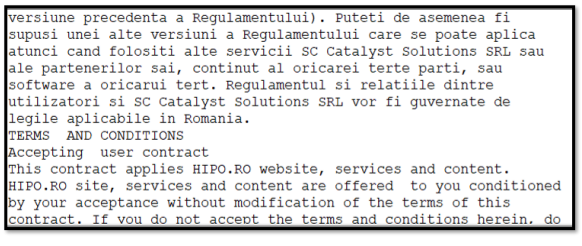
versiune precedenta a Regulamentului). Puteti de asemenea fi supusi unei alte versiuni a Regulamentului care se poate aplica atunci cand folositi alte servicii SC Catalyst Solutions SRL sau ale partenerilor sai, continut al oricarei terte parti, sau software a oricarui tert. Regulamentul si relatiile dintre utilizatori si SC Catalyst Solutions SRL vor fi guvernate de legile aplicabile in Romania.
TERMS AND CONDITIONS
Accepting user contract
This contract applies HIPO.RO website, services and content. HIPO.RO site, services and content are offered to you conditioned by your acceptance without modification of the terms of this contract. If you do not accept the terms and conditions herein, do

Fig. 2. Example of a multilingual (Romanian/English) privacy policy scraped in January 2018. Such policies challenge language detection libraries and natural language processing toolkits.

3.4 Training Sets for Privacy Policy Detection

In order to prevent noise in privacy policy corpora, this step aims to distinguish as accurately as possible between privacy and cookie policies on one hand and other texts on the other. Our first idea was to filter the "other" category by searching for predefined error terms such as "error 404" in English and "Fehler 404" in German. Unfortunately, the possible wordings of error messages are unlimited and would not necessarily lead to filtering all texts in the "other" category. A similar problem arises when "other" texts are filtered if they lack specific terms such as "privacy policy" in English and "Datenschutzerklärung" in German. The main challenge here is to correctly assign the "other" category to texts such as terms of service and privacy-related news articles, which use a vocabulary similar to privacy policies and might have been scraped from web pages due to their URL or text body containing keywords such as "privacy" or "cookie" (see Sections 3.6 and 5.3).

Our proposed strategy to detect privacy and cookie policies is to craft training sets from various data sources and use them to train machine learning classifiers. For the English privacy policy classifier, we could leverage previously published corpora of privacy policies and privacy news articles as positive and negative training data, respectively (see Section 3.1). However, no comparably sanitized corpora exist in the German language. Therefore, as described in Section 3.1, we collected German privacy news articles to include them in our training set as negative samples. We also manually labeled 4,231 German texts from the GDPR-2019 corpus to use them in the German training set.

For the latter, we applied a hybrid approach, i. e., text clustering followed by labeling the resulting clusters instead of individual texts. Each cluster was assigned '1' for privacy/cookie policies or '0' for other text. This allowed us to speed up the process by labeling multiple similar texts at once. One of the authors manually quality-checked the assigned values in each cluster. Our initial attempts to apply k-means clustering with cosine distance and bag-of-words failed because the resulting clusters did not exhibit any clear pattern. Instead, we trained for each German and English set of texts from the GDPR-2019 corpus two Doc2Vec [72] models, one using the distributed memory algorithm (DM) and the other using distributed bag-of-words (DBOW). The input for both models consisted of the lowercase tokens of each text without punctuation and white space. Even though there are other NLP toolkits with a higher tokenization accuracy [73, 74], we used the Spacy library [75] due to its justifiable speed-accuracy trade-off on our large corpora as the more accurate alternatives were significantly slower.

We trained the Doc2vec models using the Gensim library [76] and concatenated the vector representations of each of the DM and DBOW models into one vector of size 200 for each document. This approach was inspired by Gómez-Adorno et al. [77], who found that concatenation of the embeddings works best to obtain meaningful representations. Next, we calculated the cosine distance metric between the vector representations of all texts and clustered them using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [78] with a maximum cosine distance threshold of 0.1. This yielded the best fine-grained clustering result with 3,675 and 963 distinct clusters for English and German texts, respectively. These clusters could be grouped into the following categories:

1. Texts from the "other" category such as web errors (e. g., HTTP 404), CAPTCHAs, disclaimers, and privacy-related news.
2. Privacy policies, often belonging to a single domain, crawled before the enforcement date of the GDPR (25 May 2018).
3. Privacy policies, often belonging to a single domain, crawled after the enforcement date of the GDPR.

To improve data quality, we labeled 291 of the German clusters by hand. For English, the use of pre-existing corpora (see Section 3.1) provided us with enough positive samples, so we mainly searched for the largest clusters that only contained negative samples. After labeling eight large clusters of English texts, we already had over 1,000 negative samples for the English training set. In total, 1,427 English and 4,231 German texts from the GDPR-2019 corpus were labeled manually. Table 2 summarizes the training set size for each language. Note that clustering was only used to accelerate the manual labeling process.

Table 2. Number of documents in the English (en) and German (de) training sets by origin (corpora of privacy policies and privacy news articles).

Language	Corpus	Training size	
		Positive	Negative
en	OPP-115	115	0
	APP-350	350	0
	Rogue-Top-100	167	0
	GDPR-2019	424	1003
	Privacy-related news	0	386
de	GDPR-2019	3559	672
	Privacy-related news	0	112

3.5 Feature Engineering

Standard feature (independent variable, attribute, regressor, predictor) engineering for document classification involves extracting token n-grams along with the filtering of, e. g., stop words, punctuation symbols, common words, and hapax and dis legomena. Due to the highly variable and specific language of privacy policies, this approach could lead to extremely high-dimensional vector spaces with sparse matrices. Instead, we performed key phrase extraction to condense each text into a set of key phrases.

Algorithms extracting the most important sentences or key phrases of a text include, in chronological order, KEA [79], TextRank [43], SingleRank [80], Maui [81], KP-Miner [82], WINGNUS [83], RAKE [42], TopicRank [84], TopicalPageRank [85], SGRank [86], PositionRank [87], Seq2seq-keyphrase [88], MultipartiteRank [89], YAKE [90–92], and sCAKE [93]. Another proposed method is the application of Tf-Idf weighting [94].

These algorithms differ in language dependence, applicability on single documents, and the adopted statistical measures, graph algorithms, and (un)supervised learning approach. To ensure that the toolchain can be integrated between a scraper and other analysis components while still being able to process and decide on each

text individually, we defined the following requirements for the key phrase extractors:

1. Applicability on individual documents
2. No supervised training required
3. Domain independence
4. High performance
5. Multiprocessing support

The first and second requirements led to the exclusion of KEA, KPMiner, Maui, WINGNUS, TopicalPageRank, Seq2seq-keyphrase, and Tf-Idf. SGRank was excluded due to slow performance on long texts (> 5 minutes per document). We ensured that the remaining candidates were compatible with the Joblib [95] multiprocessing library for Python. The language dependence of the remaining algorithms can be compensated with a part-of-speech tagger and a list of known stop words for the corresponding language. Although stop lists are readily available, they should be selected with caution [96].

Evaluating the remaining algorithms against our requirements, we ended up with the following key phrase extractors for our toolchain:

1. TextRank
2. SingleRank
3. RAKE
4. TopicRank
5. PositionRank
6. MultiPartiteRank
7. sCAKE
8. YAKE

We added both their original (e. g., YAKE) and existing Python implementations [60, 97] to our toolchain.

3.6 Finding Privacy Policies on Websites

Our work uses existing corpora of privacy policies. While some of them (Rogue-Top-100, APP-350, OPP-115) are based on manually extracted texts, the privacy policies in others (GDPR-2019, Princeton-2020) were collected by identifying hyperlinks presumed to point to the respective policies on websites. For the latter, different approaches can be found in the literature:

1. Simple English: Link texts that contain the word "privacy" or the words "data" and "protection" [16].
2. Two-Step English: Link texts that fully match a predefined list (e. g., "privacy policy," "privacy statement") and, if no match is found, link texts containing a wider set of words (e. g., "security," "statement," "terms"), as used for the Princeton-2020 corpus [30] based on Libert [20].
3. Multilingual: Link texts containing words from multilingual word lists as used for the GDPR-2019 corpus [15].
4. Context: Besides the link texts, the text of the previous HTML element is also considered to identify cases like `"privacy policy <a>here</a>"`. This approach was used by Fabian et al. [8], but no list of words was provided.

We evaluated the performance of each approach on the top 10,000 websites from the Tranco [98] top list as of 31 January 2021 (ID: WQW9) and used the OpenWPM [99] privacy measurement framework to access and parse the websites. For context-based link identification, we used the multilingual word list. We also devised a new approach that analyzed not only the link text but also the URL. We downloaded all identified documents and classified the texts using our trained classifiers described in the next section.

4 Toolchain Performance

In this section, we present the evaluation results of our toolchain and, for each component, identify which solution works best.

4.1 Text Extraction

During the pre-tests of potential toolchain components, we extracted the plain text of scraped web pages using Boilerpipe's [36] default setting. However, manual checks revealed that large parts of the main content had not been extracted from the crawled web pages, leading to incomplete privacy policy texts. We assume the reason to be the non-standardized and greatly varying HTML/XML structures of websites and their privacy policies. More importantly, Boilerpipe was developed to extract text from online news articles whose structure and paragraph density differ from those of privacy policies, rendering the default setting unable to extract all privacy policies completely.

As the correctness and completeness of the extracted texts are crucial for privacy policy content analysis, this prompted us to compare the performance of multiple text-from-HTML extraction libraries. For this, we selected 111 raw HTML files from the ten most common languages in the GDPR-2019 corpus. Our goal was to create a balanced mixture of web pages that reflected shortcomings of the initial plain text extraction attempt and easy-to-extract privacy policies to create a test set that posed a fair challenge to all text-from-HTML extractors. We believe that this sample set provides new

Table 3. Fuzzy string matching scores between manually extracted policies from web pages and each tested text-from-HTML extraction library. The web pages were sampled from 10 European languages. The best scores are marked in bold.

	Language			
Library	**en**	**de**	**other**	**overall**
Boilerpipe				
ArticleExtractor	81.8 ± 28.5	70.4 ± 29.9	87.1 ± 18.9	85.3 ± 21.0
ArticleSentencesExtractor	79.9 ± 29.3	67.6 ± 30.3	83.8 ± 18.6	82.2 ± 20.9
CanolaExtractor	95.6 ± 3.5	86.8 ± 15.0	86.7 ± 16.7	87.4 ± 16.1
DefaultExtractor	95.0 ± 5.9	84.7 ± 14.5	85.3 ± 19.4	86.0 ± 18.5
KeepEverythingExtractor	82.8 ± 21.8	85.4 ± 14.9	78.5 ± 22.4	79.3 ± 21.8
LargestContentExtractor	81.0 ± 20.2	61.3 ± 34.4	74.3 ± 22.3	73.7 ± 23.4
NumWordsRulesExtractor	**97.5 ± 1.9**	**87.7 ± 15.0**	89.3 ± 16.4	**89.8 ± 15.8**
HTML2Text	67.6 ± 32.2	72.2 ± 19.7	62.4 ± 28.4	63.6 ± 28.0
Inscriptis	82.9 ± 20.7	75.8 ± 26.7	78.9 ± 21.1	78.9 ± 21.4
Readability.js	96.3 ± 5.7	64.6 ± 39.2	**91.1 ± 16.4**	89.4 ± 20.0
Goose3	82.1 ± 33.7	62.4 ± 32.6	51.7 ± 39.7	54.7 ± 39.4
Dragnet	89.3 ± 29.2	62.7 ± 38.5	80.3 ± 31.3	79.5 ± 31.9
BeautifulSoup	59.6 ± 30.7	55.9 ± 21.2	49.1 ± 28.8	50.5 ± 28.3

and nuanced insights into text-from-HTML extractors as considering only popular websites might introduce a bias towards well-maintained websites.

We manually extracted the plain text of these 111 policies and removed extra white spaces. Converting non-breaking spaces in Latin-1/ISO 8859-1 to normal spaces is a preprocessing step that improves text analysis quality and does not negatively influence the results.

Table 3 demonstrates the average fuzzy string matching scores between the manually extracted texts and the output of each text-from-HTML extractor per language. We calculated the scores with the Python fuzzy string matching library [100], which uses the Levenshtein distance to calculate the similarity between two strings. It turned out that the Boilerpipe library with the NumWordsRules and CanolaExtractor settings, as well as Readability.js yield the highest similarity scores with the manually extracted privacy policies. With the exception of Readability.js for German, a lower two-digit standard deviation can be observed for all three top-performing text-from-HTML extractors in both languages.

For German, Readability.js was unable to extract large portions of text from two websites whose privacy policy pages hid privacy policy content under CSS accordion elements. This phenomenon requires more profound analysis by front-end development experts. We repeated the text extraction for these two websites using the version of Readability.js included in the latest version of Firefox as of February 2021 (ver. 85.0.2). Unfortunately, this problem still occurred.

Following these findings, we chose Boilerpipe with the NumWordsRules setting as the default text-from-HTML extractor for our toolchain.

The analysis in the next section supports the selection of this extractor and demonstrates that a fallback mechanism to the text extractors with the second and third highest performance should be considered to also achieve the best possible results for certain edge cases.

4.2 Language Detection

Since language detection libraries are often sensitive to noise such as URLs, emails, or phone numbers, in this step we removed these elements from all texts using regular expressions and only kept alphanumeric tokens. Texts that did not contain at least ten space-separated tokens were filtered out as they were too short to produce a reliable outcome. We ensured that all language detection libraries output the same ISO 639-1 codes for the languages, e. g., "zh-cn" instead of "zh" for Chinese, and, whenever possible, the string "un" in case the language could not be detected instead of falling back to English or outputting other values like "unknown."

For each text, the statistical mode of the list of detected languages obtained from all libraries determined the language of the texts. If no mode could be determined, the text was marked to have no super seeding language to indicate that it required further manual inspection. These checks revealed that such texts usually consist of a single web page that contains the respective privacy policy in multiple languages, as shown in Figure 2. Among the eight used libraries, CLD2 and CLD3 can determine the languages of multilingual texts in the form of span indices and the ratios of the detected languages in a text, respectively. A text was flagged as multilingual, independently of the majority voting, if either CLD2 or CLD3 detected the text as multilingual.

This way, we identified 24,257 multilingual and 95,880 monolingual texts in the GDPR-2019 corpus. 6,973 texts were discarded for containing less than ten filtered tokens. In 218 cases, the language could not be determined via majority vote, for 181 not even by CLD2. It is not feasible to determine the language of these texts by hand as the variety of languages detected by the other six libraries is too large and would require consulting language experts.

As for monolingual texts, all libraries detected the same language for 88,976 texts, which equals 74.1 % of the GDPR-2019 data set excluding too short texts and those in an undetermined language. For the remaining 7,068 monolingual texts, the language was determined via majority voting because at least one library had output a different language than the others.

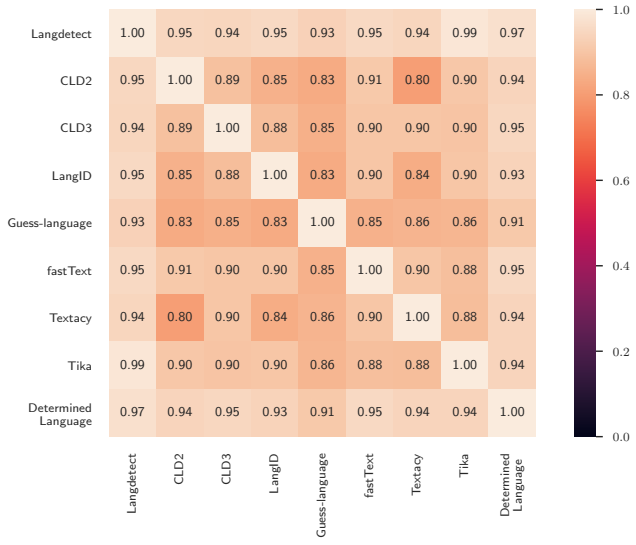

Fig. 3. Cramér's V values between the outputs of the language detection libraries and the final determined language. All values are statistically significant ($p < .0001$).

The 24,257 texts flagged as multilingual required special handling, especially the 5,366 cases with a uniform vote on the text's language. The language of 2,157 of these texts was detected as Modern Greek, 1,434 as Bulgarian, and 1,410 as Russian, which sums up to 5,001 texts. As for the texts flagged as monolingual, 36 texts were detected as Modern Greek, 255 as Bulgarian, and 189 as Russian. Although these findings must be interpreted with caution, we think that it might be necessary to preprocess privacy policies in these languages differently or to use more specialized language detection libraries. This is because the log files of CLD2 and CLD3 often indicated text spans for which the language could either not be determined or was identified as, e. g., Latin, Kyrgyz, or Western Frisian with a ratio of less than 1 % per text. For the remaining 18,891 texts without a uniform language voting and flagged as multilingual by CLD2 and CLD3, their high number made a complete manual analysis infeasible, so we focused on a subset of 1,808 texts with a majority vote for English. Manual analysis revealed that these texts could be grouped as follows:

1. Multilingual privacy policies.
2. Multilingual error messages.
3. English texts or privacy policies whose text body contained named entities that had been incorrectly identified as non-English text.
4. English texts with superfluous non-English text spans at the beginning or the end.

For the last group, manual analysis revealed that either Readability.js or Boilerpipe with the CanolaExtractor setting had managed to strip off the superfluous text spans. Therefore, implementing a fallback solution to perform text-from-HTML extraction with these two libraries followed by a recheck for multilingualism appears to yield the best result. As described in the previous section, CLD2 can output the indices of text spans of each detected language segment, while CLD3 outputs the text ratio of each detected language. If a text did not reach a uniform vote regarding its language and was flagged as multilingual, a heuristic that checks for short text spans at the beginning or the end of the text in a language different from the main language could solve the issue.

To improve performance, we evaluated whether it is possible to exclude any of the language detection libraries. Figure 3 shows the Cramér's V correlation [101] among the outputs of the language detection libraries and the final result. The Tika and Langdetect libraries have the highest correlation value of 0.99, so one of them can be dropped ($\chi^2(1872,\ N = 127,326) = 4,832,799.8,\ p < .0001$). We do not recommend removing any other library as high precision is one of the objectives of this toolchain.

As a means of comparison, we applied our language detection scheme to the Princeton-2020 corpus, in which all non-English texts had been filtered with Polyglot [102], a natural language processing library that uses CLD2 for language detection. While our voting scheme flagged most privacy policies and non-privacy policies in this corpus as monolingual English texts, some texts were detected to be either in languages other than English or multilingual, as shown in Table 4.

Table 4. Language statistics of our language voting mechanism for the Princeton-2020 corpus.

	Privacy Policy		Non-Privacy Policy	
Language	**Monolingual**	**Multilingual**	**Monolingual**	**Multilingual**
English	837,796	4,070	153,131	1,271
German	0	33	26	50
French	1	26	6	36
Indonesian	0	21	36	42
Other	39		241	
Too short	723		14,328	
No majority	19		66	

We manually reviewed the privacy policies and non-privacy policies detected to be German and found that they either consisted of pure German text with English

headings or a mix of English and German privacy policies. As expected, we also found some error messages. Our evaluation of the Princeton-2020 corpus illustrates how our language detection scheme can contribute to improve the quality of existing corpora.

5 Privacy Policy Detection

This section describes the evaluation of the toolchain's last two components – extracting useful features and training a classification scheme.

5.1 Feature Determination

Previous research has used simple tokenization and n-gram extraction combined with Tf-Idf weighting to build feature matrices (cf. Section 2). With our goal to save resources, we focus on finding the most efficient way, with the lowest possible number of features, to distinguish between privacy/cookie policies and other texts.

We performed pre-tests with the key phrase extraction algorithms described in Section 3.5. Among all utilized libraries, YAKE has the highest default number of 20 extracted key phrases per text, while the others only extract 10 by default. We set the maximum number of extracted key phrases from each text for all algorithms to 20 and lemmatized all documents to prevent repetitions of nearly identical key phrases as features. As we are not aware of any publicly available data set for evaluating the correctness and ideal number of key phrases extracted from privacy policies, we conducted an extensive analysis comprising manual inspection of the key phrases extracted by each algorithm, feature selection using the ANOVA F-value, and plotting the resulting training set for each key phrase extraction algorithm. Due to space constraints, we cannot provide results in full detail. The choice of the key phrase extraction algorithm remained nondistinctive in addition to the diversity in the wording of privacy policies, supported by previous work [7, 10]. We thus combined the extracted key phrases of each text into a set of key phrases, which results in a feature matrix with each row consisting of zeros and ones representing the presence or absence of a key phrase in the corresponding document. The decision for a binary (dichotomous) feature matrix takes into account that the absence of a key phrase in a set of extracted key phrases does not necessarily mean that it does not exist in the corresponding text but was just not selected by any of the algorithms. Therefore, weighting the extracted key phrases by the frequency of their occurrence is unreasonable, while keeping them in a binary representation is the conservative method to choose.

Not all extracted key phrases might clearly correlate with occurrence in a privacy/cookie policy or non-privacy policy, so we applied ANOVA F-value feature selection to choose the most relevant group of features. We set $p = .05$ and applied family-wise error correction due to the high number of significance tests. With this method, 570 out of 156,020 features were selected for the English training set, while 1,820 out of 57,254 features were selected for the German training set. The χ^2 test measure led to a nearly identical selection of features. Only three and four features selected with ANOVA would not have been selected using χ^2 for the English and German feature sets, respectively.

As the GDPR-2019 data set captures privacy policy changes around the GDPR enforcement date, it includes more variety in the wordings, while the English corpora were collected before the enforcement of the GDPR. The positive training samples for German result from manually labeling the GDPR-2019 corpus (see Section 2). This variety can also be observed in the t-distributed Stochastic Neighbor Embedding (t-SNE) projection in Figure 4. Compared to the English privacy policies, the German privacy policies are more widely distributed.

The extracted key phrases in both languages show common themes, including data practices such as the collection or processing of data, tracking technologies such as "web beacon" and "flash cookie," common phrases in error messages, and the names of regulations, particularly references to the GDPR in the German set. Key phrases in other languages can also be observed because the training set should include both monolingual and multilingual texts so that the resulting classifiers handle multilingual texts correctly.

5.2 Policy Detection

Considering the heterogeneity of the training set in Figure 4, we opted for an ensemble soft voting classifier [103], which predicted the label with the maximum average prediction probability of the applied classification models. Our voting classifier comprised three different classification models well established for text classification tasks [104, 105] – a linear support vector machine, random forest [106], and logistic regression [107].

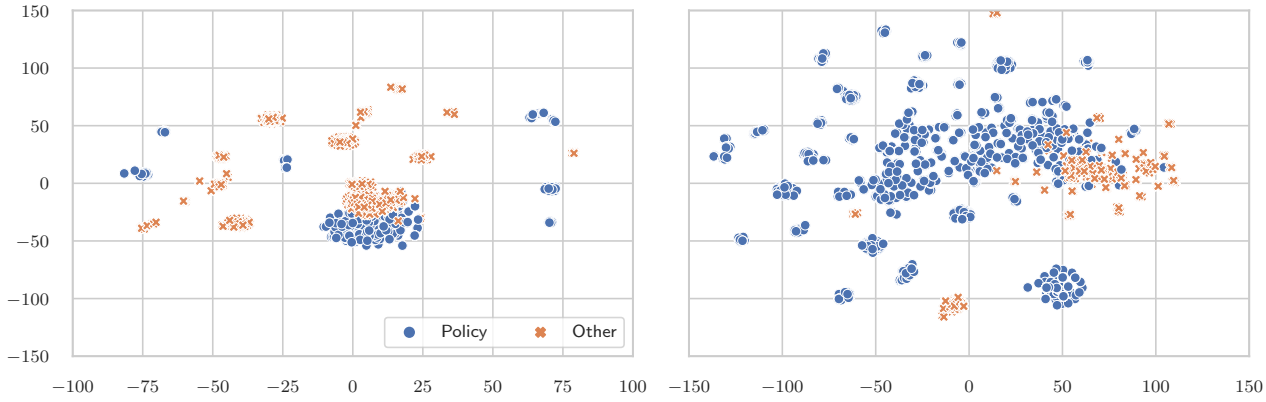

Fig. 4. Two-dimensional t-SNE projection of the English (left) and German training sets after truncated truncated singular value decomposition ($n = 50$).

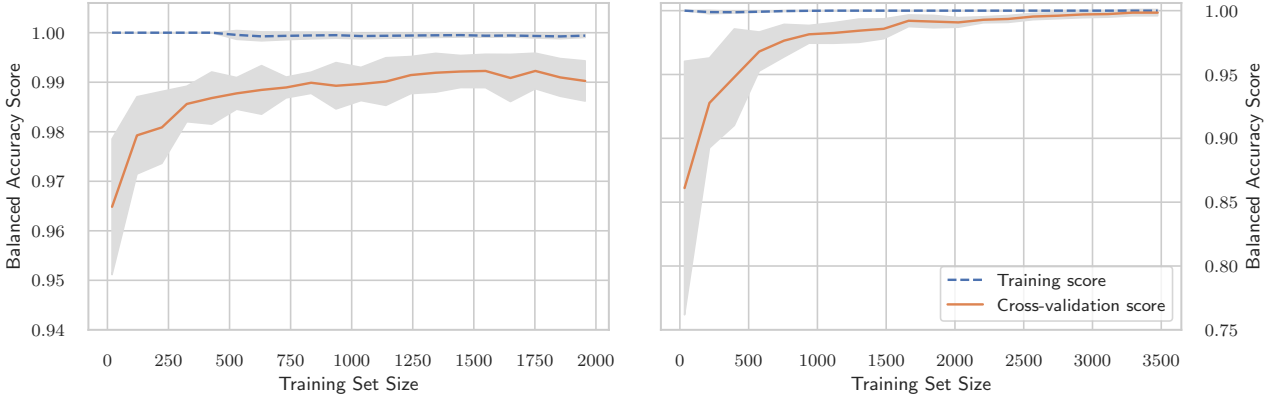

Fig. 5. Learning curves of the trained classifiers. Left: voting classifier for English texts. Right: voting classifier for German texts. The gray margin illustrates the standard deviation of the mean balanced accuracy for 5-fold cross validation.

The random forest was trained with 500 decision trees. The class for each text was determined as follows:

$$prediction = \arg\max_i \sum_{j=1}^{m} p_{ij},$$

where p_{ij} is the predicted probability of each classifier j for each class $i \in \{0, 1\}$. No extra weights were assigned to the classifiers, and all were calibrated to output precise prediction probabilities [108]. We evaluated our models' performance using stratified 5-fold cross-validation on the labeled texts. Stratification was applied because our training sets are not balanced, so the percentage of samples is preserved for each of the two classes in each fold. Figure 5 displays the learning curve for the resulting classifiers for each of the English and German texts. It can be observed that the voting classifier achieves excellent results by combining the strengths of its underlying calibrated classifiers. We measure its performance using the balanced accuracy score to prevent artificial performance estimates for our imbalanced training sets, and consider both sensitivity and specificity in our performance evaluation. The balanced accuracy score is calculated as the average of sensitivity and specificity as follows:

$$score = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

with TP meaning true positive, TN true negative, FN false negative, and FP false positive. Overall, the respective English and German classifiers achieve balanced accuracy scores of 99.1 % and 99.6 %, F1 scores of 99.1 % and 99.8 %, and precision scores of 99.2 % and 99.8 % in 5-fold cross-validation. The best performing classifier in the literature [24] reported an F1 score of 99.2 % and a precision of 99.0 % for English privacy policies. Given the use case of correctly identifying privacy/cookie policies, we achieve an improvement of 0.2

percentage points and provide the first classifier for German-language privacy policies.

We used the voting classifier to classify the unlabeled English and German texts in the GDPR-2019 corpus. We partially reviewed the labels assigned to the unlabeled policies and could not find any significant errors except for a few edge cases, like mixtures of terms of service and data practices, cases where surrounding text of short privacy policies had been extracted, or cookie banners, which are out of scope of this work. All three classifiers output the same label for 92.5 % and 91.6 % of the unlabeled English and German texts, respectively.

As there is no gold standard for benchmarking privacy policy classifiers, we compared our trained classifier's predictions with the assigned labels in the Princeton-2020 corpus, which includes both privacy policies and non-privacy policies. The authors of this corpus had labeled each text with a trained random forest classifier using word n-grams with $n = [1, 4]$ and other features extracted from the title of the texts.

First, we applied our voting classifier to the 109 texts in this corpus we identified to be in German (cf. Table 4). 32 out of the 33 texts labeled as privacy policies were also detected to be privacy policies. The single exception turned out to be a short multilingual privacy policy in both English and German. For texts labeled as non-privacy policies, we observed that 57 out of 76 were classified as privacy policies by our classifier. Manual review confirmed the labeling of these 57 cases, with one case only having a 60 % predicted probability of being a privacy policy. We checked the two mentioned cases using SHAP [109] to better understand them. The reason turned out to be the absence of specific key phrases in these short policies related to "cookies," "usage," "personal data," and "processing" while they included key phrases related to "data storage" and "purpose." The remaining texts were not privacy policies but error messages or advertisement text correctly classified as "other." For English texts, our classifier had a 96.7 % agreement with the labels of the 841,866 English privacy policies and a 41.7 % agreement with labels of the 154,402 English non-privacy policies. Manual spot checks revealed that our classifier was able to identify privacy policies that had been incorrectly labeled as non-privacy policies in the Princeton corpus. We can therefore confirm the approach of Amos et al. [30] that mainly focused on increasing the precision of their classifier, while the focus of our classifiers is to increase the balanced accuracy. Therefore, we are able to significantly reduce the number of false negatives.

5.3 Finding Privacy Policies on Websites

As described in Section 3.6, we compared different approaches to identify privacy policies on 10,000 websites from the Tranco list. Using OpenWPM, we were able to access 8,624 websites and identified documents matching any search pattern on 7,353 (85 %) of them. Overall, 26,433 of the 33,246 downloaded documents were classified as either English or German. Of the 23,848 English documents 13,475 (57 %) and 1,564 (60 %) of the 2,587 German documents were classified as privacy policies.

Table 5. Overview of different approaches to search for privacy policies on 10,000 websites. Results are reported relative to the total number of reachable websites and in absolute numbers.

	Site Statistics			Absolute	
Approach	**Doc.**	**Avg.**	**at least 1 PP**	**Doc.**	**TP**
Simple	4,938 (57 %)	1.41	4,185 (49 %)	6,942	5,200 (75 %)
2-Step	5,784 (67 %)	1.59	4,554 (53 %)	9,200	6,000 (65 %)
Multilingual	6,712 (78 %)	2.78	4,919 (57 %)	18,726	10,513 (56 %)
Context	6,782 (79 %)	3.44	4,944 (57 %)	24,082	10,630 (56 %)
Link+URL	6,989 (81 %)	2.81	4,999 (58 %)	19,062	11,881 (49 %)
Combination	7,353 (85 %)	3,91	5,173 (60 %)	28,747	12,639 (44 %)

Table 5 presents an overview of the results. As expected, the more extensive the rule set, the higher the number of found links and privacy policy candidates. For the simple methods (links that contain the word "privacy" or the words "data" and "protection"), policy candidates were found on 57 % of the reachable websites, with 75 % (and 49 % of all websites) being actual (true positive) privacy policies according to our classifier. The absolute numbers highlight that the number of false positives, where a link was identified but the downloaded text is not a privacy policy, increases with more extensive search lists.

No single approach could identify all 7,353 URLs retrieved with a combination of the Simple, 2-Step, Context, and Link+URL methods, with the latter using a multilingual set of words. With this combination, privacy policy candidates were identified on 85 % of the websites, and for 60 % the classifier confirmed that at least one text contained information about privacy practices.

6 Discussion

In this section, we discuss the findings of our experiments for each component of the toolchain. We also

point out limitations of our approach and provide suggestions how to possibly improve our toolchain and, consequently, the preprocessing of privacy and cookie policies.

6.1 Experimental Results

The results show that the preprocessing phase is crucial for producing valid and comparable results in privacy policy analysis. Our findings reveal that choosing improper components for text extraction, language detection, and classification leads to incompletely extracted privacy policies, incorrectly detected languages, and false positives and false negatives in the data set. We carefully selected and evaluated candidate tools for each of these components in our toolchain and determined the best existing solution. Although there already is extensive research on the content of privacy policies, we believe that such a well-tested toolchain is a crucial prerequisite for the validity and quality of the results of privacy policy research.

As demonstrated in this work, if the text-from-HTML extractor is not carefully selected, either substantial portions of privacy policies are not extracted or superfluous text is included in the output. For this component, we recommend the Boilerpipe library with the *NumWordsRules* extraction setting. In the case of privacy policies whose beginning and end contain small amounts of text in a language other than that of the main text, a solution could be to fall back to Boilerpipe with the *CanolaExtractor* setting or Readability.js.

Inaccuracies in the language identification process lead to an incorrect classification of policies regarding their language(s), which affects subsequent analyses such as text classification, text mining, or question answering. For instance, filtering out stop words, a common step in text processing, cannot be accurately performed. A stop word list that does not match the text's actual language causes key phrase extraction algorithms such as RAKE to perform worse because stop words (and punctuation symbols) are used to partition a text into candidate key phrases. Our proposed solution is a majority voting scheme consisting of eight language detection libraries that has the ability to detect individual language segments in multilingual texts. This voting mechanism has the benefit of self-correction and prevents loss of information if one of its underlying language detection libraries is unable to handle specific edge cases. This solution was able to identify non-English texts in the Princeton-2020 corpus, for which the Polyglot library had been used for language detection. While the number of errors regarding detected languages is relatively small in this corpus, our majority voting scheme provides an easy-to-apply solution to improve the data quality of a privacy policy corpus.

Our feature engineering scheme leverages the power of eight selected unsupervised key phrase extraction libraries. In combination with ANOVA F-value feature selection, the trained and calibrated ensemble voting classifier shows confident handling of the decision whether a text is a privacy/cookie policy or not. Although the selected features contain names of international corporations such as Google or Microsoft, filtering out organizations' names is non-trivial. Still, we believe that keeping these names does not lead to issues such as overfitting. Compared to the classifier applied in the creation of the Princeton-2020 corpus, our classifier shows much higher correctness in distinguishing privacy policy texts from other texts. Out of the 76 texts initially classified as non-privacy policy texts in German, we were able to identify 56 as privacy policies. This contribution prevents the loss of valuable information for this corpus and future research.

6.2 Limitations & Future Work

As the introduced toolchain is implemented in the Python programming language, it cannot include libraries of other programming languages that do not (yet) support Python. We consider the analysis of libraries in other programming languages important future work to further improve our toolchain.

We found many privacy policies to contain HTML tables with additional information about the website's privacy practices, e. g., third-party libraries used by the website or the purposes for which different types of data are being collected. The extraction process may lead to table headers and content being disordered in the plain text output. This is an example of how the lack of legal requirements for the format of privacy policies can lead to policy structures that are hard to automatically extract without massive data loss. Correctly handling these differences in structure and format during text extraction would require closer investigation and specialized well-tested heuristics.

Future work could also investigate possible requirements for retraining and adapting classifier models when new privacy regulations are passed to keep the models accurate and up to date.

All open-source libraries and tools have high value for the research community. Therefore, we would like to emphasize at this point that each of the tested tools we found unable to produce the required results might have been performing well in other specific use cases and us excluding them does not mean they are weak tools per se. However, their performance had not yet been evaluated in the specific domain of this research, i. e., privacy policy analysis, and this evaluation is one of the main contributions of this research.

7 Conclusion

Previous work studying the content of privacy policies usually makes use of tailor-made processes and tools to obtain and prepare data. Due to a wide variety in the structure and implementation of these policies on websites, the resulting analyses lack common ground, making it difficult to compare the results of different studies. We introduced a uniform process and a best-practice toolchain to mitigate these shortcomings and to harmonize future research. We addressed how to extract and prepare relevant information from a corpus of privacy policies, as well as the core task of detecting whether or not a given text is indeed a privacy or cookie policy and not some other type of text. The correct operation of our toolchain has been evaluated utilizing state-of-the-art privacy policy corpora, and we have given insights into data handling and preparation for this type of analysis. Our findings empower future work, providing methods for thorough data handling accessible to fellow researchers in the field of privacy policy content analysis.

Code Availability

To foster our aim of unifying privacy policy analysis and making future work comparable, we publish the developed toolchain and its code with this paper. The code of the toolchain is available at:

https://github.com/ITSec-WWU-Munster/Unifying-Privacy-Policy-Detection

8 Acknowledgments

The authors would like to thank their shepherd, Jessica Staddon, and the anonymous reviewers for their helpful comments. This research was funded by the state of North-Rhine Westphalia (MKW-NRW) through the Research Training Groups SecHuman and NERD.NRW.

References

[1] Kenneth D. Pimple. *Emerging Pervasive Information and Communication Technologies (PICT)*. Springer, 2014.

[2] Willis H. Ware. Records, Computers and the Rights of Citizens. Technical report, The Rand Corporation, Santa Monica, California, 1973.

[3] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. (Un)informed Consent: Studying GDPR Consent Notices in the Field. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 973–990, 2019.

[4] Julie M. Robillard, Tanya L. Feng, Arlo B. Sporn, Jen-Ai Lai, Cody Lo, Monica Ta, and Roland Nadler. Availability, readability, and content of privacy policies and terms of agreements of mental health apps. *Internet Interventions*, 17:100243, 2019.

[5] Noriko Tomuro, Steven Lytinen, and Kurt Hornsburg. Automatic Summarization of Privacy Policies using Ensemble Learning. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pages 133–135, 2016.

[6] Razieh Nokhbeh Zaeem, Rachel L. German, and K. Suzanne Barber. PrivacyCheck: Automatic Summarization of Privacy Policies Using Data Mining. *ACM Transactions on Internet Technology (TOIT)*, 18(4):1–18, 2018.

[7] Dhiren A. Audich, Rozita Dara, and Blair Nonnecke. Extracting keyword and keyphrase from online privacy policies. In *2016 Eleventh International Conference on Digital Information Management (ICDIM)*, pages 127–132. IEEE, 2016.

[8] Benjamin Fabian, Tatiana Ermakova, and Tino Lentz. Large-Scale Readability Analysis of Privacy Policies. In *Proceedings of the International Conference on Web Intelligence*, pages 18–25, 2017.

[9] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, et al. The Creation and Analysis of a Website Privacy Policy Corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340, 2016.

[10] Dhiren A. Audich, Rozita Dara, and Blair Nonnecke. Privacy Policy Annotation for Semi-automated Analysis: A Cost-Effective Approach. In *IFIP International Conference on Trust Management*, pages 29–44. Springer, 2018.

[11] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning. In *Proceedings of the 27th USENIX Security Symposium*, pages 531–548, 2018.

[12] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. "Your Hashed IP Address: Ubuntu." Perspectives on Transparency Tools for Online Advertising. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 702–717, 2019.

[13] Luca Bufalieri, Massimo La Morgia, Alessandro Mei, and Julinda Stefa. GDPR: When the Right to Access Personal Data Becomes a Threat. *arXiv preprint arXiv:2005.01868*, 2020.

[14] Coline Boniface, Imane Fouad, Nataliia Bielova, Cédric Lauradoux, and Cristiana Santos. Security Analysis of Subject Access Request Procedures. In *Annual Privacy Forum*, pages 182–209. Springer, 2019.

[15] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We Value Your Privacy ... Now Take Some Cookies: Measuring the GDPR's Impact on Web Privacy. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS 2019)*. The Internet Society, February 2019.

[16] Mukund Srinath, Shomir Wilson, and C. Lee Giles. Privacy at Scale: Introducing the PrivaSeer Corpus of Web Privacy Policies. *arXiv preprint arXiv:2004.11131*, 2020.

[17] Fei Liu, Rohan Ramanath, Norman Sadeh, and Noah A. Smith. A Step Towards Usable Privacy Policy: Automatic Alignment of Privacy Statements. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 884–894, 2014.

[18] Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. Can We Trust the Privacy Policies of Android Apps? In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 538–549. IEEE, 2016.

[19] Abhijith Athreya Mysore Gopinath, Shomir Wilson, and Norman Sadeh. Supervised and Unsupervised Methods for Robust Separation of Section Titles and Prose Text in Web Documents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 850–855. Association for Computational Linguistics, 2018.

[20] Timothy Libert. An Automated Approach to Auditing Disclosure of Third-Party Data Collection in Website Privacy Policies. In *Proceedings of the 2018 World Wide Web Conference*, pages 207–216, 2018.

[21] Keishiro Fukushima, Toru Nakamura, Daisuke Ikeda, and Shinsaku Kiyomoto. Challenges in Classifying Privacy Policies by Machine Learning with Word-based Features. In *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy (ICCSP 2018)*, pages 62–66, Guiyang, China, 2018. ACM.

[22] Tarun Ramadorai, Antoine Uettwiller, and Ansgar Walther. The Market for Data Privacy. https://dx.doi.org/10.2139/ssrn.3352175, 2019.

[23] Martin Boldt and Kaavya Rekanar. Analysis and Text Classification of Privacy Policies From Rogue and Top-100 Fortune Global Companies. *International Journal of Information Security and Privacy (IJISP)*, 13(2):47–66, 2019.

[24] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N. Cameron Russell, and Norman Sadeh. MAPS: Scaling Privacy Compliance Analysis to a Million Apps. *Proceedings on Privacy Enhancing Technologies*, 2019(3):66–86, 2019.

[25] David Sarne, Jonathan Schler, Alon Singer, Ayelet Sela, and Ittai Bar Siman Tov. Unsupervised Topic Extraction from Privacy Policies. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 563–568. IW3C2 (International World Wide Web Conference Committee), 2019.

[26] Mitra Bokaie Hosseini, KC Pragyan, Irwin Reyes, and Serge Egelman. Identifying and Classifying Third-party Entities in Natural Language Privacy Policies. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 18–27, 2020.

[27] Vinayshekhar Bannihatti Kumar, Roger Iyengar, Namita Nisal, Yuanyuan Feng, Hana Habib, Peter Story, Sushain Cherivirala, Margaret Hagan, Lorrie Cranor, Shomir Wilson, et al. Finding a Choice in a Haystack: Automatic Extraction of Opt-Out Statements from Privacy Policy Text. In *Proceedings of The Web Conference 2020*, 2020.

[28] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. The Privacy Policy Landscape After the GDPR. *Proceedings on Privacy Enhancing Technologies*, 2020(1):47–64, 2020.

[29] Yoon Kim. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics, 2014.

[30] Ryan Amos, Gunes Acar, Elena Lucherini, Mihir Kshirsagar, Arvind Narayanan, and Jonathan Mayer. Privacy Policies over Time: Curation and Analysis of a Million-Document Dataset. *arXiv preprint, arXiv:2008.09159*, 2020.

[31] Leonard Richardson. Beautiful Soup. https://www.crummy.com/software/BeautifulSoup/bs4/doc/, 2007. [Online; accessed 24 April 2020].

[32] Postlight. Mercury Parser – Extracting content from chaos. https://github.com/postlight/mercury-parser.

[33] Stefan Behnel, Martijn Faassen, and Ian Bicking. lxml: Processing XML and HTML with Python. https://lxml.de/, 2005. [Online; accessed 14 June 2021].

[34] Kanthashree Mysore Sathyendra, Abhilasha Ravichander, Peter Garth Story, Alan W. Black, and Norman Sadeh. Helping Users Understand Privacy Notices with Automated Query Answering Functionality: An Exploratory Study. Technical report, 2017.

[35] Marco Lui and Timothy Baldwin. langid.py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30. Association for Computational Linguistics, 2012.

[36] Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. Boilerplate Detection using Shallow Text Features. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 441–450, 2010.

[37] Rohan Ramanath, Fei Liu, Norman Sadeh, and Noah A. Smith. Unsupervised Alignment of Privacy Policies using Hidden Markov Models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*

(Volume 2: Short Papers), pages 605–610, 2014.

[38] Jim Plush and Robbie Coleman. Goose - Article Extractor. https://github.com/goose3/goose3, 2011. [Online; accessed 24 April 2020].

[39] Nakatani Shuyo. Language Detection Library for Java. http://code.google.com/p/language-detection/, 2010.

[40] Welderufael B. Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel 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*, pages 15–21, 2018.

[41] Matthew E. Peters and Dan Lecocq. Content Extraction Using Diverse Feature Sets. In *Companion Publication of the 22nd International World Wide Web Conference*, pages 89–90, 2013.

[42] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. Automatic Keyword Extraction from Individual Documents. *Text Mining: Applications and Theory*, 1:1–20, 2010.

[43] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, 2004.

[44] Jonathan Hedley. jsoup: Java HTML Parser. https://jsoup.org, 2009.

[45] Elisa Costante, Yuanhao Sun, Milan Petković, and Jerry den Hartog. A Machine Learning Solution to Assess Privacy Policy Completeness. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, pages 91–96. ACM, 2012.

[46] Niharika Guntamukkala, Rozita Dara, and Gary Grewal. A Machine-Learning Based Approach for Measuring the Completeness of Online Privacy Policies. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pages 289–294. IEEE, 2015.

[47] Shuang Liu, Renjie Guo, Baiyang Zhao, Tao Chen, and Meishan Zhang. APPCorp: A Corpus for Android Privacy Policy Document Structure Analysis. *arXiv preprint arXiv:2005.06945*, 2020.

[48] Cheng Chang, Huaxin Li, Yichi Zhang, Suguo Du, Hui Cao, and Haojin Zhu. Automated and Personalized Privacy Policy Extraction Under GDPR Consideration. In *International Conference on Wireless Algorithms, Systems, and Applications*, pages 43–54. Springer, 2019.

[49] Parvaneh Shayegh, Vijayanta Jain, Amin Rabinia, and Sepideh Ghanavati. Automated Approach to Improve IoT Privacy Policies. *arXiv preprint arXiv:1910.04133*, 2019.

[50] Statista. Percentage of mobile device website traffic worldwide from 1st quarter 2015 to 1st quarter 2021. https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices/. [Online; accessed 14 June 2021].

[51] Pradeep K. Murukannaiah, Chinmaya Dabral, Karthik Sheshadri, Esha Sharma, and Jessica Staddon. Learning a Privacy Incidents Database. In *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, pages 35–44, 2017.

[52] Aaron Swartz and Alireza Savand. HTML2Text. https://alir3z4.github.io/html2text/, 2011. [Online; accessed 20 April 2020].

[53] Albert Weichselbraun and Fabian Odoni. inscriptis – HTML to text conversion library, command line client and Web service. https://inscriptis.readthedocs.io/en/latest/, 2016. [Online; accessed 20 April 2020].

[54] Mozilla. Readability.js. https://github.com/mozilla/readability, 2015. [Online; accessed 24 April 2020].

[55] Jorj X. McKie and Ruikai Liu. PyMuPDF. https://github.com/pymupdf/PyMuPDF, 2016. [Online; accessed 7 January 2021].

[56] The Apache Software Foundation. Apache Tika – a content analysis toolkit. https://tika.apache.org/, 2019. Online; accessed 15 June 2021.

[57] Dick Sites. Compact Language Detector 2. https://github.com/CLD2Owners/cld2, 2013. Online; accessed 15 June 2021.

[58] Alex Salcianu, Andy Golding, Anton Bakalov, Chris Alberti, Daniel Andor, David Weiss, Emily Pitler, Greg Coppola, Jason Riesa, Kuzman Ganchev, et al. Compact Language Detector v3. https://github.com/google/cld3, 2018.

[59] Kent Johnson and Phi-Long Do. Goose – Article Extractor. https://bitbucket.org/spirit/guess_language/, 2008. [Online; accessed 24 April 2020].

[60] Burton DeWilde. textacy: NLP, before and after spaCy. https://github.com/chartbeat-labs/textacy, 2016. [Online; accessed 24 April 2020].

[61] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. FastText.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[62] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759*, 2016.

[63] Trang Ho and Allan Simon. Tatoeba: Collection of sentences and translations. https://tatoeba.org, 2016. [Online; accessed 15 June 2020].

[64] Jörg Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC '12)*, Istanbul, Turkey, May 2012. European Language Resources Association (ELRA).

[65] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC '12)*.

[66] Liling Tan, Marcos Zampieri, Nikola Ljubešic, and Jörg Tiedemann. Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 11–15, Reykjavik, Iceland, 2014.

[67] Mitja Trampus. Evaluating language identification performance. https://blog.twitter.com/engineering/en_us/a/2015/evaluating-language-identification-performance, 2015. [Online; accessed 15 April 2021].

[68] Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. The JRC-Acquis: A Multilingual Aligned Parallel Corpus with 20+ Languages. *arXiv preprint cs/0609058*, 2006.

[69] Jamie Callan, Mark Hoy, Changkuk Yoo, and Le Zhao. The ClueWeb09 Dataset. http://boston.lti.cs.cmu.edu/Data/clueweb09, 2009. [Online; accessed 14 June 2021].

[70] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. RCV1: A New Benchmark Collection for Text Categorization Research. *The Journal of Machine Learning Research*, 5:361–397, 2004.

[71] Tomohiro Kubota. Introduction to i18n. https://www.debian.org/doc/manuals/intro-i18n/, 2003. Online; accessed 24 April 2021.

[72] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML'14)*, pages II–1188–II–1196, 2014.

[73] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. *arXiv preprint arXiv:2003.07082*, 2020.

[74] Katrin Ortmann, Adam Roussel, and Stefanie Dipper. Evaluating Off-the-Shelf NLP Tools for German. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 212–222, 2019.

[75] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. https://sentometrics-research.com/publication/72/. [To appear].

[76] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[77] Helena Gómez-Adorno, Juan-Pablo Posadas-Durán, Grigori Sidorov, and David Pinto. Document embeddings learned on various types of n-grams for cross-topic authorship attribution. *Computing*, 100(7):741–756, 2018.

[78] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[79] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. KEA: Practical Automatic Keyphrase Extraction. *arXiv preprint arXiv:cs/9902007*, 1999.

[80] Xiaojun Wan and Jianguo Xiao. CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969–976, 2008.

[81] Olena Medelyan, Eibe Frank, and Ian H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP '09)*, pages 1318–1327, 2009.

[82] Samhaa R. El-Beltagy and Ahmed Rafea. KP-Miner: Participation in SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 190–193. Association for Computational Linguistics, 2010.

[83] Thuy Dung Nguyen and Minh-Thang Luong. WINGNUS: Keyphrase Extraction Utilizing Document Logical Structure. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 166–169. Association for Computational Linguistics, 2010.

[84] Adrien Bougouin, Florian Boudin, and Béatrice Daille. TopicRank: Graph-based Topic Ranking for Keyphrase Extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551. Asian Federation of Natural Language Processing, 2013.

[85] Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. Topical Word Importance for Fast Keyphrase Extraction. In *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion)*, pages 121–122, 2015.

[86] Soheil Danesh, Tamara Sumner, and James H. Martin. SGRank: Combining Statistical and Graphical Methods to Improve the State of the Art in Unsupervised Keyphrase Extraction. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 117–126. Association for Computational Linguistics, 2015.

[87] Corina Florescu and Cornelia Caragea. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115. Association for Computational Linguistics, 2017.

[88] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. Deep Keyphrase Generation. *arXiv preprint arXiv:1704.06879*, 2017.

[89] Florian Boudin. Unsupervised Keyphrase Extraction with Multipartite Graphs. *arXiv preprint arXiv:1803.08721*, 2018.

[90] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. A Text Feature Based Automatic Keyword Extraction Method for Single Documents. In *European Conference on Information Retrieval*, pages 684–691. Springer, 2018.

[91] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. YAKE! Collection-independent Automatic Keyword Extractor. In *European Conference on Information Retrieval*, pages 806–810. Springer, 2018.

[92] Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Jorge, Célia Nunes, and Adam Jatowt. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020.

[93] Swagata Duari and Vasudha Bhatnagar. sCAKE: Semantic Connectivity Aware Keyword Extraction. *Information Sciences*, 477:100–117, 2019.

[94] Claude Sammut and Geoffrey I. Webb. Tf-idf. In *Encyclopedia of Machine Learning and Data Mining*, pages 1274–1274. Springer US, Boston, MA, 2017.

[95] Gael Varoquaux. Joblib: running Python functions as pipeline jobs. https://joblib.readthedocs.io/, 2020. [Online; accessed 15 June 2021].

[96] Joel Nothman, Hanmin Qin, and Roman Yurchak. Stop Word Lists in Free Open-source Software Packages. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 7–12, 2018.

[97] Florian Boudin. pke: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING*

2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 69–73, Osaka, Japan, December 2016.

[98] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS 2019)*. The Internet Society, February 2019.

[99] Steven Englehardt and Arvind Narayanan. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the 26th ACM Conference on Computer and Communications Security*, pages 1388–1401, 2016.

[100] Adam Cohen. FuzzyWuzzy: Fuzzy String Matching in Python. https://github.com/seatgeek/fuzzywuzzy, 2011. [Online; accessed 15 December 2020].

[101] Harald Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[102] Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed Word Representations for Multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192. Association for Computational Linguistics, 2013.

[103] Sebastian Raschka. *Python Machine Learning*. Packt Publishing Ltd, 2015.

[104] Fabrice Colas and Pavel Brazdil. Comparison of SVM and Some Older Classification Algorithms in Text Classification Tasks. In *IFIP AI: International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer, 2006.

[105] Kanish Shah, Henil Patel, Devanshi Sanghvi, and Manan Shah. A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. *Augmented Human Research*, 5(1):1–16, 2020.

[106] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[107] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4):550–560, 1997.

[108] Pedro G. Fonseca and Hugo D. Lopes. Calibration of Machine Learning Classifiers for Probability of Default Modelling. *arXiv preprint arXiv:1710.08901*, 2017.

[109] Scott M. Lundberg and Su-In Lee. A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, page 4768–4777. ACM, 2017.

9 Appendix

Table T1. Top 100 selected key phrases in our training sets. This list does not include all key phrases due to space restrictions.

English	German
information, use, service, possible, problem, default setting, connect, clock, firefox, -pron- computer's clock, -pron- securely, change -pron- date, correct time, hsts, http strict, http strict transport, http strict transport security, malicious site, mozilla identify, network security setting, report error, restore, restore default setting, secure connection, strict transport, strict transport security, time setting, transport security, wrong time, mozilla, result, setting, date, -pron- computer, computer, server, http, device, -pron- service, firewall, network connection, content, third party, certificate, exception, connection, privacy policy, cookie, personal information, -pron- personal information, product, privacy, privacy statement, use -pron- service, policy, example, -pron- personal datum, good faith belief, -pron- information, time, use -pron- product, credit risk reduction, collect, feature, -pron- microsoft account, health care professional, microsoft service, other microsoft service, -pron- device use delete browse history, -pron- health datum, -pron- health record datum, -pron- other device, -pron- personal device, -pron- personal microsoft onedrive account, child use -pron- device, control -pron- personal datum, internet explorer use -pron- search query, microsoft donot use -pron- individual recovery key, microsoft edge, microsoft health, microsoft health service, office use other microsoft, other app -pron- install, other microsoft software such, personal microsoft, personal microsoft account, personalized computing environment, privacy shield principles, stick figure representation, use -pron- device, -pron- device use, certain microsoft office product, cortana, headache / migraine, microsoft privacy, microsoft privacy statement, microsoft product, microsoft update service, other microsoft product, website use -pron- personal microsoft account	datum, cookies, personenbezogenen datum, information, ander datum, google, bitte direkt, angezeigt sponsored, angezeigt sponsored listings, beziehung, dienstanbieter, dienstanbieter in irgendeiner, domaininhaber, domaininhaber noch, dritt seite, dritt seite automatisch, irgendeiner beziehung, markenrechtliche problem, markenrechtliche problem auftreten, problem auftreten, seite automatisch generieren, sponsored listings, stehen weder, wenden, whois, whois ersichtlich, all rights, reserved, rights, rights reserved, personenbezogene datum, seite automatisch, welch, all rights reserved, copyright, direkt, 2018 copyright, nutzung, datenschutzerklärung, person, erhoben datum, dienst, beispiel, browser, website, google weit, google-konto, ander geeignet vertraulichkeitsund sicherheitsmaßnahmen, dienst erhoben datum, identifizierend datum, streng vertraulichkeitsverpflichtungen unterwerfen, websites, google analytics, dienst nutzen, nutzern gut dienst, konto, startseite, verarbeitung, suche, weitere information, url, werbeanzeigen, personenbezogener datum, fehler, angefordert datum, ander technologieund kommunikationsunternehmen, google analytics generieren datum, google regelmäßig anfrage, partner google analytics, art, leider, welch datum, verarbeitung personenbezogener datum, sonstig datum, nicht-personenbezogene datum, aktivität, besonder kategorie personenbezogener datum, ander dienst, beispiel werbung, apps, welch datum google, sensibel personenbezogenen datum, datum google, erhobene datum, haben, inhalt, einstellung, google datum, google cookies verwenden, nutzer, sensible personenbezogene datum, google cookies, b. google analytics, neu dienst, alle browsersitzungen beibehalten, ander öffentlich quelle verfügbar, automatische produktupdates anbieten, beispielsweise, beschwerde einreichen haben, bestehend dienst erhoben datum