CREATING A HIERARCHICAL ADVICE SYSTEM FOR PRIVACY POLICY CLASSIFICATION

Student: Lukas Busch
Amsterdam University College
Amsterdam
lukas.busch@student.auc.nl

Supervisor: Kasper Welbers
Faculty of Social Sciences, Communication Science
Vrije Universiteit Amsterdam
Amsterdam
k.welbers@vu.nl

Reader: Giovanni Colavizza
Faculty of Humanities, Departement Mediastudies
University of Amsterdam
Amsterdam
g.colavizza@uva.nl

Tutor: Breanndán O Nualláin
Informatics Institute
University of Amsterdam
Amsterdam
o@uva.nl

May 26, 2021

ABSTRACT

A website's privacy policy is meant to inform users, however policies are often long and difficult to understand. A growing number of research uses machine learning to address this problem. This paper focuses on privacy policy classification and introduces a so-called 'advice system' which leverages the hierarchical structure of the annotated data in the OPP-115 privacy policy corpus. Combining this system with a BERT language model that is trained on a corpus of 340K privacy policies yields results that are on par, or by some metrics slightly better ($\approx$1%) than the state of the art. These results are achieved by sacrificing some precision for a higher recall. Furthermore the difference between segment-based and policy-based stratification is explored. It is found that the segment-based method yields better results, but also incorrectly assumes indepence of segements.

List of Abbreviations:

CNN	Convolutional Neural Network
LR	Logistic Regression
SVM	Support Vector Machines
LCN	Local Classifier per Node
LCPN	Local Classifier per Parent Node
LCL	Local Classifier per Level
MLM	Masked Language Model
EPIC	Electronic Privacy Information Center
NLP	Natural Language Processing
FKGL	FleschKincaid Grade Level
BERT	Bidirectional Encoder Representations from Transformers
TPU	Tensor Processing Units
ASIC	Application-Specific Integrated Circuit
GDPR	General Data Protection Regulation

Word Count: 6260

Major: Sciences

***K*eywords** Machine Learning · Hierarchical Classification · Privacy Policy · Advice System · BERT

1 Introduction

In the modern online landscape data is a valuable resource. Therefore, it comes as no surprise that companies are collecting and selling users' data. The specifics of these data collection practices can be found in a company's privacy policy. This is an important resource internet users have to gain insight in their digital trace. However, a study from 2004 shows that only 4,5% of Americans actually read privacy policies (G. R. Milne & Culnan, 2004). In 2008 McDonald and Cranor (2008) calculated that on average a person in the US visits nearly 1500 websites per year, and estimated that it would require almost 200 hours to read the privacy policies of all these websites. Even if one is willing to spend this time, a number of studies have shown that the average education required to comprehend a privacy policy is over high school level (G. R. Milne & Culnan, 2004), or by other measures even at college-junior level (Srinath, Wilson, & Giles, 2020).

There have been multiple studies with the purpose of making it easier for people to read privacy policies. Most of these practices focus on summarization or classification. Either by manually analyzing them (Jensen & Potts, 2004; Bowers, Reaves, Sherman, Traynor, & Butler, 2017), through crowd sourcing (ToS:DR, n.d.), or by using machine learning to automate the process. This last method has gained a lot of popularity in recent years. Harkous et al. (2018) released the Convolutional Neural Network (CNN) based "Polisis", Liu et al. (2018) reported on using Logistic Regression Neural Network (LR), Support Vector Machines (SVM) and CNNs. Kumar et al. (2019) explored the possibilities of FeedForward Networks and Qiu and Lie (2020) developed "Calpric", which is based on deep active learning.

Most research mentioned above used the OPP-115 corpus, which contains annotated data for privacy policies of 115 websites (Wilson et al., 2016). In 2020 Mousavi et al. (2020) fine-tuned the deep contextual model "BERT" (Devlin, Chang, Lee, & Toutanova, 2019) with 140K policies to establish a baseline in privacy policy classification for this corpus. The data in the OPP-115 corpus has a hierarchical structure with three different levels: Category, Subcategory and Value. In all previous research — except the work of Harkous et al. (2018) — this hierarchical structure was ignored. In this research I leverage the hierarchy of data to improve classification results on the Category level. Concretely I make the following contributions:

- Following in the footsteps of Mousavi et al. (2020) I fine-tune a BERT model on a privacy policy corpus, the corpus I use is created by Amos et al. (2020), and contains 340K policies.
- I point out and investigate a difference in stratification methods in previous research. Based on experimental results this research proposes that method of stratification is taken into account when comparing results with previous research, this is not done by Mousavi et al. (2020) when they established their baseline.
- I use the hierarchical structure of the OPP-115 corpus to create an 'advice system' that classifies a privacy policy segment on different levels and uses the combined results to give a final classification score. The system achieves an increase of macro F1 score of around 1% over the previous state of the art. It does so by increasing recall at the cost of precision.

In doing so this research adds to a rapidly evolving literary field, and contributes to online transparency and the readability of privacy policies by exploring different methods of classification. Everything that is mentioned in this research is available on Github. [1]

2 Research Context

The research context for this proposal is divided into three parts. First 2.1 discusses an early historical overview of privacy policy related topics and goes into further detail on the criticisms of natural language privacy policies. Then 2.2 focuses on machine learning applications for privacy policies. Finally 2.3 gives an overview of hierarchical classification and BERT.

2.1 Early historical overview

A privacy policy is a legal document that explains how a user's data is used by another party. In 1995 the European Parliament (1995) released a directive regarding privacy policies. The United States released guidelines shortly thereafter

[1] `https://github.com/luka5132/NLPToS`

(COMMISSION, 1998). In 2018 the rules regarding privacy in Europe have been sharpened, with the introduction of the General Data Protection Regulation (GDPR) (Council of European Union, 2018). This indicates how data protection is very relevant today.

In the earlier years of the internet attempts were made by researchers to create machine readable privacy policies. In 1996 there was PICS, an ambitious project that tried to label online content (Resnick & Miller, 1996). In 2002 W3C's Platform for Privacy Preferences, or P3P launched a similar idea but with privacy as their main focus. P3P inspired other ideas, such as the European initiative of PRIME (PRIME, 2004), or TAMI which was a project from MIT by Weitzner et al. (2006). However, none of these projects gained much traction. With the main criticism that these protocols were too complex (EPIC, 2000). This makes natural language privacy policies the standard today.

As mentioned in the introduction these natural language policies are not without problems. There are two important criticisms that classification models aim to solve: the first is the length of documents. In 2006 Milne et al. (2006) identified a trend of increased document length of website privacy policies. A recent large scale study by Amos et al. (2020) confirms this. Where in 2009 the median length of a privacy policy was around 1600 words, In 2019 this doubled to 3200, with a sharper increase of length in recent years.

The second criticism is on the readability of policies: The study by Amos et al. (2020) also shows an increase in difficulty of readability, they show that readability based on the FleschKincaid Grade Level (FKGL) (Kincaid, Fishburne, Robert P., Richard L., & S., 1975) has increased from 12 to 13, which is equivalent to college level. Other studies show similar results (Jensen & Potts, 2004; G. Milne et al., 2006; Ermakova, Fabian, & Babina, 2015; Zimmeck et al., 2019). Some even show results closer to a FKGL score of 14 (Jensen & Potts, 2004; Srinath et al., 2020). Ermakova et al. (2015) and Zimmeck et al. (2019) highlight the difference in readability between fields and company sizes.

2.2 Machine learning Applications for Privacy Policy Classification

Because the machine readable initiatives of privacy policies did not gain traction, researchers have adopted Natural Language Processing (NLP) techniques to process privacy policies. These techniques are often aimed at classifying pieces of information within a policy to relieve the burden on a reader.

Since 2012 a nonprofit organization called "Terms of Service; Didn't Read" (ToS:DR, n.d.) has used crowd sourcing to annotate privacy policies. They created an extension that is available for users to quickly access this annotated data when on a website. This initiative led to the first few machine learning tools created in this field, as it provided valuable annotated data. In 2012 Ammar, Sadeh, Smith and Wilson (2012) retrieved 50 annotated policies from the ToS;DR website to create a classification tool. Then in 2013 a number of researchers , including Sadeh, Smith and Wilson, launched the "Usable Privacy Policy Project" (Sadeh et al., 2014). They began this project to create more effective privacy notice and they have been responsible for multiple key developments in the field. For example, Ramanath et al. (2014) used semi Hidden Markov Models to exploit the similarity of privacy policies. Zimmeck et al (2014) created a browser extension called Privee, using the annotated data from ToS;DR. Privee is still available, but does not function properly in many cases. Zimmeck et al. (2014) concluded that there was a need for more and better structured annotated data. This came in 2016, when Wilson et al. (2016) created the OPP-115 corpus, which contains 115 privacy policies with over 23K annotations.

This corpus enabled a number of new research initiatives. To mention a few: In 2018 Harkous et al. (2018) used the OPP-115 corpus in combination with 130K policies from Android apps to perform multi label classification with CNNs. They released their results on a website and created a chatbot called PriBot, which you can ask questions regarding privacy policies. In the same year Liu et al. (2018) released a paper on automatic classification of privacy policies, using pre-trained word vectors from the works of Kim (2014). In the next year Kumar et al. (2019) combined the OPP-115 corpus with 150K privacy policies to create a classifier using FeedForward networks. In 2020 Qiu and Lie (2020) created "Calpric", a deep active learning model that has the benefit of only requiring a small amount of labeled data. Mousavi et al. (2020) fine-tuned a BERT model (Devlin et al., 2019) with a corpus of 130K privacy policies and used their model to obtain state of the art results and established a baseline for the OPP-115 corpus. Their results outperformed prevoius state of the art by 5%.

There are other initiatives that did not use the OPP-115 corpus. For example, Nokbeh et al (2018) created their own annotated dataset, which they used in combination with Google prediction API to summarize policies. Recently Zimmeck et al. (2019) — on behalf of the Usable Privacy Policy Project — created a new dataset called APP-350, which contains annotated data of 350 android apps. However, this corpus has not been used in much research yet.

2.3 Hierarchical classification and BERT

In this subsection an overview of two fundamental concepts is given: hierarchical classification and BERT (Devlin et al., 2019).

2.3.1 Hierarchical Classification

Hierarchical classification makes use of the hierarchical relationship of data. Silla and Freitas (2011) explain that all classifications problems can be seen as hierarchical classification. A *normal* classification problem, where the hierarchy of data is ignored or not present, would be considered a flat classification. When the hierarchy is present it is often leveraged using a local approach. Stein et al. (2019) differentiate between three local / top-down approaches: (1) Local Classifier per Node (LCN), with this approach a binary classifier is trained for every child-node in the structure. (2) Local Classifier per Parent Node (LCPN), here one trains a multi-label classifier for every parent-node. Harkous et al. (2018) used this approach to create Polisis. (3) Local Classifier per Level (LCL), this approach trains a multi-label classifier for an entire level in the hierarchical structure.

As was the case with the work of Harkous et al. (2018), hierarchical classification is often used to better predict the leave-nodes of a hierarchical structure. There are however other applications as well. For example, Zhang et al. (2020) used the hierarchical structure of images to improve image generation. Wyk et al. (2019) were able to use the hierarchical structure of data related to patients with a risk of sepsis for better predictions. Both examples show how a hierarchical structure can be leveraged for different purposes.

2.3.2 BERT

BERT was introduced by Devlin et al. (2019) and stands for Bidirectional Encoder Representations from Transformers (BERT). Contrary to previous models BERT was bidirectional instead of unidirectional. This, combined with transformers (Vaswani et al., 2017) and the Masked Language Model (MLM) (Taylor, 1953) that BERT uses, make it an impressive language model that significantly outperformed the previous state of the art. Since its introduction BERT has been the focus point of much research. Liu et al. (2019) — with RoBERTa — and Joshi et al. (2020) — with SpanBert — have contributed to better pre-training of BERT. Lan et al. (2020) released ALBERT, which scales better than BERT. Lee et al. (2019) — BioBERT — and Beltagy et al (2019) — SciBERT — released models with in-domain word embeddings. These embeddings have been proven to be effective when used for problems in their respective domain. Thanks to the works of Zimmeck et al. (2019), Amos et al. (2020) and Srinath et al. (2020), who have each created large privacy policy datasets, it is now possible to fine tune a privacy specific BERT model with more data than ever before.

3 Methodology

The methodology for this research paper is divided into three parts. 3.1 explains the methods and data that were used for training the standard Bert-base-uncased model on a privacy policy corpus of Amos et al. (2020). 3.2 discusses the OPP-115 corpus created by Wilson et al. (2016) and mentions the different stratification methods used in previous research on this corpus. 3.3 introduces the advice system and explains how the hierarchical structure of the OPP-115 corpus is leveraged to improve classification results.

3.1 Domain Specific Training

It has been shown that training Bert on a domain specific corpus can increase results on tasks in that domain, examples of this are SciBert (Beltagy et al., 2019) and BioBERT (Lee et al., 2019). In 2020 Mousavi et al. (2020) trained a BERT model on a corpus of 130K documents and obtained state of the art results when they applied their model to the OPP-115 corpus. They have set a strong baseline for privacy policy classification. Leveraging the recent increase of available data, in this research I mimic this successful approach, but with more data: a corpus by Amos et al. (2020) that contains 340K English privacy policies. From these policies 9.6M segments — or 'sentences' as referred to in the original BERT paper — were retrieved. Each segment had an average of 58 tokens, which adds up to a total of 556M tokens.

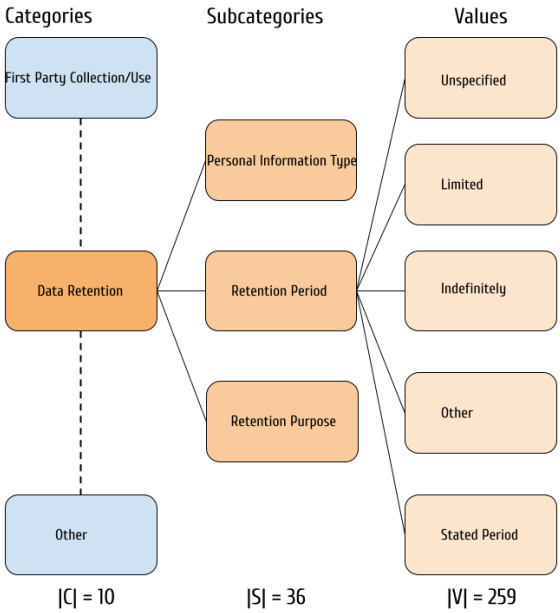

Figure 1: Example structure of OPP-115 dataset. Highlighting the class *Data Retention* and its subclass *Retention Period*.

Training the model was done on Google Colab, where one can use a Tensor Processing Unit (TPU) free of charge (Bisong, 2019). A TPU is an Application-Specific Integrated Circuit (ASIC) created by Google and is currently the quickest and most efficient tool available for machine learning projects. TPUs were created to be used with Tensorflow, but it is now possible to use pytorch by using the pytorch_XLA package. To make use of the pre-training scripts provided on the Bert Github Repo (Devlin et al., 2019) this project uses Tensorflow version 1.15.

The default method of fine-tuning a BERT model was used, e.g. the Adam optimizer and the tokenizer provided in the Bert repository. First the model was trained for 1M steps with a maximum sequence length of 128 per segment and a learning rate of 2e-5. Then the sequence length was increased to 512 and the model was trained for an additional 200K steps with the same learning rate. Finally, with the same maximum sequence length but with a learning rate of 1e-5, the model was trained for another 200K steps. The final loss of the model was 0.22. When training was done, the tensorflow model was converted to a pytorch model that was used to create classification models.

3.2 OPP-115 corpus

Wilson et al (2016), as a part of the Usable Privacy Policy Project, created the OPP-115 corpus in 2016. This corpus contains annotations for 115 privacy policies of US companies. The policies were annotated by law students and each paper was annotated by three different annotators. For every paragraph-sized-segment of a policy an annotator could choose out of 10 categories (C) that best described that segment. These categories were then subdivided into subcategories (S). For example if a segment was classified as on Data Retention, the subcategories for this category are: Personal Information Type; Retention Period; and Retention Purpose (See fig 1 for an example). These subcategories could then be divided into smaller classes, which correspond to the value of the subcategory (V). Thus the data has a hierarchical structure with three different levels.

Because every policy was annotated by three different annotators there can be disagreements among annotators as to which labels should be given to a segment. This can be addressed by using a majority vote: where one only considers a label for a segment if two or more annotators agree on the label. Liu et al. (2018) used this in their work. However, in the case of Harkous et al (2018) the disagreement between annotators is not mentioned and thus it can be assumed that all segments were used in their research. Mousavi et al. (2020) used both methods in setting their standard and showed that using only the data that had a majority vote gave better results. This is to be expected, since a data driven model can only function as well as the data it trains on. For this research only labels that received a majority vote are considered.

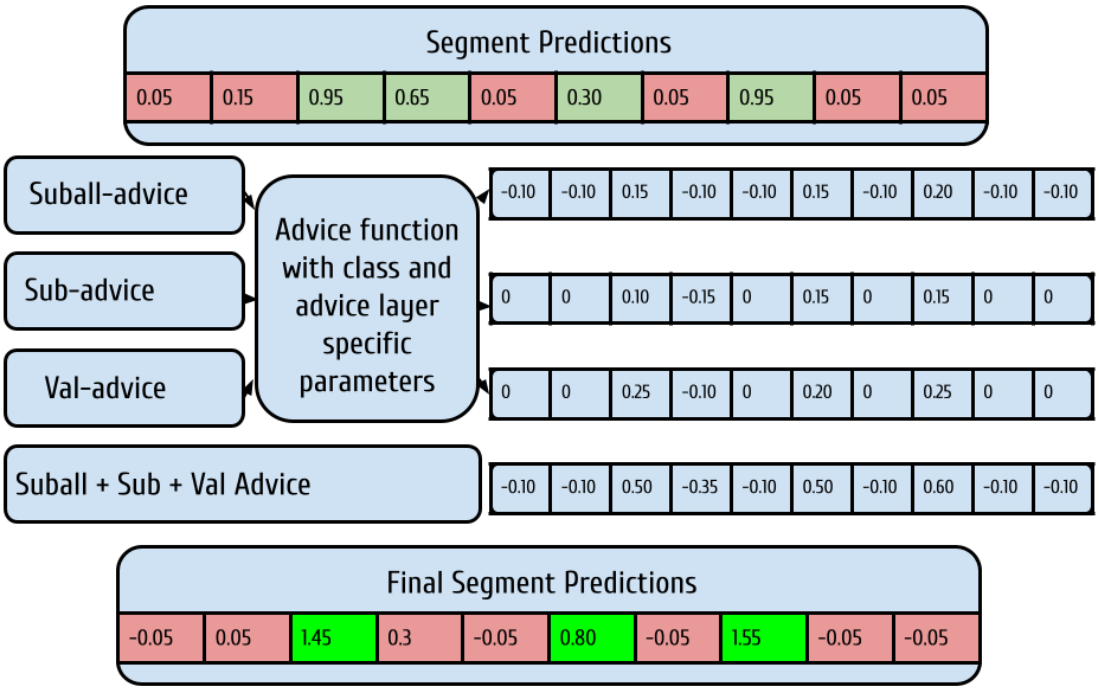

Figure 2: Example of a fictional segment and predictions. In this example classes 3,4,6 and 8 are considered a *candidate* and for each advice layer the corresponding advice is calculated. Here all advice layers are taken into account and thus are added to the original prediction. Finally the labels are decided with a threshold value of 0.6

Previous research has indicated a sparsity of data for some categories (Wilson et al., 2016; F. Liu et al., 2018). For example — using a majority vote — the 'Do Not Track' category appears in only 31 out of 3729 segments. However, Kumar et al. (2019) showed that this does not necessarily lead to poorer results for those classes, as long as the data is properly stratified. In this research, the data is stratified per policy, the reasoning behind this will be explained in the next paragraph. For this research a 5-fold validation was performed with 75 policies as training data, 15 policies as validation data and 30 policies for testing. For more information on label distribution please see table 8

The methods of dividing the data into a training, test and/or validation set varies per paper. Whereas Liu et al. (2018) and Mousavi et al. (2020) use a segment-based stratification method, others such as Wilson et al. (2016) and Harkous et al. (2018) divide the data per policy. For this research the second option was chosen, because the method used by Liu et al. (2018) and Mousavi et al. (2020) assume an independence of segments. However, since segments are all nested within a policy, they are not independent and thus should be split accordingly. I have ran a 15 fold experiment using a classification model for the categories to test if there is a difference between stratification method, outcomes of which can be found in section 4.2.

3.3 Classification structure

The goal of this research is to increase classification results for the top class (C), which is the 10 categories. In most previous research the hierarchical structure of the data is ignored and multi-label classification for only the 10 categories is performed on each segment (Wilson et al., 2016; F. Liu et al., 2018; Kumar et al., 2019; Mousavi et al., 2020). This is not the case in the research by Harkous et al. (2018). They trained 20 separate classifiers. One for the categories, and for each category a classifier that predicts the respective subcategories and values, which is LCPN classification. For this research the fine-tuned BERT model — that was created using the method described in subsection 3.1 — was used as a basis for 18 different classification models: Two LCL models were trained to classify all Categories and Subcategories, and another 16 LCPN models were trained for each category to its respective subcategories and values. For categories with only one subcategory no category-subcategory models were trained. This is the case for the following categories: *Data Security*; *Do Not Track*; *International and Specific Audiences*, and *Other*. The training parameters for each model were determined using a grid search and can be found in table 9 underneath the bibliography.

With the approach of Harkous et al. (2018) one is able to classify on all levels of the data. However, their model can only be as good as the top-level classifier. Inspired by Zhang et al. (2020) and Franco et al. (2019), who were able to get better results by using all levels of hierarchical data in their respective domains, I use the following method to leverage the hierarchical structure of privacy policies:

First multi-label classification is performed to obtain prediction-scores for a segment. Now, instead of using one threshold to determine which labels are considered 'true' and which are 'false' the model uses a 'candidate threshold'. This threshold decides whether a prediction is considered a 'candidate', i.e. will be further evaluated. The optimal value for the candidate threshold was found to be 0.1 through a grid search. One benefit of the candidate threshold is that it reduces time complexity, because not all 16 LCPN models are used for every segment.

Then, for each candidate the corresponding classifier predicts the subcategories and the values. Based on the number of labels that the classifier finds an 'advice' is given, which is simply an increase or decrease of the original prediction score for the 'candidate'. The advice value is determined by a combination of 'advice layers'. In total there are three advice layers. The first is the 'suball' layer, here one classifier predicts the labels for all 36 subcategories, this is a Local Classifier per Level (LCL) approach and is used for every segment. The second layer is the 'sub' layer, this layer also predicts the subcategories but now only for a respective candidate. In the example of figure 1, if *Data Retention* would be considered a candidate — i.e. the category classifier predict a score that is higher than 0.1 — this layer predicts the three subcategories. The third layer is called the 'val' layer and predicts the values for a candidate. In case of the example that would mean the five values that belong to the *Retention Period* subcategory and the values that belong to the *Personal Information Type* and *Retention Purpose* subcategories respectively. Both the 'sub' and 'val' layers use Local Classifier per Parent Node (LCPN) classifiers. The advice given is then decided with the following simple equation :

$$advice(x_c) = \begin{cases} \alpha_c + n * \beta_c, & \text{if } n \geq 1 \\ \gamma_c, & \text{if } n = 0 \end{cases}$$

Here x_c is the candidate, *n* the number of labels that were found, α_c the base-advice value, β_c the subsequent-advice value and γ_c the negative-advice value. The value of α_c, β_c and γ_c depend on the category of the candidate. Thus there are 10 different values for these parameters. The reason why the values differ per category is due to the imbalance in number of subcategories ($|S_c|$) and values ($|V_c|$) per category. The values are chosen through a grid search and can be found in table 7, located underneath the bibliography. The simplicity of this function makes it so that it is easily interpretable and with it one is able to quickly gauge how a given advice was determined. For demonstration purposes one can see fictional advice for a fictional segment in figure 2. In this example all the advice layers are added to the original predictions. This is of course not the only possibility, with three different layers there are 8 possible combinations, for research purposes the results of all possible combinations are considered.

After all advice has been given the model produces a final prediction for a segment, using a threshold value of 0.6.

By using the process described above this research leverages the hierarchical data of privacy policies to create a model that is able to pick up on a multiple aspects that could determine a segments' category and makes this interpretable in the form of advice. However, it should be noted that this does come with an increase in space- and time complexity, since in total 18 classification models are used. Therefore this model would not be suited for private direct use, for example as a browser extension such as ToS:DR and Privee (ToS:DR, n.d.; Zimmeck & Bellovin, 2014). Instead this model could be used when the output (i.e. classification results) can be stored and used for further purpose.

4 Results

In table 1 one can see the average prediction scores for the five different test sets. For each category the scores shown represent the micro F1 scores. As mentioned earlier, all possible combinations layers are shown. In reality one would have to make a decision and pick one combination. When discussing the result of the advice system in further references the combination of the *suball* and *val* layer is meant, since this combination yields the best results. It shows an improvement of 1.6% over the base prediction — from 78.2% to 79.8% — with respect to the Macro Avg score, while the micro Avg stays roughly the same, at 84.4%. The scores for this combination per category are also close to the maximum scores per respective category, there are no major outliers, which is a desirable quality. Note however, how there is not one clear combination that significantly outperforms the others.

Category	Base	+sub	+suball	+val	+suball,sub	+sub,val	+suball,val	+suball, sub,val
First Party Collection/Use	0.911	0.913	0.914	0.914	**0.918**	0.914	0.917	0.917
Third Party Sharing/Collection	0.871	0.871	**0.875**	0.871	**0.875**	0.870	0.874	**0.875**
User Access, Edit & Deletion	0.793	0.802	0.807	0.801	0.813	0.797	**0.815**	0.808
Data Retention	0.398	0.442	0.373	**0.503**	0.444	0.498	0.495	0.486
Data Security	**0.837**	**0.837**	0.834	0.811	0.834	0.811	0.822	0.822
International/Specific Audiences	0.900	0.900	0.903	**0.904**	0.903	**0.904**	0.898	0.898
Do Not Track	0.702	0.702	0.586	**0.781**	0.586	**0.781**	**0.781**	**0.781**
Policy Change	0.876	0.876	0.879	0.834	**0.880**	0.829	0.849	0.832
User Choice/Control	0.773	0.771	**0.794**	0.750	0.789	0.749	0.760	0.754
Other	0.762	0.762	0.767	0.761	**0.768**	0.761	**0.768**	0.766
Micro Avg	0.842	0.844	0.848	0.839	**0.849**	**0.849**	0.844	0.842
Macro Avg	0.782	0.788	0.773	0.793	0.781	0.791	**0.798**	0.794

Table 1: Results per advice layer for hierarchical classification. Scores shown per category represent the *micro* F1 score. Each score is the mean of the scores for the respective test sets. Boldface numbers are the highest scores per category. The base prediction is the initial multi label prediction for the 10 categories. Each subsequent column shows the scores with a combination of advice layers added. The *+suball* layer is the LCL classifier for the subcategories. Both *+sub* and *+val* layers are LCPN classifier from the category nodes to the subcategory and value nodes respectively

In table 2 one can see the results of the *suball, val* combination layer compared to the macro and micro scores obtained by Mousavi et al. (2020) who have set a baseline using their own fine-tuned BERT model, their baseline outperformed the previous state-of-the art by 5%. Although the base model — the fine-tuned BERT model — used for this research performs slightly worse, the results with the advice layers are on par with the state of the art.

Metric	Baseline	Results
Micro Avg	0.85	0.84
Macro Avg	0.79	0.80

Table 2: Comparison of results

4.1 Precision and Recall

In the previous section one can see how the results are promising, however F1 scores do not tell the full story, in this subsection I look at what these F1 scores signify. Upon further inspection the following interesting observation can be made: the advice system nearly always gives a positive advice. This results in a model that has more true positives but also more false positives. Thus, it trades of some precision for a higher recall.

This can be seen quickly when one compares precision and recall of the original predictions and the predictions with the *suball* and *val* advice layer, which is shown in table 3. The decision of whether one should use the base model or the model with advice layers is thus not as simple as the F1 scores would suggest, but rather comes down to what one finds more important. The implications of this are further discussed in the Discussion.

To understand this trade-off between precision and recall even better the following paragraph discusses results for the fifth test set. Please do note that results might vary per stratification.

Category	Base		+suball,val	
	prec	rec	prec	rec
First Party Collection/Use	0.926	0.896	0.921	0.913
Third Party Sharing/Collection	0.885	0.858	0.876	0.873
User Access, Edit & Deletion	0.862	0.739	0.854	0.784
Data Retention	0.900	0.263	0.704	0.406
Data Security	0.898	0.787	0.830	0.820
International/Specific Audiences	0.925	0.878	0.901	0.896
Do Not Track	1.000	0.547	1.000	0.657
Policy Change	0.931	0.832	0.860	0.849
User Choice/Control	0.850	0.712	0.730	0.807
Other	0.813	0.724	0.780	0.757
Total Avg	0.899	0.724	0.846	0.776

Table 3: Precision and recall scores for the base model and the model with the *+suball, val* advice layer

The fifth test set had 959 segments, and for 257 of these segments the base prediction had an incorrect classification, i.e. it classified at least one label as either a false positive or false negative. For 15 of these segments the advice system turned the base predictions into a correct classification, in all cases this was done by turning one false negative from the base prediction into a true positive. On the other hand, the advice system caused 39 extra incorrect classifications on predictions that were previously correct. This was all done by turning at least one true negative into a false positive. That leaves 242 segments that both the base prediction and the advice system classified incorrectly. For the exact values in this test set please see table 4.

Pedictions	TP	TN	FP	FN	Precision	Recall
Base	927	8304	152	207	0.859	0.817
+suball,val	970	8247	209	164	0.823	0.855

Table 4: number of True Positives (TP), True Negatives (TN), False Positives (FN) and False Negatives (FN) as well as precision and recall for the fifth test set

In this test set the base prediction produced a total of 152 false positives and 192 false negatives, whereas the advice system produced 169 false positives and 164 false negatives. Thus in total the advice system made 11 mistakes less for this set. Roughly half of the mislabeled segments had at least a miss-classification for the *Other* category, with 126 and 122 cases for the base prediction and advice system respectively.

4.1.1 Error analysis

To get a better understanding of the errors the system makes I discuss three example cases in detail: (1) The advice system improved the prediction; (2) the advice system worsened the prediction; (3) both the base and advice prediction are wrong.

Example 1 *[segment from http://www.randomhouse.com (policy id: 3737) (Wilson et al., 2016)]*

In the following example the base prediction had correctly predicted that the text should be categorized as 'Other', it also had the correct suspicion that it might be about 'First Party Collection/Use', however it wasn't sure enough to pass the final threshold of 0.6. Considering the advice function parameters — found in table 7 — one can conclude that the suball classifier found 4 labels related to 'First Party Collection/Use' and the val classifier found 3 labels. Thus both layers gave a positive advice which correctly resulted in this segment being classified as 'First Party Collection/Use' as well as 'Other'.

> Combination of Your Information
> When you use more than one of our Sites, we may match information collected from you through each of those Sites and combine that information into a single user record. We may also use and/or combine information we collect off-line or from third party sources to enhance and check the accuracy of your user records.

Base	0.469	0.007	0.004	0.004	0.007	0.008	0.002	0.008	0.004	0.888
+ suball	0.13	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
+ val	0.11	0	0	0	0	0	0	0	0	0.28
Final	0.709	-0.043	-0.046	-0.046	-0.043	-0.042	-0.048	-0.042	-0.046	1.118

Example 2 *[segment from http://www.highgearmedia.com (policy id: 3837) (Wilson et al., 2016)]*

In this example the advice layers incorrectly classify two extra labels. The base prediction correctly identifies 'Do Not Track' as the only present category for this segment. However, it is not entirely certain as there are 3 possible 'candidates'. The *suball* layer is correct in it's predictions as it only strengthens the prediction of 'Do Not Track' where it found 1 label. The *val* layer on the other hand found labels for all candidates. The three labels found for 'Third Party Sharing/Collection' were not enough to change the outcome of the final prediction. This is not the case for the 'International/Specific Audiences' and 'Policy Change' categories, here the *val* classifiers found 3 and 11 labels respectively. One can see how the advice from the *val* layer dominates the advice from the *suball* layer and incorrectly pushes the prediction scores for 'International/Specific Audiences' and 'Policy Change' just over the final threshold of 0.6.

> How we respond to Do Not Track Disclosures .
>
> Some browsers have a do not track feature that lets you tell websites that you do not want to have your online activities tracked. Because these features are not yet uniform, we are not

currently set up to respond to do not track signals. Our information and disclosure practices will continue to apply as described in this privacy policy regardless of any Do Not Track signals that are sent by certain browsers.

Base	0.069	0.124	0.028	0.061	0.094	0.203	0.910	0.113	0.028	0.042
+ suball	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	0.25	-0.05	-0.05	-0.05
+ val	0	0.11	0	0	0	0.45	0.25	0.54	0	0
Final	0.019	0.184	-0.032	0.011	0.044	0.603	1.310	0.603	-0.022	-0.008

Example 3 *[segment from http://www.archives.gov (policy id: 3712) (Wilson et al., 2016)]*

In the following example one can see the most common scenario with mistakes, which is when both the base prediction and the advice layers are wrong. In the example the base prediction was able to correctly identify the segment to discuss 'First Party Collection/Use'. However, its prediction score for 'Third Party Sharing/Collection' is so low that the category will not be considered a candidate, even though the segment is labeled as such. Because of this the *val* layer will not give advice for this segment. The *suball* layer is also not able to find one label for this category and thus the prediction score will remain low. The other category that was annotated is 'Other'. Although the base prediction was high enough for the category to be considered a candidate, the 2 labels that were found with the *val* classifier were not enough to increase the prediction score to a minimum of 0.6.

Some of our websites (such as Founder's Online) use a feature called "Web Storage" (which includes HTML5 "local storage") so users can retrieve searches or user-defined preferences, or data marked as a favorite. This feature creates a storage file on the user"s local hard drive that holds links to the resources from the website that the individual user previously searched or identified as a preference or favorite. No information from this file is transferred to NARA or any other website. Local storage files can be cleared through your browser settings or disabled, similar to cookies.

Base	0.958	0.002	0.010	0.009	0.005	0.005	0.003	0.007	0.016	0.274
+ suball	0.21	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05	-0.05
+ val	0.13	0	0	0	0	0	0	0	0	0.22
Final	1.296	-0.048	-0.040	-0.041	-0.045	-0.045	-0.047	-0.043	-0.034	0.444

4.2 Stratification Method

In table 6 one can see the average label distribution over 15 training sets for segment and policy stratification respectively. One can see that the number of labels are very similar for the two different methods. A bigger difference can be found when one compares the standard deviation. For the standard method the standard-deviation of total segments for the 15 training sets is 4.1, whereas the standard-deviation for the policy based method is 94.3. The difference in standard deviation is highest for categories that have many labels, such as *First party Collection/Use*, *Third Party Sharing/Collection* and *Other*. For these classes the standard-deviations are: 0 against 41.1; 0 against 30.3 and 0.5 against 30.7 respectively.

Evaluation	Segment	Policy
Micro Avg	**0.847**	0.836
Macro Avg	**0.803**	0.789

Table 5: F1-scores for different methods of stratification

In table 5 one sees the mean prediction scores that were obtained using only the classification model for the categories. These results are further discussed in the discussion.

Category Name	Standard Method	Policy Based Method
First Party Collection/Use	907	910
Third Party Sharing/Collection	709	699
User Access, Edit & Deletion	112	111
Data Retention	58.7	59.6
Data Security	157	161
International/Specific Audiences	226	226
Do Not Track	23.0	22.5
Policy Change	89.0	90.1
User Choice/Control	268	267
Other	793	806
Total Segments	2804	2800

Table 6: Label distribution for different methods of stratification

5 Discussion

From the results showcased in the previous section it can be concluded that the advice system is on par with the state of the art and that the advice layers trade off some precision for a higher recall. For privacy policy classification the trade-off between precision and recall means a trade-off between adding incorrect information and leaving out information about a segment. For applications that focus on making a privacy policy more navigable a higher recall could be desirable, since a user of such an application would still be able to verify whether the application directed the user to the correct segment. The implications of making a mistake for such an application are small, at worst a user has to go look for the correct segment a little bit longer. For applications that aim to quickly inform a user and where no user verification is possible this is different. A higher recall would imply that users would be misinformed more often. At the same time a user would also have a lower chance of missing out on important information. The possible applications mentioned above are only a few, but showcase how the choice between a classification model that prioritizes either precision or recall depends on the use-case.

Harkous et al. (2018) made the argument that it is just as important to detect the absence of a label as the presence. With respect to that statement the advice system becomes less favourable over the base model. Since — for the fifth test set, see table 4 — the advice layer adds 43 true positives at the cost of 57 true negatives. On the other hand, Mousavi et al. (2020) say that they do not find this way of measurement fair for multi-label classification.

Another observation made in this research is about the difference in stratification methods of privacy policies. In the results shown in section 4.2 one can see that the segment-based method of stratification yields better results than the policy-based method. This could be the result of the higher variance per training and testing set. However, it could be that when one divides the data based only on segments the classifier learns something about the *style* of a policy and that this information is used to better classify segments for the test set. The segment-based method seems to assume an independence of segments, whereas the segments are nested within a policy and are thus dependent. This might create an unfair advantage in favour of the segment-based stratification method. Whether it is due to the first or second reason — or perhaps a combination of both — this difference should be taken into account when comparing results with previous research. However, when Mousavi et al. (2020) established their baseline, they did not address this.

Overall the results of the advice system are promising and more research is needed to explore the full potential of such a system for privacy policy classification. Some potential areas for further research are:

- A different — more complex — advice function. The tendency of giving positive advice is partly due the *naive* advice function used for this research. Since some of the value (V) classes have 69 labels and the training data is sparse it is rare that no labels are found for such a class, which nearly always leads to a positive advice. Addressing this could increase the effectiveness of advice from the 'val' layer.
- Although there is certainly room for improvement for the advice function, using the current advice function with different parameters for each advice layer could already improve results. For this research a grid search was performed to find the parameters per category, these parameters were then applied to each layer. This is of course not optimal since for each layer the number of labels per category are different. To find parameters with a grid search one is required to increase the computations with an exponent per layer. It scales according to the formula: P^n, where P is the total number of parameter combinations and n the number of layers . For the 56000 different combinations of parameters used in this research running a grid search for multiple layers would quickly become too much. If one is able to find a better solution to finding optimal parameters that is able to account for the difference in layers one might find better results, even when using the same advice function.
- Big improvements could me made in terms of space- and time-complexity. Since the purpose of this research was to explore the use of the advice system not a lot of attention has been given to complexity optimization. Each classification model was trained using the *BertForSequenceClassification* function from the Huggingface library (Huggingface, 2018). This way way each model takes up 440MB. Instead, one could only train a classification layer for each model and would only have to store the pre-trained weights once.
- By increasing space efficiency one could explore the use of an additional advice layer. By adding a layer from subcategories to values one reduces the number of labels for the value class considerably. The obtained results could be used to 'strengthen' the results of the individual subcategories in the same way that was done for the categories in this research. If one sees an improvement in results for the classification of the subcategories this could translate in an improvement for the categories as well.

6 Conclusion

In this research paper the use of an 'advice system' is investigated for the purpose of privacy policy classification. Using the OPP-115 corpus, created by Wilson et al. (2016) 18 BERT models (Devlin et al., 2019) were trained that together form the advice system. For the best combination of advice layers the system slightly outperforms the base classification model in average macro F1 score and is on par with the state of the art. The advice layers have a bias towards giving positive advice, which results in a higher recall at the cost of some precision. Depending on a user's choice this might be preferable or not. In this paper it is also shown how results may vary for different methods of privacy policy stratification. Stratifying the data by segments yields better results than stratification by policies. I argue that this should be taken into account when comparing results with previous studies. Overall, the results obtained by the advice system lead to promising directions.

References

Ammar, W., Wilson, S., Sadeh, N., & Smith, N. A. (2012). Automatic categorization of privacy policies: A pilot study..

Amos, R., Acar, G., Lucherini, E., Kshirsagar, M., Narayanan, A., & Mayer, J. (2020). *Privacy policies over time: Curation and analysis of a million-document dataset.*

Beltagy, I., Lo, K., & Cohan, A. (2019). *Scibert: A pretrained language model for scientific text.*

Bisong, E. (2019). *Building machine learning and deep learning models on google cloud platform: A comprehensive guide for beginners*. doi: 10.1007/978-1-4842-4470-8

Bowers, J., Reaves, B., Sherman, I., Traynor, P., & Butler, K. R. B. (2017). Regulators, mount up! analysis of privacy policies for mobile money services. In *Soups.*

COMMISSION, F. T. (1998). Privacy online: A report to congress. federal trade commission june 1998.. Retrieved from `https://www.ftc.gov/sites/default/files/documents/reports/privacy-online-report-congress/priv-23a.pdf`

Council of European Union. (2018). *General data protection regulation.* (`https://gdpr-info.eu/`)

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *Bert: Pre-training of deep bidirectional transformers for language understanding.*

EPIC. (2000, June). Pretty poor privacy: An assessment of p3p and internet privacy.. Retrieved from https://epic.org/reports/prettypoorprivacy.html

Ermakova, T., Fabian, B., & Babina, E. (2015, 03). Readability of privacy policies of healthcare websites..

Harkous, H., Fawaz, K., Lebret, R., Schaub, F., Shin, K. G., & Aberer, K. (2018). *Polisis: Automated analysis and presentation of privacy policies using deep learning.*

Huggingface. (2018). Bert.. Retrieved from https://huggingface.co/transformers/model_doc/bert.html

Jensen, C., & Potts, C. (2004, 01). Privacy policies as decision-making tools: An evaluation of online privacy notices. In (p. 471-478).

Joshi, M., Chen, D., Liu, Y., Weld, D. S., Zettlemoyer, L., & Levy, O. (2020). *Spanbert: Improving pre-training by representing and predicting spans.*

Kim, Y. (2014). *Convolutional neural networks for sentence classification.*

Kincaid, J. P., Fishburne, J., Robert P., R., Richard L., C., & S., B. (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel:*. Retrieved from http://www.dtic.mil/docs/citations/ADA006655 doi: 10.21236/ADA006655

Kumar, V. B., Ravichander, A., Story, P., & Sadeh, N. (2019). Quantifying the effect of in-domain distributed word representations : A study of privacy policies..

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). *Albert: A lite bert for self-supervised learning of language representations.*

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019, Sep). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*. Retrieved from http://dx.doi.org/10.1093/bioinformatics/btz682 doi: 10.1093/bioinformatics/btz682

Liu, F., Wilson, S., Story, P., Zimmeck, S., & Sadeh, N. (2018). Towards automatic classification of privacy policy text..

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). *Roberta: A robustly optimized bert pretraining approach.*

McDonald, A. M., & Cranor, L. (2008). The cost of reading privacy policies..

Milne, G., Culnan, M., & Greene, H. (2006, 09). A longitudinal assessment of online privacy notice readability. *Journal of Public Policy Marketing - J PUBLIC POLICY MARKETING*, *25*, 238-249. doi: 10.1509/jppm.25.2.238

Milne, G. R., & Culnan, M. J. (2004). Strategies for reducing online privacy risks: Why consumers read (or don't read) online privacy notices. *Journal of Interactive Marketing*, *18*(3), 15–29. doi: https://doi.org/10.1002/dir.20009

Mousavi, N., Jabat, P., Nedelchev, R., Scerri, S., & Graux, D. (2020, 02). Establishing a strong baseline for privacy policy classification..

Nokhbeh Zaeem, R., German, R., & Barber, K. (2018, 08). Privacycheck: Automatic summarization of privacy policies using data mining. *ACM Transactions on Internet Technology*, *18*, 1-18. doi: 10.1145/3127519

PRIME. (2004). Privacy and identity management for europe. fp6-ist - information society technologies: thematic priority under the specific programme "integrating and strengthening the european research area".. Retrieved from https://cordis.europa.eu/project/id/507591

Qiu, W., & Lie, D. (2020, 08). Deep active learning with crowdsourcing data for privacy policy classification.

Ramanath, R., Liu, F., Sadeh, N., & Smith, N. (2014, 06). Unsupervised alignment of privacy policies using hidden markov models. In (Vol. 2, p. 605-610). doi: 10.3115/v1/P14-2099

Resnick, P., & Miller, J. (1996). Pics: Internet access controls without censorship. *Commun. ACM*, *39*, 87-93.

Sadeh, N., Acquisti, A., Breaux, T., Cranor, L., Smith, N. A., Reidenbergb, J. R., ... Schaub, F. (2014). The usable privacy policy project : Combining crowdsourcing , machine learning and natural language processing to semi-automatically answer those privacy questions users care about..

Silla, C., & Freitas, A. (2011, 01). A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, *22*, 31-72. doi: 10.1007/s10618-010-0175-9

Srinath, M., Wilson, S., & Giles, C. L. (2020). *Privacy at scale: Introducing the privaseer corpus of web privacy policies.*

Stein, R. A., Jaques, P. A., & Valiati, J. F. (2019, Jan). An analysis of hierarchical text classification using word embeddings. *Information Sciences*, *471*, 216–232. Retrieved from http://dx.doi.org/10.1016/j.ins.2018.09.001 doi: 10.1016/j.ins.2018.09.001

Taylor, W. L. (1953). "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, *30*(4), 415-433. Retrieved from https://doi.org/10.1177/107769905303000401 doi: 10.1177/107769905303000401

ToS:DR. (n.d.). *Frontpage – tos;dr.* Retrieved from https://tosdr.org/

Union, E. (1995). Directive 95/46/ec of the european parliament and of the council of 24 october 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data.. Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A31995L0046

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). *Attention is all you need.*

Weitzner, D., Abelson, H., Berners-Lee, T., Hanson, C., Hendler, J., Kagal, L., ... Waterman, K. (2006, 01). Transparent accountable data mining: New strategies for privacy protection.

Wilson, S., Schaub, F., Dara, A. A., Liu, F., Cherivirala, S., Giovanni Leon, P., ... Sadeh, N. (2016, August). 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)* (pp. 1330–1340). Berlin, Germany: Association for Computational Linguistics. Retrieved from `https://www.aclweb.org/anthology/P16-1126` doi: 10 .18653/v1/P16-1126

Wyk, F., Khojandi, A., & Kamaleswaran, R. (2019, 01). Improving prediction performance using hierarchical analysis of real-time data: A sepsis case study. *IEEE Journal of Biomedical and Health Informatics*, *PP*, 1-1. doi: 10.1109/JBHI.2019.2894570

Zhang, R., Mou, L., & Xie, P. (2020). *Treegan: Incorporating class hierarchy into image generation.*

Zimmeck, S., & Bellovin, S. (2014). Privee: An architecture for automatically analyzing web privacy policies. In *Usenix security symposium.*

Zimmeck, S., Story, P., Smullen, D., Ravichander, A., Wang, Z., Reidenberg, J., ... Sadeh, N. (2019, 07). Maps: Scaling privacy compliance analysis to a million apps. *Proceedings on Privacy Enhancing Technologies*, *2019*, 66-86. doi: 10.2478/popets-2019-0037

Category	Label Threshold	Base Advice (α_c)	Subsequent Advice (β_c)	Negative Advice (γ_c)
First Party Collection/Use	0.5	0.05	0.02	-0.05
Third Party Sharing/Collection	0.5	0.05	0.02	-0.05
User Access, Edit & Deletion	0.6	0.15	0.02	-0.05
Data Retention	0.4	0.05	0.02	0.05
Data Security	0.4	0.05	0.08	-0.05
International/Specific Audiences	0.4	0.15	0.1	-0.05
Do Not Track	0.4	0.25	0.1	-0.05
Policy Change	0.4	0.1	0.04	-0.05
User Choice/Control	0.5	0.15	0.06	-0.05
Other	0.4	0.1	0.06	-0.05

Table 7: Parameter for the advice function per category

Category	Full Support	Full % Support	Train Support	Val Support	Test Support
First Party Collection/Use	1209	27.1	747	146	315
Third Party Sharing/Collection	945	21.2	594	120	231
User Access, Edit & Deletion	149	3.3	95	18.0	36.0
Data Retention	78	1.7	52.8	7.2	18
Data Security	210	4.7	129.2	25.6	55.2
International/Specific Audiences	301	6.8	183	37.8	80.2
Do Not Track	31	0.7	19.8	3.6	7.6
Policy Change	119	2.7	69.0	16.4	33.6
User Choice/Control	358	8.0	217	51.8	89.2
Other	1058	23.7	681	121	255
Total	4458	100	2788	548	1122

Table 8: Average number of labels per category for 5 fold validation

Model name	max_length	learning rate	epochs
Categories	128	2e-5	9
Subcategories	128	2e-5	9
cs_Data Retention	128	5e-6	6
cs_First Party Collection/Use	512	2e-5	9
cs_Policy Change	128	5e-6	3
cs_Third Party Sharing/Collection	128	1e-5	9
cs_User Access, Edit & Deletion	128	5e-6	6
cs_User Choice/Control	128	2e-5	9
cv_Data Security	128	5e-6	3
cs_Data Retention	512	1e-5	3
cv_Do Not Track	128	1e-5	9
cv_First Party Collection/Use	128	2e-5	9
cv_International/Specific Audiences	128	2e-5	6
cv_Other	128	5e-6	9
cv_Policy Change	512	5e-6	6
cv_Third Party Sharing/Collection	128	2e-5	9
cv_User Access, Edit & Deletion	128	5e-6	9
cv_User Choice/Control	512	5e-6	3

Table 9: Training parameters for all classification models that were trained using BertForSequenceClassification. Categories and Subcategories are LCL classifiers. Other names represent LCPN classifiers where cs stands for *category to subcategory* and cv for *category to value*. All models were trained with a batch size of 16.