Surfacing Privacy Settings Using Semantic Matching

Rishabh Khandelwal, Asmit Nayak*, Yao Yao* and Kassem Fawaz

University of Wisconsin–Madison

{*rkhandelwal3, anayak6, yyao69, kfawaz*}*@wisc.edu*

Abstract

Online services utilize privacy settings to provide users with control over their data. However, these privacy settings are often hard to locate, causing the user to rely on provider-chosen default values. In this work, we train privacy-settings-centric encoders and leverage them to create an interface that allows users to search for privacy settings using free-form queries. In order to achieve this goal, we create a custom Semantic Similarity dataset, which consists of real user queries covering various privacy settings. We then use this dataset to fine-tune a state of the art encoder. Using this fine-tuned encoder, we perform semantic matching between the user queries and the privacy settings to retrieve the most relevant setting. Finally, we also use the encoder to generate embeddings of privacy settings from the top 100 websites and perform unsupervised clustering to learn about the online privacy settings types. We find that the most common type of privacy settings are 'Personalization' and 'Notifications', with coverage of 35.8% and 34.4%, respectively, in our dataset.

1 Introduction

Online services provide their users with privacy settings to control information access, collection, processing, and sharing. These settings include, but are not limited to, the option to opt-out of data collection, manage notifications, and marketing emails. However, recent work [6] shows that privacy settings are often hard to locate for an average user. Even when the privacy settings pages are easily accessible, they contain numerous settings that are usually distributed over several URLs, making it challenging to find the settings of interest. For example, Facebook contains 60 privacy settings located on several different URLs. Further, several service providers expose similar settings to the user. It is desirable for the user to apply the same preference to these settings at once, instead of locating and setting their preferences at each domain.

The current privacy settings ecosystem prevents the users from making informed choices for the privacy settings; users unknowingly accept the default options for privacy settings, which tend to favor the interests of the web services over those of the users [19, 22]. Due to their non-standard nature, privacy settings have received little attention in the privacy literature than the more standard opt-out options [7, 30, 13] and privacy policies [8].

There is a need to make privacy settings to be more accessible so that users can exercise more informed choices. Achieving this objective requires crawling the websites for privacy control pages, parsing their HTML to extract the privacy settings, and presenting these settings in a more usable manner. In this paper, we tackle the latter problem of presenting privacy settings to the user by investigating the feasibility of an intra- and inter- domain search interface for privacy settings. In particular, we create a privacy setting-centric encoder and leverage it to answer the following questions:

- *Given privacy settings pages, can we build an interface to allow users to search for privacy settings accurately?*
- *Can we utilize NLP techniques to learn about the landscape of online privacy settings?*

To answer these questions, we first create a custom Semantic Similarity dataset consisting of real user queries covering various privacy settings. We use this dataset to fine-tune the Universal Sentence encoder [2] to match user queries with the

*Equal Contribution

Proceedings of the Second Workshop on Privacy in Natural Language Processing, pages 28–38

https://doi.org/10.18653/v1/P17

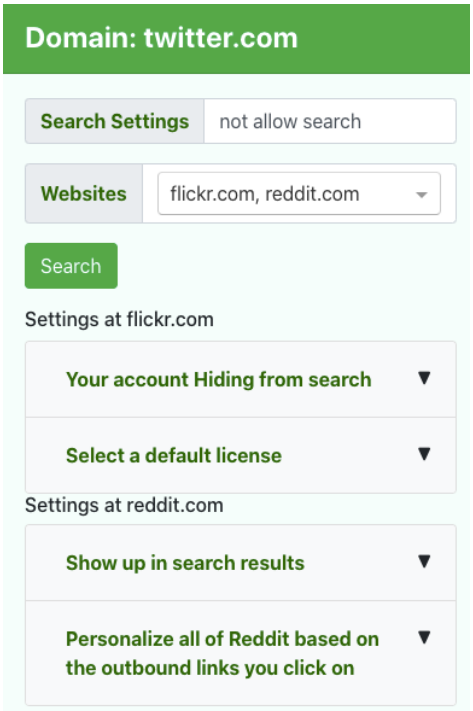

Figure 1: Example of searching across domains

settings text. Next, we represent a privacy settings page as an abstract data structure consisting of privacy groups. We manually extract the text for these privacy groups for the top 100 websites. We demonstrate the feasibility of developing an accurate intra- and inter- domain search interface for privacy settings. Finally, we leverage the developed encoder to analyze the landscape of privacy settings across different websites.

To summarize, we make the following contributions in this paper:

- We provide a manually curated Sentence Similarity Dataset of free-form user queries for privacy settings. Further, we train a privacy-setting-centric sentence encoder using this dataset.

- We develop a browser extension that uses a fine-tuned sentence encoder to enable users to search for settings in the same domain and across domains. We also evaluate the encoder using real-world queries.

- Finally, we leverage the privacy settings' embeddings to analyze the top 100 websites' privacy settings using unsupervised clustering to understand the landscape of privacy settings for popular websites.

2 Related Work

While the existing literature on understanding the landscape of privacy settings is scarce, there has been some work done in the extraction and automation of privacy settings, particularly in mobile apps' context [17, 16, 15]. Chen et al. [3] conducted a large-scale study on the usability of privacy settings for Android applications. Their methodology leverages the semantic relationship between the text descriptions of UI elements and the titles of application views to discover privacy menus hidden in apps. Liu et al. [17] studied the feasibility of generalized privacy profiles, predicting user permission decisions by modeling it as a classification problem. Jialiu et al. [15] similarly studied privacy profiles using unsupervised clustering of user preferences and app permissions. Relatedly, a new privacy application called Jumbo [27] allows users to select one of three available privacy profiles and after that sets the user's privacy settings for their set of supported mobile apps. These works are specific to Android and do not provide insights into the type of privacy settings employed by the services.

In the web domain, Nisal el al. [24] created a browser extension built on top of a previously proposed automated-extraction methodology [30] to capture opt-out choices from a website's privacy policy. The extension developed by Nisal et al. displays the extracted information to users and helps the users to enforce their opt-out settings.

In the privacy domain, prior works have used NLP techniques primarily to understand and analyze privacy policies [8, 25, 23, 1]. In particular, Harkous et al. [8] used a hierarchical text classification system to annotate segments from the privacy policy automatically. They also developed a free-form QA system for queries on privacy policies. Similarly, Oltramari et al. [25] developed PrivOnto, which analyzes privacy policies based on an ontology that uses a semantic framework to represent annotated policies; Nejad et al. [23] used text mining and rule-based semantic technologies for information extraction from contractual agreements. Andow et al. [1] proposed PolicyLint which leverages sentence level NLP techniques to identify internal contradictions within policy by capturing both positive and negative statements of data collection and sharing. There has also been some research done in using NLP for verification of compliance [33, 32] of privacy practices by smartphone apps by comparing the permissions to the

privacy policy text. All these works are essentially focused on privacy policies whereas, in this work, we fine-tune the state-of-the-art sentence encoders to create privacy-setting-centric embeddings.

3 Datasets

In this section, we describe the two datasets that we created for fine-tuning the Universal Sentence Encoder and evaluating semantic matching of the natural language interpreter present in the browser extension.

3.1 Semantic Similarity Dataset

In order to fine-tune the sentence encoders, we created a custom Sentence Similarity Dataset for privacy settings, which consists of real user queries posted on Reddit. These queries are derived from the threads that ask for privacy settings related questions for the top 100[1] domains. This dataset aims to capture semantic relations between a user query and the privacy setting text.

To extract the queries, we followed a methodology similar to that of Harkous et al [8]. In particular, we searched for "privacy setting" and "notification setting" to search the subreddits of each domain from the top 100 domains. To filter out unrelated threads, we keep the ones that are posted as questions by checking for the question mark in the title of the thread. In order to refine our search such that user questions that do not contain question marks are not simply discarded, we also used two classifiers, a Multinomial Naive Bayes classifier [21], and the Stanford NLP package [21].

Using the above process, we collected 596 candidate queries automatically. Next, two of the authors manually analyzed the candidate queries with the following objectives: a) Filter out the queries if it is not related to privacy settings; b) For the remaining queries, find the corresponding setting that resolves the query and c) assign a similarity score between each pair of query and the setting. The assignment of similarity score is done according to the following rules:

- **Score 0 :** The setting and the query are completely unrelated. Example :
  Q : How to get profile edits to show up publicly?
  S : Display media that may contain sensitive content

- **Score 0.5 :** The setting and the query refer to the same general concept but the setting does not resolve the query. Example:
  Q : How to fix issues with comment notifications?
  S : Mentions Notify me when others mention my channel

- **Score 1 :** The query refers to the setting; e.g.
  Q : Comments - No notifications from replies?
  S : Replies to my comments Notify me about replies to my comments

The annotators had an overlapping set of 150 queries to measure the inter-annotator agreement. We observe that they showed high agreement on manual tagging of this set. In particular, they had a high value for Cohen's Kappa ($\kappa = 0.84$) [14]. The queries that the annotators disagreed on were resolved after further discussion. At the end of this process, we have the Sentence Similarity Dataset with a total of 596 query-setting pair consisting of 219 pairs with score 1, 150 pairs with score 0.5, and 227 pairs with score 0.

3.2 Evaluation Dataset

To evaluate the natural language interpreter backed by semantic matching in the browser extension, we created another dataset similar to Semantic Similarity Dataset. The key goals of this dataset are: a) privacy setting queries are free-form, b) each query has a corresponding setting for it. Essentially, this set is an independent set of query-setting pair with a similarity score of 1. To create such a dataset, we followed the same procedure as described in the section above, but only kept the queries for which we could find a corresponding settings option. The final outcome of this process is a set of 110 query-setting pairs covering 20 popular domains including Twitter, Reddit, Amazon etc. It is important to note that the two datasets described in this section have no overlapping queries - the first one is used for fine-tuning the encoders while the second one is used to evaluate the browser extension's semantic

[1] According to top sites ranking by Tranco: https://tranco-list.eu/

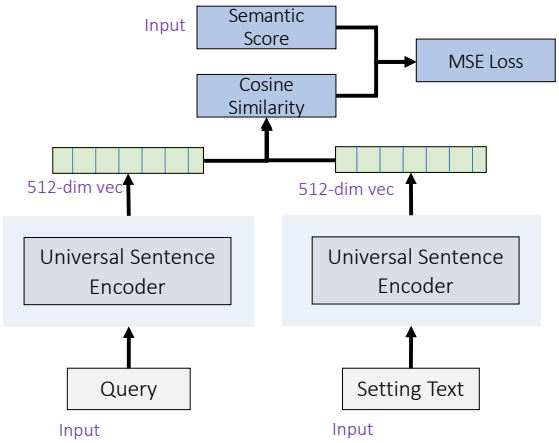

Figure 2: Schematic showing the training procedure for semantic similarity task which is used to fine-tune the Universtal Sentence Encoder

search which is built on top of the fine-tuned encoders. We also note that the Reddit queries might be structurally different than the actual queries. In particular, Reddit queries can be more descriptive whereas the actual queries are generally limited to phrases or keywords. However, since the semantic content is similar (privacy settings), we expect the encoder to perform well in both scenarios.

3.3 Takeaway

The key takeaway here is the semantic similarity dataset consisting of user-queries and privacy settings pairs. This dataset can be utilized to generate privacy-centric embeddings which can then be used to build more privacy-enhancing applications.

4 Fine Tuning the Encoder

Transfer learning has proven to be very effective in several NLP tasks [10, 4, 31, 18]. The key idea is to pre-train general embeddings on large text corpora using an unsupervised loss and then use transfer learning for downstream tasks like text classification, semantic matching, sentiment analysis, etc. The intuition behind this is inspired by the fact that humans do not learn everything from scratch, but extend learned knowledge to new domains. There are two standard techniques used for transfer learning in NLP: fine-tuning and feature-based transfer learning. In feature-based transfer learning, the main idea is to find good feature representation to minimize domain divergence and classification error [26]. In fine-tuning, the pre-trained embeddings are re-trained on downstream tasks (classification, etc.) using labeled data. Further, prior work has shown that fine-tuning achieves better performance than feature-based transfer learning [10]

In this paper, our goal is to generate privacy-settings-centric embeddings for semantic matching to enable users to search for privacy settings. Further, using these embeddings, we can understand the landscape of privacy settings by using unsupervised clustering techniques. To achieve this, we utilize the Semantic Similarity Dataset that we collected (Sec. 3.1). In particular, we fine-tune Universal Sentence Encoder (USE) [2] using the sentence similarity task with the dataset described in Sec. 3.1. USE is trained with a Deep Averaging Network. [11] and is considered as one of the state of the art sentence encoders [2]. We further note that we did not perform any pre-processing on the text as USE is designed to work with the raw textual data.

Finally, we set aside 100 query-setting pairs from the dataset for testing and we train on the remaining 496 query-setting pairs. The schematic of the training procedure is shown in Fig. 2. During training, we first pass the query and the setting text through the networks to get the embedding vectors; then we compute the cosine similarity score using the two vectors. Finally, this score and the manually annotated score are used to compute the Mean Squared Error (MSE) loss. We used Adam [12] optimizer to optimize the loss function and update the weights. We used the test set loss to decide how many iterations the training should go on for. We found that the test set loss was minimized with 10 iterations.

4.1 Evaluation of the Encoder

Next, we evaluate the encoders and the semantic search feature of the browser extension using the evaluation dataset described in Sec. 3.2. This dataset consists of 110 queries from 20 popular domains. For a baseline comparison, we evaluate the performance of semantic matching using the Universal Sentence Encoder (USE) [2] without fine-tuning. We further include two other encoders, based on SBERT [28] and SRoBERTa [28], which are retrained versions of BERT [5] and RoBERTa [20] using siamese and triplet network structures. These models are first pre-trained on Natural Language Inference (NLI) datasets and

Model	Top-1	Top-3	Top-5	Top-7
USE	63.7	84.6	92.6	97.4
SBERT-nli-stsb-base	49.2	74.3	83.4	93.6
SRoBERTa-nli-stsb-base	48.9	72.6	84.2	92.4
USE Fine-tuned	72.9	92.4	98.8	100

Table 1: Top-k accuracy in % for the different encoders in semantic matching on the Evaluation Dataset

then fine-tuned on the Semantic Textual Similarity dataset (STSB).

The evaluation results are compared in Table 1. The results show that the base model of Universal Sentence Encoder outperforms the other encoders in this task. Further, we find that the fine-tuned encoder performs better than all the other encoders tested. In particular, the results show that the user query is answered by the top-3 results 92% of the time on a given domain. Even if the relevant setting is not found in the top 3 results, the user is almost certain to find the relevant group in the top 7 results. This is particularly useful for websites like Facebook, where the number of extracted groups is more than 60. These results further show that the NLP techniques can play an important role in reducing the user's burden, particularly in making it easier to find the relevant privacy settings.

We further analyze the queries and find that the length of the query has an observable effect on semantic matching. In particular, we find that as the length increases (or multiple contexts are present), the performance of semantic matching decreases. For example, consider this user query on Reddit: *"anyone know if there is a way to stop replies on tweets appearing from people you follow to accounts you do not or floods of tweets from people you do not follow but others do"*. This query concerns with replies on the tweets, but is not matched with *"Push Notifications Mentions and Replies"*. However, if we remove the second half of the query (*"from people you follow to accounts you do not or floods of tweets from people you do not follow but others do"*), then the matching is accurate. We hypothesize that the reason for this behavior is that in such cases, the extra information dilutes the the context which results in poor matching.

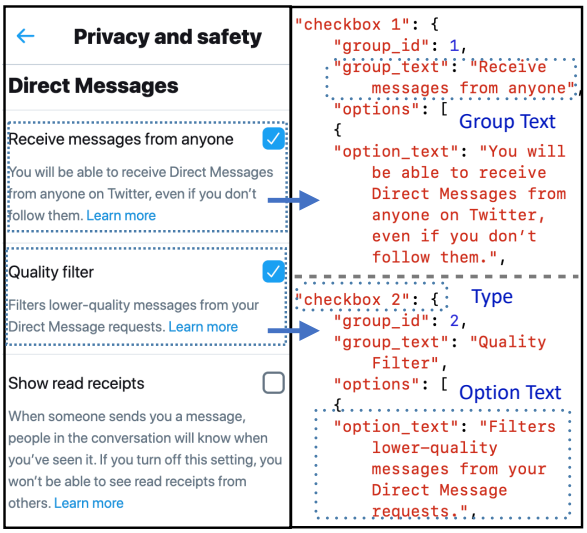

Figure 3: (Left) Example of a privacy settings page on Twitter. (Right) The abstraction of this page showing the, group texts, option texts and the type of setting. This abstraction is used to represent the privacy control pages for all the domains in our set.

4.2 Takeaway

In this section, we demonstrate the use of the Semantic similarity dataset by fine-tuning the Universal Sentence Encoder to generate privacy setting centric embeddings. We further evaluate the fine-tuned encoder with real user-queries and show that it outperforms the other state-of-the-art encoders.

5 The Browser Extension

This section describes the browser extension which leverages the fine-tuned encoder to allow the user to search for privacy settings across multiple domains. We, first, start with a brief introduction of our collection method for privacy settings. We, then, describe the browser extension.

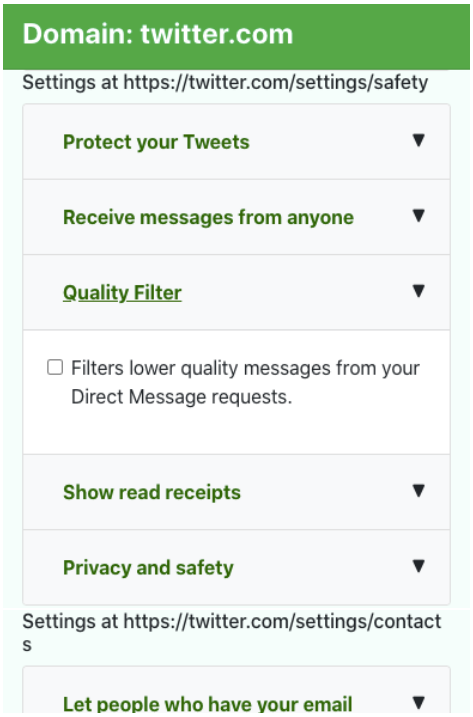

Figure 4: Displays all available settings on twitter.com in accordance with the recipe in Figure 3

Privacy Settings Extraction

In order to allow the extension to perform semantic searches, we need to extract the privacy settings text. Starting with the list of top 100 websites, we visit each domain, go to the privacy settings page, and manually extract the relevant setting text. To represent the privacy control page, we propose an abstraction of the privacy settings page as a set of privacy control groups. The control group is defined as a set of all the individual options. For example, in Fig. 3, the left panel shows the privacy settings page for Twitter. In this figure, there are three control groups, each having one option. This way, each setting option is represented as part of a group with the following attributes: (a) Group Text: text providing context for the entire group, (b) Option Text: text providing context for the option, (c) Type of setting (*radio*, *checkbox*, etc). For example, Fig. 3 shows the representation of the settings page for Twitter. During this process, we extracted privacy settings text for 67 domains. The rest of the domains either did not have privacy control pages or were not in English.

Browser Extension

We implemented the user interface in the form of a Chrome browser extension supported by a natural language query interpreter. The interpreter allows the users to search for privacy settings. The extension is backed with a back-end server where the semantic matching occurs to find the appropriate setting for the user query. The client-side is responsible for handling user interactions as well as displaying the query results from the server.

Once activated by clicking on the icon, the extension renders a basic list interface that shows the privacy settings groups for the given domain. These settings are categorized by the control URLs for that particular domain, as shown in Fig. 4. To change any setting, the user needs to click on the setting text, which opens up the privacy setting URL in a new tab where the user can choose their preferences.

The search interface of the extension allows the users to search for their setting of interest with free-form queries, as shown in Fig. 1. By default, the extension only searches for the currently active website. The user can choose to include other websites they prefer to search from a list of available websites. For example, Fig. 1 shows the search results for the query 'not allow search' for *flickr.com* and *reddit.com*. In this example, the user started from *twitter.com* but then restricted the search list to *flickr.com* and *reddit.com*. The semantic search returns the top two matches for the query. From here, the user can again click on the setting text to navigate to the privacy control page and change their preference.

Thus, the users can now search for privacy settings with free-form queries, which addresses the reachability issues of the privacy settings. Simultaneously, it also reduces the time and effort that would otherwise be needed for the user to look for privacy settings.

5.1 Takeaway

In this section, we present a browser extension that leverages semantic matching using the fine-tuned encoder to enable the users to search for privacy settings. Combined with the high top-3 accuracy of the encoder, the extension addresses the reachability issues of privacy settings and reduces the user effort in locating the settings.

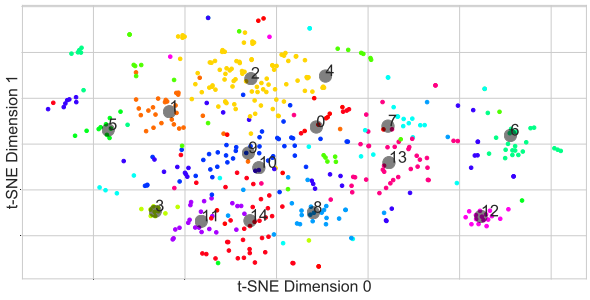

Figure 5: A 2D plot created by applying t-SNE on the multidimensional dataset containing the embeddings of the settings texts, along with the cluster centers from K-Means

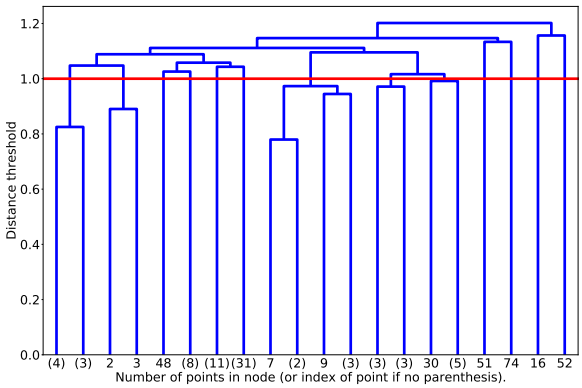

Figure 6: A dendogram showing the hierarchical tree formed from the dataset. The red line is the threshold after which the clustering stops

6 Analysis of Privacy Settings

To further understand the types of privacy settings employed by the web services, we take the privacy settings text that we extracted (Sec. 3.1), encode them using the fine-tuned encoder described in Sec. 4, and group them using unsupervised clustering. We use two types of clustering techniques; K-Means [9] and Agglomerative Hierarchical clustering [29] to perform the clustering. We further compare the results from the two techniques and find that Agglomerative Hierarchical clustering performs significantly better than K-Means clustering.

K-Means Clustering In K-Means clustering, the dataset is broken into 'k'-partitions by initially assigning k-random cluster centers, called centroids. Based on proximity to these clusters, each point in the dataset is assigned one of these random centroids. Finding the mid-point of these centers gives us new centroids and the process is repeated until the new centroid coincides with the previous ones.

To select the optimum value of k in K-Means, we manually analyze the clusters with $k = 5$, 7, 10, 13, 15, 17, and 20. We found that 13 and 15 clusters produced the best results. However, there were certain discrepancies in these clustering. For example, some of the clusters focused on narrow topics like 'Email Notifications' while other clusters contained settings from more than one topic like 'Personalizations' and 'Ads'. Furthermore, the random initial clustering employed by K-means resulted in a number of initial misclassifications of the settings.

Agglomerative Hierarchical Clustering In Agglomerative clustering, each data point is initially a cluster of its own. Then, through repeated iterations, the clusters that are the closest to each other are merged until we get one single cluster with all the data points. The above method creates a tree, as shown in Fig. 6, with the root node containing all the data points. The clusters are obtained from this tree by deciding a threshold, after which the merging stops. We used the maximum Euclidean distance between similar settings as the distance threshold for clustering sentences, which was about 1 unit. Using this, we found a total of 15 clusters with 9 main clusters and 6 outliers. The outliers in clustering can be attributed to their sentence structure and size, which diluted the embeddings.

To manually compare the two clustering techniques, two of the authors independently annotate the clusters generated by K-Means and Agglomerative clustering and concluded that the Agglomerative clustering was more accurate and coherent. They further compare their annotations and obtain high agreement with Cohen's Kappa ($\kappa = 0.87$). We attribute the reason for better performance of Agglomerative clustering to its nature, where the number of clusters is not fixed and a distance threshold is used above which clusters will not be merged. Moreover, the cluster centers of K-Means are randomly assigned. As a result, it is possible that due to the position of the initial cluster center, informative and distinct clusters are not formed. Whereas in Agglomerative clustering, each sentence/setting text is an individual cluster initially, and then these clusters are merged as long as the distance between these clusters is less than equal to the distance threshold.

Since Agglomerative clustering is the better performing method in this context, we used it to

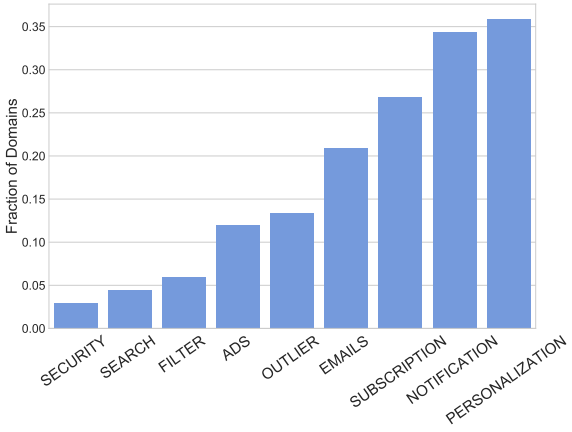

Figure 7: Coverage of the domains for the categories found using unsupervised clustering. We note that Personalization and Notification are the most common type of privacy settings employed by the web services.

cluster all the settings in our database. The main categories we found are: Notification, Emails, Subscriptions, Personalization, Ads, Security, Filter, and Search. Apart from these categories, we also notice a few outlier clusters which represent the specific features of some websites. For example, we observe a cluster with settings enabling contacts management from `twitter.com` and `alibaba.com`.

The coverage of the domains in our set for the categories found above is shown in Fig. 7. We observe that settings corresponding to Personalization and Notifications have the highest coverage. The Personalization category here includes settings directly related to the privacy of the user. For example, limiting the audience for posts for Facebook or allowing people to share images on Instagram. Other categories like Emails and Subscriptions are also covered fairly regularly. These include marketing emails and emails regarding the account - like pull request emails for Github, respectively.

Search and Filter category are examples of website-specific categories: Search includes the settings about the search engine (Bing, Google, Duckduckgo), while Filter includes settings related to quality filters from Google, Twitter, Flickr, and Bing.

We further observe that Security settings, which primarily consist of multi-factor authentication settings, are covered less. This behaviour is not surprising as multi-factor authentication is a new concept and websites are slowly starting to incorporate it. However, the trend observed in the coverage of domain showcases the lack of privacy settings provided by the websites. For example, even in the top-100 websites, only 36% provide the users with Personalization settings.

It is important to note here that the analysis done is highly dependent on the small dataset that we have; it is very likely that we missed several types of settings. For example, cookie settings that allow the user to disable ad tracking are missing. The analysis presented here however does shed light on the common privacy settings like Notification, Subscription and Personalization.

6.1 Takeaway

In this section, we provide another use case of the privacy-centric encoder. We generate the embeddings for privacy setting text of the top 100 websites and use unsupervised clustering to learn about the type of settings used by the websites. We find that there are 7 main categories with notification and personalization settings being the most common ones.

7 Limitations

In this section, we describe the technical limitations of this project.

Scaling The privacy setting text used in the analyses is extracted manually. This approach does not scale very well and is used in this work primarily to show the applicability of privacy-settings-centric encoder. It is however possible to scale the technique by developing web scraping techniques, which is left for future work. Further, scaling the analysis in Sec. 6 using hierarchical clustering to larger datasets is also a limitation as this clustering technique's complexity is quadratic with the number of data points.

Impact on Users While the extension that we propose makes it easy for the users to find privacy settings, the impact and usability of the extension are not studied in this work. In particular, studies exploring the impact of the extension on user choices are considered out of scope and are left as future work.

Completeness The analysis conducted in Sec. 6 with privacy settings is not exhaustive. The goal

here is to show applications of privacy-settings-centric encoders; hence, the analysis is treated as a proof of concept analysis. In particular, we only considered first-party privacy settings, however, some websites outsource their privacy (cookies) settings to third parties. Such settings are not considered here, as a result, some type of settings might have been missed. We accept this as a limitation of this work.

8 Conclusion

In this paper, we present a Sentence Similarity Dataset for privacy settings consisting of real-world user queries about privacy settings and their corresponding settings from several domains. We further use this dataset to fine-tune Universal Sentence Encoder to generate privacy settings centric sentence encoder. To demonstrate the use case of these embeddings, we develop a browser extension that allows the users to search for their privacy settings for popular websites, including Google, Facebook, Twitter, etc. The extension is backed by a natural language interpreter that takes in the user query and performs semantic matching with the privacy settings text. Additionally, we also use the embeddings of privacy settings text to understand the type of privacy settings employed by the website by using unsupervised clustering to find that notification and personalization settings are the most common settings.

Acknowledgement

We would like to thank the anonymous reviewers for their useful comments. The work reported in this paper was supported in part by the NSF under grants 1838733, 1942014, and 2003129.

Availability

The datasets collected in this paper will be made available at https://github.com/wi-pi/surface_privacy_data. We also plan to make the source code public and release the plugin.

References

[1] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. Policylint: investigating internal privacy policy contradictions on google play. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 585–602, 2019.

[2] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[3] Yi Chen, Mingming Zha, Nan Zhang, Dandan Xu, Qianqian Zhao, Xuan Feng, Kan Yuan, Fnu Suya, Yuan Tian, Kai Chen, et al. Demystifying hidden privacy settings in mobile apps. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 570–586. IEEE, 2019.

[4] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in neural information processing systems*, pages 3079–3087, 2015.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[6] Hana Habib, Sarah Pearman, Jiamin Wang, Yixin Zou, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. "it's a scavenger hunt": Usability of websites' opt-out and data deletion choices. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2020.

[7] Hana Habib, Yixin Zou, Aditi Jannu, Neha Sridhar, Chelse Swoopes, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. An empirical analysis of data deletion and opt-out choices on 150 websites. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, 2019.

[8] H Harkous, K Fawaz, R Lebret, F Schaub, KG Shin, and K Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th USENIX Security Symposium (USENIX Security 18)*. USENIX Association, 2018.

[9] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[10] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[11] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691, 2015.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Vinayshekhar Bannihatti Kumar, Roger Iyengar, Namita Nisal, Yuanyuan Feng, Hana Habib, Peter Story, Sushain Cherivirala, Margaret Hagan, Lorrie Faith Cranor, Shomir Wilson, et al. Finding a choice in a haystack: Automatic extraction of opt-out statements from privacy policy text. In *The Web Conference (the Web Conf)*, 2020.

[14] J Richard Landis and Gary G Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

[15] Jialiu Lin, Bin Liu, Norman Sadeh, and Jason I Hong. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 199–212, 2014.

[16] Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun Aerin Zhang, Norman Sadeh, Yuvraj Agarwal, and Alessandro Acquisti. Follow my recommendations: A personalized privacy assistant for mobile app permissions. In *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 27–41, 2016.

[17] Bin Liu, Jialiu Lin, and Norman Sadeh. Reconciling mobile app privacy and usability on smartphones: Could user privacy profiles help? In *Proceedings of the 23rd international conference on World wide web*, pages 201–212, 2014.

[18] Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019.

[19] Yabing Liu, Krishna P Gummadi, Balachander Krishnamurthy, and Alan Mislove. Analyzing facebook privacy settings: user expectations vs. reality. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement*, pages 61–70. ACM, 2011.

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[21] Kartik Nagre. nlp-question-detection. https://github.com/kartikn27/nlp-question-detection, 2019.

[22] Kaweh Djafari Naini, Ismail Sengor Altingovde, Ricardo Kawase, Eelco Herder, and Claudia Niederée. Analyzing and predicting privacy settings in the social web. In *International Conference on User Modeling, Adaptation, and Personalization*, pages 104–117. Springer, 2015.

[23] Najmeh Mousavi Nejad. Semantic analysis of contractual agreements to support end-user interpretation. In *EKAW (Doctoral Consortium)*, 2018.

[24] Namita Nisal, Florian Schaub, Sushain K. Cherivirala, Shomir Wilson, Kanthashree M. Sathyendra, Lorrie Faith Cranor, Margaret Hagan, and Norman Sadeh. Increasing the salience of data use opt-outs online. In *Symposium on Usable Privacy and Security 2017)*, 2017.

[25] Alessandro Oltramari, Dhivya Piraviperumal, Florian Schaub, Shomir Wilson, Sushain Cherivirala, Thomas B Norton, N Cameron Russell, Peter Story, Joel Reidenberg, and

Norman Sadeh. Privonto: A semantic framework for the analysis of privacy policies. *Semantic Web*, 9(2):185–203, 2018.

[26] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[27] Jumbo Privacy. Jumbo. https://www.jumboprivacy.com/, 2019.

[28] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

[29] K Sasirekha and P Baby. Agglomerative hierarchical clustering algorithm-a. *International Journal of Scientific and Research Publications*, 83:83, 2013.

[30] Kanthashree Mysore Sathyendra, Shomir Wilson, Florian Schaub, Sebastian Zimmeck, and Norman Sadeh. Identifying the provision of choices in privacy policy text. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2774–2779, 2017.

[31] Alon Talmor and Jonathan Berant. Multiqa: An empirical investigation of generalization and transfer in reading comprehension. *arXiv preprint arXiv:1905.13453*, 2019.

[32] 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.

[33] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven Bellovin, and Joel Reidenberg. Automated analysis of privacy requirements for mobile apps. In *2016 AAAI Fall Symposium Series*, 2016.