Intent Classification and Slot Filling for Privacy Policies

Wasi Uddin Ahmad[†,*], **Jianfeng Chi**[‡,*], **Tu Le**[‡]
Thomas Norton[§], **Yuan Tian**[‡], **Kai-Wei Chang**[†]
[†]University of California, Los Angeles, [‡]University of Virginia, [§]Fordham University
[†]{wasiahmad, kwchang}@cs.ucla.edu
[‡]{jc6ub,tnl6wk,yuant}@virginia.edu
[§]tnorton1@law.fordham.edu

Abstract

Understanding privacy policies is crucial for users as it empowers them to learn about the information that matters to them. Sentences written in a privacy policy document explain privacy practices, and the constituent text spans convey further specific information about that practice. We refer to predicting the privacy practice explained in a sentence as intent classification and identifying the text spans sharing specific information as slot filling. In this work, we propose PolicyIE, an English corpus consisting of 5,250 intent and 11,788 slot annotations spanning 31 privacy policies of websites and mobile applications. PolicyIE corpus is a challenging real-world benchmark with limited labeled examples reflecting the cost of collecting large-scale annotations from domain experts. We present two alternative neural approaches as baselines, (1) intent classification and slot filling as a joint sequence tagging and (2) modeling them as a sequence-to-sequence (Seq2Seq) learning task. The experiment results show that both approaches perform comparably in intent classification, while the Seq2Seq method outperforms the sequence tagging approach in slot filling by a large margin. We perform a detailed error analysis to reveal the challenges of the proposed corpus.

1 Introduction

Privacy policies inform users about how a service provider collects, uses, and maintains the users' information. The service providers collect the users' data via their websites or mobile applications and analyze them for various purposes. The users' data often contain sensitive information; therefore, the users must know how their information will be used, maintained, and protected from unauthorized and unlawful use. Privacy policies are meant to explain all these use cases in detail. This makes privacy policies often very long, complicated, and confusing (McDonald and Cranor, 2008; Reidenberg et al., 2016). As a result, users do not tend to read privacy policies (Commission et al., 2012; Gluck et al.; Marotta-Wurgler, 2015), leading to undesirable consequences. For example, users might not be aware of their data being sold to third-party advertisers even if they have given their consent to the service providers to use their services in return. Therefore, automating information extraction from verbose privacy policies can help users understand their rights and make informed decisions.

In recent years, we have seen substantial efforts to utilize natural language processing (NLP) techniques to automate privacy policy analysis. In literature, information extraction from policy documents is formulated as text classification (Wilson et al., 2016a; Harkous et al., 2018; Zimmeck et al., 2019), text alignment (Liu et al., 2014; Ramanath et al., 2014), and question answering (QA) (Shvartzshanider et al., 2018; Harkous et al., 2018; Ravichander et al., 2019; Ahmad et al., 2020). Although these approaches effectively identify the sentences or segments in a policy document relevant to a privacy practice, they lack in extracting fine-grained structured information. As shown in the first example in Table 1, the privacy practice label "Data Collection/Usage" informs the user how, why, and what types of user information will be collected by the service provider. The policy also specifies that users' "username" and "icon or profile photo" will be used for "marketing purposes". This informs the user precisely what and why the service provider will use users' information.

The challenge in training models to extract fine-grained information is the lack of labeled examples. Annotating privacy policy documents is expensive as they can be thousands of words long and requires domain experts (e.g., law students). Therefore, prior works annotate privacy policies at the

[*]Equal contribution. Listed by alphabetical order.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4402–4417
August 1–6, 2021.

[We]$_{\text{Data Collector: First Party Entity}}$ may also [use]$_{\text{Action}}$ or display [your]$_{\text{Data Provider: user}}$ [username]$_{\text{Data Collected: User Online Activities/Profiles}}$ and [icon or profile photo]$_{\text{Data Collected: User Online Activities/Profiles}}$ on [marketing purpose or press releases]$_{\text{Purpose: Advertising/Marketing}}$. *Privacy Practice*. Data Collection/Usage
[We]$_{\text{Data Sharer: First Party Entity}}$ do [not]$_{\text{Polarity: Negation}}$ [sell]$_{\text{Action}}$ [your]$_{\text{Data Provider: user}}$ [personal information]$_{\text{Data Shared: General Data}}$ to [third parties]$_{\text{Data Receiver: Third Party Entity}}$. *Privacy Practice*. Data Sharing/Disclosure

Table 1: Annotation examples from PolicyIE Corpus. Best viewed in color.

sentence level, without further utilizing the constituent text spans to convey specific information. Sentences written in a policy document explain privacy practices, which we refer to as *intent classification* and identifying the constituent text spans that share further specific information as *slot filling*. Table 1 shows a couple of examples. This formulation of information extraction lifts users' burden to comprehend relevant segments in a policy document and identify the details, such as how and why users' data are collected and shared with others.

To facilitate fine-grained information extraction, we present PolicyIE, an English corpus consisting of 5,250 intent and 11,788 slot annotations over 31 privacy policies of websites and mobile applications. We perform experiments using sequence tagging and sequence-to-sequence (Seq2Seq) learning models to jointly model intent classification and slot filling. The results show that both modeling approaches perform comparably in intent classification, while Seq2Seq models outperform the sequence tagging models in slot filling by a large margin. We conduct a thorough error analysis and categorize the errors into seven types. We observe that sequence tagging approaches miss more slots while Seq2Seq models predict more spurious slots. We further discuss the error cases by considering other factors to help guide future work. We release the code and data to facilitate research.[1]

2 Construction of PolicyIE Corpus

2.1 Privacy Policies Selection

The scope of privacy policies primarily depends on how service providers function. For example, service providers primarily relying on mobile applications (e.g., Viber, Whatsapp) or websites and applications (e.g., Amazon, Walmart) have different privacy practices detailed in their privacy policies. In PolicyIE, we want to achieve broad coverage across privacy practices exercised by the service providers such that the corpus can serve a wide variety of use cases. Therefore, we go through the following steps to select the policy documents.

Initial Collection Ramanath et al. (2014) introduced a corpus of 1,010 privacy policies of the top websites ranked on `Alexa.com`. We crawled those websites' privacy policies in November 2019 since the released privacy policies are outdated. For mobile application privacy policies, we scrape application information from Google Play Store using `play-scraper` public API[2] and crawl their privacy policy. We ended up with 7,500 mobile applications' privacy policies.

Filtering First, we filter out the privacy policies written in a non-English language and the mobile applications' privacy policies with the app review rating of less than 4.5. Then we filter out privacy policies that are too short (< 2,500 words) or too long (> 6,000 words). Finally, we randomly select 200 websites and mobile application privacy policies each (400 documents in total).[3]

Post-processing We ask a domain expert (working in the security and privacy domain for more than three years) to examine the selected 400 privacy policies. The goal for the examination is to ensure the policy documents cover the four privacy practices: (1) *Data Collection/Usage*, (2) *Data Sharing/Disclosure*, (3) *Data Storage/Retention*, and (4) *Data Security/Protection*. These four practices cover how a service provider processes users' data in general and are included in the General Data Protection Regulation (GDPR). Finally, we shortlist 50 policy documents for annotation, 25 in each category (websites and mobile applications).

[1] https://github.com/wasiahmad/PolicyIE

[2] https://github.com/danieliu/play-scraper

[3] We ensure the mobile applications span different application categories on the Play Store.

2.2 Data Annotation

Annotation Schema To annotate sentences in a policy document, we consider the first four privacy practices from the annotation schema suggested by Wilson et al. (2016a). Therefore, we perform sentence categorization under five *intent classes* that are described below.

(1) *Data Collection/Usage*: What, why and how user information is collected;
(2) *Data Sharing/Disclosure*: What, why and how user information is shared with or collected by third parties;
(3) *Data Storage/Retention*: How long and where user information will be stored;
(4) *Data Security/Protection*: Protection measures for user information;
(5) *Other*: Other privacy practices that do not fall into the above four categories.

Apart from annotating sentences with privacy practices, we aim to identify the text spans in sentences that explain specific details about the practices. For example, in the sentence *"we collect personal information in order to provide users with a personalized experience"*, the underlined text span conveys the purpose of data collection. In our annotation schema, we refer to the identification of such text spans as *slot filling*. There are 18 slot labels in our annotation schema (provided in Appendix). We group the slots into two categories: type-I and type-II based on their role in privacy practices. While the type-I slots include participants of privacy practices, such as *Data Provider*, *Data Receiver*, type-II slots include purposes, conditions that characterize more details of privacy practices. Note that type-I and type-II slots may overlap, e.g., in the previous example, the underlined text span is the *purpose* of data collection, and the span "user" is the *Data Provider* (whose data is collected). In general, type-II slots are longer (consisting of more words) and less frequent than type-I slots.

In total, there are 14 type-I and 4 type-II slots in our annotation schema. These slots are associated with a list of attributes, e.g., *Data Collected* and *Data Shared* have the attributes *Contact Data*, *Location Data*, *Demographic Data*, etc. Table 1 illustrates a couple of examples. We detail the slots and their attributes in the Appendix.

Annotation Procedure General crowdworkers such as Amazon Mechanical Turkers are not suitable to annotate policy documents as it requires specialized domain knowledge (McDonald and Cranor, 2008; Reidenberg et al., 2016). We hire two law students to perform the annotation. We use the web-based annotation tool, BRAT (Stenetorp et al., 2012) to conduct the annotation. We write a detailed annotation guideline and pretest them through multiple rounds of pilot studies. The guideline is further updated with notes to resolve complex or corner cases during the annotation process.[4] The annotation process is closely monitored by a domain expert and a legal scholar and is granted IRB exempt by the Institutional Review Board (IRB). The annotators are presented with one segment from a policy document at a time and asked to perform annotation following the guideline. We manually segment the policy documents such that a segment discusses similar issues to reduce ambiguity at the annotator end. The annotators worked 10 weeks, with an average of 10 hours per week, and completed annotations for 31 policy documents. Each annotator is paid $15 per hour.

Dataset	Train	Test
# Policies	25	6
# Sentences	4,209	1,041
# Type-I slots	7,327	1,704
# Type-II slots	2,263	494
Avg. sentence length	23.73	26.62
Avg. # type-I slot / sent.	4.48	4.75
Avg. # type-II slot / sent.	1.38	1.38
Avg. type-I slot length	2.01	2.15
Avg. type-II slot length	8.70	10.70

Table 2: Statistics of the PolicyIE Corpus.

Post-editing and Quality Control We compute an inter-annotator agreement for each annotated segment of policy documents using Krippendorff's Alpha (α_K) (Klaus, 1980). The annotators are asked to discuss their annotations and re-annotate those sections with token-level α_K falling below 0.75. An α_K value within the range of 0.67 to 0.8 is allowed for tentative conclusions (Artstein and Poesio, 2008; Reidsma and Carletta, 2008). After the re-annotation process, we calculate the agreement for the two categories of slots individually. The inter-annotator agreement is 0.87 and 0.84 for type-I and type-II slots, respectively. Then the adjudicators discuss and finalize the annotations. The adjudication process involves one of the annotators, the legal scholar, and the domain expert.

[4] We release the guideline as supplementary material.

Joint intent and slot tagging
Input: [CLS] We may also use or display your username and icon or profile photo on marketing purpose or press releases .
Type-I slot tagging output
Data-Collection-Usage B-DC.FPE O O B-Action O O B-DP.U B-DC.UOAP O B-DC.UOAP I-DC.UOAP I-DC.UOAP I-DC.UOAP O O O O O O O
Type-II slot tagging output
Data-Collection-Usage O O O O O O O O O O O O O O B-P.AM I-P.AM I-P.AM I-P.AM I-P.AM O
Sequence-to-sequence (Seq2Seq) learning
Input: We may also use or display your username and icon or profile photo on marketing purpose or press releases .
Output: [IN:Data-Collection-Usage [SL:DC.FPE *We*] [SL:Action *use*] [SL:DP.U *your*] [SL:DC.UOAP *username*] [SL:DC.UOAP *icon or profile photo*] [SL:P.AM *marketing purpose or press releases*]]

Table 3: An example of input / output used to train the two types of models on PolicyIE. For brevity, we replaced part of label strings with symbols: DP.U, DC.FPE, DC.UOAP, P.AM represents Data-Provider.User, Data-Collector.First-Party-Entity, Data-Collected.User-Online-Activities-Profiles, and Purpose.Advertising-Marketing.

Data Statistics & Format Table 2 presents the statistics of the PolicyIE corpus. The corpus consists of 15 and 16 privacy policies of websites and mobile applications, respectively. We release the annotated policy documents split into sentences.[5] Each sentence is associated with an intent label, and the constituent words are associated with a slot label (following the BIO tagging scheme).

3 Model & Setup

PolicyIE provides annotations of privacy practices and corresponding text spans in privacy policies. We refer to privacy practice prediction for a sentence as *intent classification* and identifying the text spans as *slot filling*. We present two alternative approaches; the first approach jointly models intent classification and slot tagging (Chen et al., 2019), and the second modeling approach casts the problem as a sequence-to-sequence learning task (Rongali et al., 2020; Li et al., 2021).

3.1 Sequence Tagging

Following Chen et al. (2019), given a sentence $s = w_1, \ldots, w_l$ from a privacy policy document D, a special token ($w_0 = [\text{CLS}]$) is prepended to form the input sequence that is fed to an encoder. The encoder produces contextual representations of the input tokens $h_0, h_1, \ldots, h_l$ where h_0 and $h_1, \ldots, h_l$ are fed to separate softmax classifiers to predict the target intent and slot labels.

$$y^i = \text{softmax}(W_i^T h_0 + b_i),$$
$$y_n^s = \text{softmax}(W_s^T h_n + b_s), n \in 1, \ldots l,$$

where $W_i \in R^{d \times I}, W_s \in R^{d \times S}, b_r \in R^I$ and $b_i \in R^I, b_s \in R^S$ are parameters, and I, S are the total number of intent and slot types, respectively. The sequence tagging model (composed of an encoder and a classifier) learns to maximize the following conditional probability to perform intent classification and slot filling jointly.

$$P(y^i, y^s|s) = p(y^i|s) \prod_{n=1}^{l} p(y_n^s|s).$$

We train the models end-to-end by minimizing the cross-entropy loss. Table 3 shows an example of input and output to train the joint intent and slot tagging models. Since type-I and type-II slots have different characteristics as discussed in § 2.2 and overlap, we train two separate sequential tagging models for type-I and type-II slots to keep the baseline models simple.[6] We use BiLSTM (Liu and Lane, 2016; Zhang and Wang, 2016), Transformer (Vaswani et al., 2017), BERT (Vaswani et al., 2017), and RoBERTa (Liu et al., 2019) as encoder to form the sequence tagging models.

Besides, we consider an embedding based baseline where the input word embeddings are fed to the softmax classifiers. The special token (w_0 =

[5] We split the policy documents into sentences using UDPipe (Straka et al., 2016).

[6] Span enumeration based techniques (Wadden et al., 2019; Luan et al., 2019) can be utilized to perform tagging both types of slots jointly, and we leave this as future work.

Model	# param (in millions)	Intent F1	Type-I		Type-II	
			Slot F1	EM	Slot F1	EM
Human	-	96.5	84.3	56.6	62.3	55.6
Embedding	1.7	$50.9_{\pm 27.3}$	$19.1_{\pm 0.3}$	$0.8_{\pm 0.3}$	$0.0_{\pm 0.0}$	$0.0_{\pm 0.0}$
BiLSTM	8	$75.9_{\pm 1.1}$	$40.8_{\pm 0.9}$	$7.6_{\pm 0.9}$	$3.9_{\pm 3.0}$	$10.0_{\pm 2.7}$
Transformer	34.8	$80.1_{\pm 0.6}$	$41.0_{\pm 3.5}$	$6.5_{\pm 2.8}$	$3.5_{\pm 1.0}$	$13.1_{\pm 2.4}$
BERT	110	$\mathbf{84.7}_{\pm 0.7}$	$55.5_{\pm 1.1}$	$17.0_{\pm 1.1}$	$29.6_{\pm 2.4}$	$24.2_{\pm 4.2}$
RoBERTa	124	$84.5_{\pm 0.7}$	$54.2_{\pm 1.9}$	$14.3_{\pm 2.4}$	$29.8_{\pm 1.7}$	$24.8_{\pm 1.4}$
Embedding w/ CRF	1.7	$67.9_{\pm 0.6}$	$26.0_{\pm 1.5}$	$1.20_{\pm 0.3}$	$5.7_{\pm 4.6}$	$3.1_{\pm 0.6}$
BiLSTM w/ CRF	8	$76.7_{\pm 1.4}$	$45.1_{\pm 1.2}$	$9.2_{\pm 0.9}$	$26.8_{\pm 2.2}$	$18.1_{\pm 2.0}$
Transformer w/ CRF	34.8	$77.9_{\pm 2.7}$	$43.7_{\pm 2.3}$	$8.9_{\pm 3.0}$	$5.7_{\pm 0.9}$	$11.0_{\pm 2.1}$
BERT w/ CRF	110	$82.1_{\pm 2.0}$	$56.0_{\pm 0.8}$	$\mathbf{19.2}_{\pm 1.1}$	$31.7_{\pm 1.9}$	$19.7_{\pm 2.6}$
RoBERTa w/ CRF	124	$83.3_{\pm 1.6}$	$\mathbf{57.0}_{\pm 0.6}$	$18.2_{\pm 1.2}$	$\mathbf{34.5}_{\pm 1.3}$	$\mathbf{27.7}_{\pm 3.9}$

Table 4: Test set performance of the sequence tagging models on PolicyIE corpus. We individually train and evaluate the models on intent classification and type-I and type-II slots tagging and report average intent F1 score.

[CLS]) embedding is formed by applying average pooling over the input word embeddings. We train WordPiece embeddings with a 30,000 token vocabulary (Devlin et al., 2019) using *fastText* (Bojanowski et al., 2017) based on a corpus of 130,000 privacy policies collected from apps on the Google Play Store (Harkous et al., 2018). We use the hidden state corresponding to the first WordPiece of a token to predict the target slot labels.

Conditional Random Field (CRF) helps structure prediction tasks, such as semantic role labeling (Zhou and Xu, 2015) and named entity recognition (Cotterell and Duh, 2017). Therefore, we model slot labeling jointly using a conditional random field (CRF) (Lafferty et al., 2001) (only interactions between two successive labels are considered). We refer the readers to Ma and Hovy (2016) for details.

3.2 Sequence-to-Sequence Learning

Recent works in semantic parsing (Rongali et al., 2020; Zhu et al., 2020; Li et al., 2021) formulate the task as sequence-to-sequence (Seq2Seq) learning. Taking this as a motivation, we investigate the scope of Seq2Seq learning for joint intent classification and slot filling for privacy policy sentences. In Table 3, we show an example of encoder input and decoder output used in Seq2Seq learning. We form the target sequences by following the template: [IN:LABEL [SL:LABEL $w_1, \ldots, w_m$] ...]. During inference, we use greedy decoding and parse the decoded sequence to extract intent and slot labels. Note that we only consider text spans in the decoded sequences that are surrounded by "[]"; the rest are discarded. Since our proposed PolicyIE corpus consists of a few thousand examples, instead of training Seq2Seq models from scratch, we fine-tune pre-trained models as the baselines. Specifically, we consider five state-of-the-art models: MiniLM (Wang et al., 2020), UniLM (Dong et al., 2019), UniLMv2 (Bao et al., 2020), MASS (Song et al.), and BART (Lewis et al., 2020).

3.3 Setup

Implementation We use the implementation of BERT and RoBERTa from `transformers` API (Wolf et al., 2020). For the Seq2Seq learning baselines, we use their public implementations.[7,8,9] We *train* BiLSTM, Transformer baseline models and *fine-tune* all the other baselines for 20 epochs and choose the best checkpoint based on validation performance. From 4,209 training examples, we use 4,000 examples for training (~95%) and 209 examples for validation (~5%). We tune the learning rate in [1e-3, 5e-4, 1e-4, 5e-5, 1e-5] and set the batch size to 16 in all our experiments (to fit in one GeForce GTX 1080 GPU with 11gb memory). We train (or fine-tune) all the models five times with different seeds and report average performances.

Evaluation Metrics To evaluate the baseline approaches, we compute the F1 score for intent classification and slot filling tasks.[10] We also compute an exact match (EM) accuracy (if the predicted intent matches the reference intent and slot F1 = 1.0).

[7] https://github.com/microsoft/unilm

[8] https://github.com/microsoft/MASS

[9] https://github.com/pytorch/fairseq/tree/master/examples/bart

[10] We use a micro average for intent classification.

Model	# param (in millions)	Intent F1	Type-I		Type-II	
			Slot F1	EM	Slot F1	EM
Human	-	96.5	84.3	56.6	62.3	55.6
MiniLM	33	$83.9_{\pm 0.3}$	$52.4_{\pm 1.5}$	$19.8_{\pm 1.6}$	$40.4_{\pm 0.4}$	$27.9_{\pm 1.6}$
UniLM	110	$83.6_{\pm 0.5}$	$58.2_{\pm 0.7}$	$28.6_{\pm 1.2}$	$53.5_{\pm 1.4}$	$\mathbf{35.4}_{\pm 1.9}$
UniLMv2	110	$\mathbf{84.7}_{\pm 0.5}$	$61.4_{\pm 0.9}$	$\mathbf{29.9}_{\pm 1.2}$	$53.5_{\pm 1.5}$	$33.5_{\pm 1.5}$
MASS	123	$81.8_{\pm 1.2}$	$54.1_{\pm 2.5}$	$21.3_{\pm 2.0}$	$44.9_{\pm 1.2}$	$25.3_{\pm 1.3}$
BART	140	$83.3_{\pm 1.1}$	$53.6_{\pm 1.7}$	$10.6_{\pm 1.7}$	$52.4_{\pm 2.7}$	$27.5_{\pm 2.2}$
	400	$83.6_{\pm 1.3}$	$\mathbf{63.7}_{\pm 1.3}$	$23.0_{\pm 1.3}$	$\mathbf{55.2}_{\pm 1.0}$	$31.6_{\pm 2.0}$

Table 5: Test set performance of the Seq2Seq models on PolicyIE corpus.

Human Performance is computed by considering each annotator's annotations as predictions and the adjudicated annotations as the reference. The final score is an average across all annotators.

4 Experiment Results & Analysis

We aim to address the following questions.

1. How do the two modeling approaches perform on our proposed dataset (§ 4.1)?
2. How do they perform on different intent and slot types (§ 4.2)?
3. What type of errors do the best performing models make (§ 4.3)?

4.1 Main Results

Sequence Tagging The overall performances of the sequence tagging models are presented in Table 4. The pre-trained models, BERT and RoBERTa, outperform other baselines by a large margin. Using conditional random field (CRF), the models boost the slot tagging performance with a slight degradation in intent classification performance. For example, RoBERTa + CRF model improves over RoBERTa by 2.8% and 3.9% in terms of type-I slot F1 and EM with a 0.5% drop in intent F1 score. The results indicate that predicting type-II slots is difficult compared to type-I slots as they differ in length (type-I slots are mostly phrases, while type-II slots are clauses) and are less frequent in the training examples. However, the EM accuracy for type-I slots is lower than type-II slots due to more type-I slots (~4.75) than type-II slots (~1.38) on average per sentence. Note that if models fail to predict one of the slots, EM will be zero.

Seq2Seq Learning Seq2Seq models predict the intent and slots by generating the labels and spans following a template. Then we extract the intent and slot labels from the generated sequences. The experiment results are presented in Table 5. To our surprise, we observe that all the models perform well in predicting intent and slot labels. The best performing model is BART (according to slot F1 score) with 400 million parameters, outperforming its smaller variant by 10.1% and 2.8% in terms of slot F1 for type-I and type-II slots, respectively.

Sequence Tagging vs. Seq2Seq Learning It is evident from the experiment results that Seq2Seq models outperform the sequence tagging models in slot filling by a large margin, while in intent classification, they are competitive. However, both the modeling approaches perform poorly in predicting all the slots in a sentence correctly, resulting in a lower EM score. One interesting factor is, the Seq2Seq models significantly outperform sequence tagging models in predicting type-II slots. Note that type-II slots are longer and less frequent, and we suspect conditional text generation helps Seq2Seq models predict them accurately. In comparison, we suspect that due to fewer labeled examples of type-II slots, the sequence tagging models perform poorly on that category (as noted before, we train the sequence tagging models for the type-I and type-II slots individually).

Next, we break down RoBERTa (w/ CRF) and BART's performances, the best performing models in their respective model categories, followed by an error analysis to shed light on the error types.

4.2 Performance Breakdown

Intent Classification In the PolicyIE corpus, 38% of the sentences fall into the first four categories: Data Collection, Data Sharing, Data Storage, Data Security, and the remaining belong to the Other category. Therefore, we investigate how much the models are confused in predicting the accurate intent label. We provide the confusion matrix of the models in Appendix. Due to an imbalanced distribution of labels, BART makes many

Intent labels	Intent F1	Slot F1	
		Type-I	Type-II
RoBERTa			
Data Collection	$74.1_{\pm 1.1}$	$59.8_{\pm 0.8}$	$28.9_{\pm 2.7}$
Data Sharing	$67.2_{\pm 2.0}$	$53.6_{\pm 5.7}$	$34.4_{\pm 3.4}$
Data Storage	$61.7_{\pm 3.6}$	$40.1_{\pm 3.7}$	$31.6_{\pm 3.1}$
Data Security	$68.9_{\pm 2.9}$	$53.9_{\pm 4.9}$	$21.9_{\pm 2.5}$
BART			
Data Collection	$73.5_{\pm 2.3}$	$67.0_{\pm 4.2}$	$56.2_{\pm 2.8}$
Data Sharing	$70.4_{\pm 2.7}$	$61.2_{\pm 1.6}$	$53.5_{\pm 3.4}$
Data Storage	$63.1_{\pm 4.7}$	$56.2_{\pm 8.2}$	$64.9_{\pm 2.5}$
Data Security	$67.2_{\pm 3.9}$	$66.0_{\pm 2.2}$	$32.8_{\pm 1.3}$

Table 6: Test performance of the RoBERTa and BART model for each intent type.

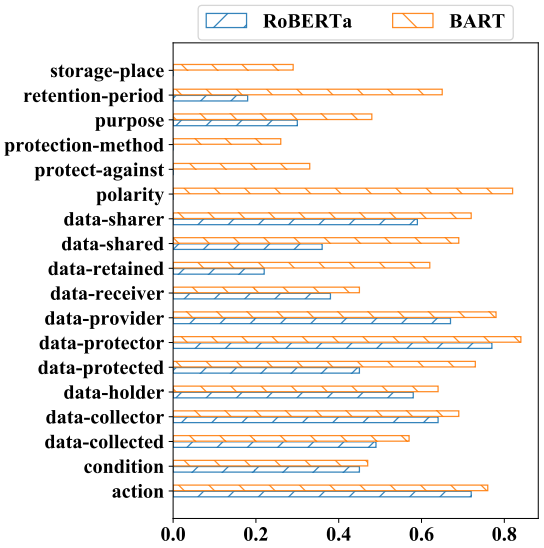

Figure 1: Test set performance (Recall score) on PolicyIE for the eighteen slot types.

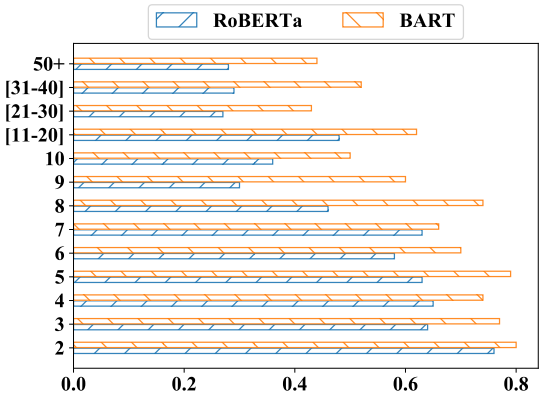

Figure 2: Test set performance (Recall score) on PolicyIE for slots with different length.

incorrect predictions. We notice that BART is confused most between *Data Collection* and *Data Storage* labels. Our manual analysis reveals that BART is confused between slot labels {"Data Collector", "Data Holder"} and {"Data Retained", "Data Collected"} as they are often associated with the same text span. We suspect this leads to BART's confusion. Table 6 presents the performance breakdown across intent labels.

Slot Filling We breakdown the models' performances in slot filling under two settings. First, Table 6 shows slot filling performance under different intent categories. Among the four classes, the models perform worst on slots associated with the "Data Security" intent class as PolicyIE has the lowest amount of annotations for that intent category. Second, we demonstrate the models' performances on different slot types in Figure 1. RoBERTa's recall score for "polarity", "protect-against", "protection-method" and "storage-place" slot types is zero. This is because these slot types have the lowest amount of training examples in PolicyIE. On the other hand, BART achieves a higher recall score, specially for the "polarity" label as their corresponding spans are short.

We also study the models' performances on slots of different lengths. The results show that BART outperforms RoBERTa by a larger margin on longer slots (see Figure 2), corroborating our hypothesis that conditional text generation results in more accurate predictions for longer spans.

4.3 Error Analysis

We analyze the incorrect intent and slot predictions by RoBERTa and BART. We categorize the errors into seven types. Note that a predicted slot is considered correct if its' label and span both match (*exact match*) one of the references. We characterize the error types as follows.

1. **Wrong Intent (WI)**: The predicted intent label does not match the reference intent label.
2. **Missing Slot (MS)**: None of the predicted slots *exactly* match a reference slot.
3. **Spurious Slot (SS)**: Label of a predicted slot does not match any of the references.
4. **Wrong Split (WSp)**: Two or more predicted slot spans with the same label could be merged to match one of the reference slots. A merged span and a reference span may *only* differ in punctuations or stopwords (e.g., and).
5. **Wrong Boundary (WB)**: A predicted slot span is a sub-string of the reference span or vice versa. The slot label must exactly match.

+ [IN:data-collection-usage [SL:data-provider.third-party-entity *third parties*] [SL:action collect] [SL:data-provider.user *your*] [SL:data-collected.data-general *information*] [SL:data-collector.first-party-entity *us*]] − [IN:data-sharing-disclosure [SL:data-receiver.third-party-entity *third parties*] [SL:action *share*] [SL:data-provider.user *your*] [SL:data-shared.data-general *information*] [SL:data-sharer.first-party-entity *us*] [SL:condition *where applicable*] [SL:condition *based on their own privacy policies*]] **Error types**: Wrong Intent (WI), Wrong Label (WL), Wrong Slot (WS), Spurious Slot (SS)
+ [. . . [SL:data-provider.third-party-entity *third parties*] [SL:condition *it is allowed by applicable law or according to your agreement with third parties*]] − [. . . [SL:condition *allowed by applicable law or according to your agreement with third parties*]] **Error types**: Wrong Boundary (WB), Missing Slot (MS)
+ [. . . [SL:data-receiver.third-party-entity *social media and other similar platforms*] . . .] − [. . . [SL:data-receiver.third-party-entity *social media*] [SL:data-receiver.third-party-entity *other similar platforms*] . . .] **Error types**: Wrong Split (WSp)

Table 7: Three examples showing different error types appeared in BART's predictions. + and − indicates the reference and predicted sequences, respectively. Best viewed in color.

Error	RoBERTa	BART
Wrong Intent	161	178
Spurious Slot	472	723
Missing Slot	867	517
Wrong Boundary	130	160
Wrong Slot	103	143
Wrong Split	32	27
Wrong Label	18	19
Total Slots	2,198	2,198
Correct Prediction	1,064	1,361
Total Errors	1,622	1,589
Total Predictions	2,686	2,950

Table 8: Counts for each error type on the test set of PolicyIE using RoBERTa and BART models.

6. **Wrong Label (WL)**: A predicted slot span matches a reference, but the label does not.
7. **Wrong Slot (WS)**: All other types of errors fall into this category.

We provide one example of each error type in Table 7. In Table 8, we present the counts for each error type made by RoBERTa and BART models. The two most frequent error types are SS and MS. While BART makes more SS errors, RoBERTa suffers from MS errors. While both the models are similar in terms of total errors, BART makes more correct predictions resulting in a higher Recall score, as discussed before. One possible way to reduce SS errors is by penalizing more on wrong slot label prediction than slot span. On the other hand, reducing MS errors is more challenging as many missing slots have fewer annotations than others. We provide more qualitative examples in Appendix (see Table 11 and 12) .

In the error analysis, we exclude the test examples (sentences) with the intent label "Other" and no slots. Out of 1,041 test instances in PolicyIE, there are 682 instances with the intent label "Other". We analyze RoBERTa and BART's predictions on those examples separately to check if the models predict slots as we consider them as spurious slots. While RoBERTa meets our expectation of performing highly accurate (correct prediction for 621 out of 682), BART also correctly predicts 594 out of 682 by precisely generating "[IN:Other]". Overall the error analysis aligns with our anticipation that the Seq2Seq modeling technique has promise and should be further explored in future works.

5 Related Work

Automated Privacy Policy Analysis Automating privacy policy analysis has drawn researchers' attention as it enables the users to know their rights and act accordingly. Therefore, significant research efforts have been devoted to understanding privacy policies. Earlier approaches (Costante et al., 2012) designed rule-based pattern matching techniques to extract specific types of information. Under the *Usable Privacy Project* (Sadeh et al., 2013), several works have been done (Bhatia and Breaux, 2015; Wilson et al., 2016a,b; Sathyendra et al., 2016; Bhatia et al., 2016; Hosseini et al., 2016; Mysore Sathyendra et al., 2017; Zimmeck et al., 2019; Bannihatti Kumar et al., 2020). No-

table works leveraging NLP techniques include text alignment (Liu et al., 2014; Ramanath et al., 2014), text classification (Wilson et al., 2016a; Harkous et al., 2018; Zimmeck et al., 2019), and question answering (QA) (Shvartzshanider et al., 2018; Harkous et al., 2018; Ravichander et al., 2019; Ahmad et al., 2020). Bokaie Hosseini et al. (2020) is the most closest to our work that used named entity recognition (NER) modeling technique to extract third party entities mentioned in policy documents.

Our proposed PolicyIE corpus is distinct from the previous privacy policies benchmarks: OPP-115 (Wilson et al., 2016a) uses a hierarchical annotation scheme to annotate text segments with a set of data practices and it has been used for multi-label classification (Wilson et al., 2016a; Harkous et al., 2018) and question answering (Harkous et al., 2018; Ahmad et al., 2020); PrivacyQA (Ravichander et al., 2019) frame the QA task as identifying a list of relevant sentences from policy documents. Recently, Bui et al. (2021) created a dataset by tagging documents from OPP-115 for privacy practices and uses NER models to extract them. In contrast, PolicyIE is developed by following semantic parsing benchmarks, and we model the task following the NLP literature.

Intent Classification and Slot Filling Voice assistants and chat-bots frame the task of natural language understanding via classifying intents and filling slots given user utterances. Several benchmarks have been proposed in literature covering several domains, and languages (Hemphill et al., 1990; Coucke et al., 2018; Gupta et al., 2018; Upadhyay et al., 2018; Schuster et al., 2019; Xu et al., 2020; Li et al., 2021). Our proposed PolicyIE corpus is a new addition to the literature within the security and privacy domain. PolicyIE enables us to build conversational solutions that users can interact with and learn about privacy policies.

6 Conclusion

This work aims to stimulate research on automating information extraction from privacy policies and reconcile it with users' understanding of their rights. We present PolicyIE, an intent classification and slot filling benchmark on privacy policies with two alternative neural approaches as baselines. We perform a thorough error analysis to shed light on the limitations of the two baseline approaches. We hope this contribution would call for research efforts in the specialized privacy domain from both privacy and NLP communities.

Acknowledgments

The authors acknowledge the law students Michael Rasmussen and Martyna Glaz at Fordham University who worked as annotators to make the development of this corpus possible. This work was supported in part by National Science Foundation Grant OAC 1920462. Any opinions, findings, conclusions, or recommendations expressed herein are those of the authors, and do not necessarily reflect those of the US Government or NSF.

Broader Impact

Privacy and data breaches have a significant impact on individuals. In general, security breaches expose the users to different risks such as financial loss (due to losing employment or business opportunities), physical risks to safety, and identity theft. Identity theft is among the most severe and fastest-growing crimes. However, the risks due to data breaches can be minimized if the users know their rights and how they can exercise them to protect their privacy. This requires the users to read the privacy policies of websites they visit or the mobile applications they use. As reading privacy policies is a tedious task, automating privacy policy analysis reduces the burden of users. Automating information extraction from privacy policies empowers users to be aware of their data collected and analyzed by service providers for different purposes. Service providers collect consumer data at a massive scale and often fail to protect them, resulting in data breaches that have led to increased attention towards data privacy and related risks. Reading privacy policies to understand users' rights can help take informed and timely decisions on safeguarding data privacy to mitigate the risks. Developing an automated solution to facilitate policy document analysis requires labeled examples, and the PolicyIE corpus adds a new dimension to the available datasets in the security and privacy domain. While PolicyIE enables us to train models to extract fine-grained information from privacy policies, the corpus can be coupled with other existing benchmarks to build a comprehensive system. For example, PrivacyQA corpus (Ravichander et al., 2019) combined with PolicyIE can facilitate building QA systems that can answer questions with fine-grained details. We believe our experiments and analysis will help direct future research.

References

Wasi Ahmad, Jianfeng Chi, Yuan Tian, and Kai-Wei Chang. 2020. PolicyQA: A reading comprehension dataset for privacy policies. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 743–749, Online. Association for Computational Linguistics.

Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596.

Vinayshekhar Bannihatti Kumar, Roger Iyengar, Namita Nisal, Yuanyuan Feng, Hana Habib, Peter Story, Sushain Cherivirala, Margaret Hagan, Lorrie Cranor, Shomir Wilson, et al. 2020. Finding a choice in a haystack: Automatic extraction of opt-out statements from privacy policy text. In *Proceedings of The Web Conference 2020*, pages 1943–1954.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Jianfeng Gao, Songhao Piao, Ming Zhou, et al. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *International Conference on Machine Learning*, pages 642–652. PMLR.

Jaspreet Bhatia and Travis D Breaux. 2015. Towards an information type lexicon for privacy policies. In *2015 IEEE eighth international workshop on requirements engineering and law (RELAW)*, pages 19–24. IEEE.

Jaspreet Bhatia, Morgan C Evans, Sudarshan Wadkar, and Travis D Breaux. 2016. Automated extraction of regulated information types using hyponymy relations. In *2016 IEEE 24th International Requirements Engineering Conference Workshops (REW)*, pages 19–25. IEEE.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Mitra Bokaie Hosseini, Pragyan K C, Irwin Reyes, and Serge Egelman. 2020. Identifying and classifying third-party entities in natural language privacy policies. In *Proceedings of the Second Workshop on Privacy in NLP*, pages 18–27, Online. Association for Computational Linguistics.

Duc Bui, Kang G Shin, Jong-Min Choi, and Junbum Shin. 2021. Automated extraction and presentation of data practices in privacy policies. *Proceedings on Privacy Enhancing Technologies*, 2021(2):88–110.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Federal Trade Commission et al. 2012. Protecting consumer privacy in an era of rapid change. *FTC report*.

Elisa Costante, Jerry den Hartog, and Milan Petković. 2012. What websites know about you. In *Data Privacy Management and Autonomous Spontaneous Security*, pages 146–159. Springer.

Ryan Cotterell and Kevin Duh. 2017. Low-resource named entity recognition with cross-lingual, character-level neural conditional random fields. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 91–96, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075.

Joshua Gluck, Florian Schaub, Amy Friedman, Hana Habib, Norman Sadeh, Lorrie Faith Cranor, and Yuvraj Agarwal. How short is too short? implications of length and framing on the effectiveness of privacy notices. In *Twelfth Symposium on Usable Privacy and Security ({SOUPS} 2016)*.

Abhirut Gupta, Anupama Ray, Gargi Dasgupta, Gautam Singh, Pooja Aggarwal, and Prateeti Mohapatra. 2018. Semantic parsing for technical support questions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3251–3259, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. 2018. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 531–548.

Charles T Hemphill, John J Godfrey, and George R Doddington. 1990. The atis spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*.

Mitra Bokaei Hosseini, Sudarshan Wadkar, Travis D Breaux, and Jianwei Niu. 2016. Lexical similarity of information type hypernyms, meronyms and synonyms in privacy policies. In *2016 AAAI Fall Symposium Series*.

Krippendorff Klaus. 1980. Content analysis: An introduction to its methodology.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. 2021. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2950–2962, Online. Association for Computational Linguistics.

Bing Liu and Ian Lane. 2016. Attention-based recurrent neural network models for joint intent detection and slot filling. In *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association*, pages 685–689.

Fei Liu, Rohan Ramanath, Norman Sadeh, and Noah A. Smith. 2014. A step towards usable privacy policy: Automatic alignment of privacy statements. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 884–894, Dublin, Ireland.

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

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Florencia Marotta-Wurgler. 2015. Does "notice and choice" disclosure regulation work? an empirical study of privacy policies,". In *Michigan Law: Law and Economics Workshop*.

Aleecia M McDonald and Lorrie Faith Cranor. 2008. The cost of reading privacy policies. *Isjlp*, 4:543.

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

Rohan Ramanath, Fei Liu, Norman Sadeh, and Noah A Smith. 2014. Unsupervised alignment of privacy policies using hidden markov models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 605–610.

Abhilasha Ravichander, Alan W Black, Shomir Wilson, Thomas Norton, and Norman Sadeh. 2019. Question answering for privacy policies: Combining computational and legal perspectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4947–4958.

Joel R Reidenberg, Jaspreet Bhatia, Travis D Breaux, and Thomas B Norton. 2016. Ambiguity in privacy policies and the impact of regulation. *The Journal of Legal Studies*, 45(S2):S163–S190.

Dennis Reidsma and Jean Carletta. 2008. Reliability measurement without limits. *Computational Linguistics*, 34(3):319–326.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. In *Proceedings of The Web Conference 2020*, pages 2962–2968.

Norman Sadeh, Alessandro Acquisti, Travis D Breaux, Lorrie Faith Cranor, Aleecia M McDonald, Joel R Reidenberg, Noah A Smith, Fei Liu, N Cameron Russell, Florian Schaub, et al. 2013. The usable privacy policy project. *Technical report, Technical Report, CMU-ISR-13-119*.

Kanthashree Mysore Sathyendra, Florian Schaub, Shomir Wilson, and Norman Sadeh. 2016. Automatic extraction of opt-out choices from privacy policies. In *2016 AAAI Fall Symposium Series*.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 3795–3805.

Yan Shvartzshanider, Ananth Balashankar, Thomas Wies, and Lakshminarayanan Subramanian. 2018. RECIPE: Applying open domain question answering to privacy policies. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 71–77, Melbourne, Australia. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107.

Milan Straka, Jan Hajic, and Jana Straková. 2016. Udpipe: trainable pipeline for processing conll-u files performing tokenization, morphological analysis, pos tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297.

Shyam Upadhyay, Manaal Faruqui, Gokhan Tür, Hakkani-Tür Dilek, and Larry Heck. 2018. (almost) zero-shot cross-lingual spoken language understanding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6034–6038. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*.

Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N. Cameron Russell, Thomas B. Norton, Eduard Hovy, Joel Reidenberg, and Norman Sadeh. 2016a. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340.

Shomir Wilson, Florian Schaub, Rohan Ramanath, Norman Sadeh, Fei Liu, Noah A Smith, and Frederick Liu. 2016b. Crowdsourcing annotations for websites' privacy policies: Can it really work? In *Proceedings of the 25th International Conference on World Wide Web*, pages 133–143. International World Wide Web Conferences Steering Committee.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Weijia Xu, Batool Haider, and Saab Mansour. 2020. End-to-end slot alignment and recognition for cross-lingual NLU. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5052–5063, Online. Association for Computational Linguistics.

Xiaodong Zhang and Houfeng Wang. 2016. A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, volume 16, pages 2993–2999.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. 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)*, pages 1127–1137, Beijing, China. Association for Computational Linguistics.

Qile Zhu, Haidar Khan, Saleh Soltan, Stephen Rawls, and Wael Hamza. 2020. Don't parse, insert: Multilingual semantic parsing with insertion based decoding. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 496–506, Online. Association for Computational Linguistics.

Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N Cameron Russell, and Norman Sadeh. 2019. Maps: Scaling privacy compliance analysis to a million apps. *Proceedings on Privacy Enhancing Technologies*, 2019(3):66–86.

Type-I slots	Attributes
Action	None
Data Provider	(1) User (2) Third party entity
Data Collector	(1) First party entity
Data Collected	(1) General Data (2) Aggregated/Non-identifiable data (3) Contact data (4) Financial data (5) Location data (6) Demographic data (7) Cookies, web beacons and other technologies (8) Computer/Device data (9) User online activities/profiles (10) Other data
Data Sharer	(1) First party entity
Data Shared	(1) General Data (2) Aggregated/Non-identifiable data (3) Contact data (4) Financial data (5) Location data (6) Demographic data (7) Cookies, web beacons and other technologies (8) Computer/Device data (9) User online activities/profiles (10) Other data
Data Receiver	(1) Third party entity
Data Holder	(1) First party entity (2) Third party entity
Data Retained	(1) General Data (2) Aggregated/Non-identifiable data (3) Contact data (4) Financial data (5) Location data (6) Demographic data (7) Cookies, web beacons and other technologies (8) Computer/Device data (9) User online activities/profiles (10) Other data
Storage Place	None
Retention Period	None
Data Protector	(1) First party entity (2) Third party entity
Data Protected	(1) General Data (2) Aggregated/Non-identifiable data (3) Contact data (4) Financial data (5) Location data (6) Demographic data (7) Cookies, web beacons and other technologies (8) Computer/Device data (9) User online activities/profiles (10) Other data
Protect Against	Security threat
Type-II slots	Attributes
Purpose	(1) Basic service/feature (2) Advertising/Marketing (3) Legal requirement (4) Service operation and security (5) Personalization/customization (6) Analytics/research (7) Communications (8 Merge/Acquisition (9) Other purpose
Condition	None
Polarity	(1) Negation
Protection Method	(1) General safeguard method (2) User authentication (3) Access limitation (5) Encryptions (6) Other protection method

Table 9: Slots and their associated attributes. “None” indicates there are no attributes for the those slots.

Privacy Practices	Data Collection/Usage	Data Sharing/Disclosure	Data Storage/Retention	Data Security/Protection
Type-I slots				
Action	750 / 169	344 / 70	198 / 57	102 / 31
Data Provider	784 / 172	247 / 54	139 / 44	65 / 20
Data Collector	653 / 151	-	-	-
Data Collected	1833 / 361	-	-	-
Data Sharer	-	288 / 54	-	-
Data Shared	-	541 / 110	-	-
Data Receiver	-	456 / 115	-	-
Data Holder	-	-	192 / 59	-
Data Retained	-	-	291 / 119	-
Storage Place	-	-	70 / 21	-
Retention Period	-	-	101 / 17	-
Data Protector	-	-	-	105 / 31
Data Protected	-	-	-	119 / 34
Protect Against	-	-	-	49 / 15
Type-II slots				
Purpose	894 / 193	327 / 65	168 / 40	5 / 0
Condition	337 / 81	154 / 26	81 / 25	43 / 7
Polarity	50 / 15	21 / 1	22 / 1	18 / 5
Protection Method	-	-	-	143 / 35
# of slots	5301 / 1142	2378 / 495	1262 / 383	649 / 178
# of sequences	919 / 186	380 / 83	232 / 61	103 / 29

Table 10: Privacy practices and the associated slots with their distributions. "X / Y" indicates there are X instances in the train set and Y instances in the test set.

Figure 3: Confusion matrix for intent classification using the RoBERTa model.

Figure 4: Confusion matrix for intent classification using the BART model.

		Label	Text
Ground truth		data-holder.first-party-entity	We
		action	keep
		data-retained.data-general	records
		retention-period.retention-period	a period of no more than 6 years
RoBERTa (P:1.0, R: 0.75)	✓	data-holder.first-party-entity	We
	✓	action	keep
	✓	retention-period.retention-period	a period of no more than 6 years
BART (P:1.0, R: 1.0)	✓	data-holder.first-party-entity	We
	✓	action	keep
	✓	data-retained.data-general	records
	✓	retention-period.retention-period	a period of no more than 6 years
Ground truth		data-collector.first-party-entity	We
		action	access
		data-collected.data-general	information
RoBERTa (P:0.0, R: 0.0)	✗	data-sharer.first-party-entity	We
	✗	data-shared.data-general	information
BART (P:0.0, R: 0.0)	✗	data-sharer.first-party-entity	We
	✗	action	disclose
	✗	data-shared.data-general	information
Ground truth		data-sharer.first-party-entity	Marco Polo
		data-receiver.third-party-entity	third party
		data-shared.data-general	Personal Information
		data-provider.user	users
		action	transferred
RoBERTa (P:0.6, R: 0.6)	✗	data-receiver.third-party-entity	Marco
	✗	data-sharer.first-party-entity	our
	✓	data-receiver.third-party-entity	third party
	✓	data-shared.data-general	Personal Information
	✓	action	transferred
BART (P:0.83, R: 1.0)	✓	data-sharer.first-party-entity	Marco Polo
	✓	data-receiver.third-party-entity	third party
	✓	data-shared.data-general	Personal Information
	✗	data-sharer.first-party-entity	us
	✓	data-provider.user	users
	✓	action	transferred
Ground truth		data-sharer.first-party-entity	We
		data-receiver.third-party-entity	third parties
		action	provide
		data-shared.data-general	information
RoBERTa (P:1.0, R: 1.0)	✓	data-sharer.first-party-entity	We
	✓	data-receiver.third-party-entity	third parties
	✓	action	provide
	✓	data-shared.data-general	information
BART (P:0.25, R: 0.25)	✗	data-collector.first-party-entity	We
	✗	data-provider.third-party-entity	third parties
	✓	action	provide
	✗	data-collected.data-general	information

Table 11: Sample RoBERTa and BART predictions of Type-I slots. (✓) and (✗) indicates correct and incorrect predictions, respectively. Precision (P) and recall (R) score is reported for each example in the left column.

Ground truth		[Label] condition [Text] you use our product and service or view the content provided by us
RoBERTa (P:1.0, R: 1.0)	✓	[Label] condition [Text] you use our product and service or view the content provided by us
BART (P:1.0, R: 1.0)	✓	[Label] condition [Text] you use our product and service or view the content provided by us
Ground truth		[Label] purpose.other [Text] their own purposes
		[Label] purpose.advertising-marketing [Text] inform advertising related services provided to other clients
RoBERTa (P:0.0, R: 0.0)	✗	[Label] None [Text] None
BART (P:1.0, R: 1.0)	✓	[Label] purpose.other [Text] their own purposes
	✓	[Label] purpose.advertising-marketing [Text] inform advertising related services provided to other clients
Ground truth		[Label] purpose.personalization-customization [Text] provide more tailored services and user experiences
		[Label] purpose.basic-service-feature [Text] remembering your account identity
		[Label] purpose.service-operation-and-security [Text] analyzing your account 's security
		[Label] purpose.analytics-research [Text] analyzing your usage of our product and service
		[Label] purpose.advertising-marketing [Text] advertisement optimization (helping us to provide you with more targeted advertisements instead of general advertisements based on your information)
RoBERTa (P:0.17, R: 0.2)	✗	[Label] purpose.basic-service-feature [Text] provide
	✗	[Label] purpose.other [Text] purposes
	✗	[Label] purpose.analytics-research [Text] remembering your account identity
	✗	[Label] purpose.analytics-research [Text] analyzing your account 's security
	✓	[Label] purpose.analytics-research [Text] analyzing your usage of our product and service
	✗	[Label] purpose.advertising-marketing [Text] advertisement optimization
BART (P:0.43, R: 0.6)	✓	[Label] purpose.personalization-customization [Text] provide more tailored services and user experiences
	✗	[Label] purpose.service-operation-and-security [Text] remembering your account identity
	✓	[Label] purpose.service-operation-and-security [Text] analyzing your account 's security
	✓	[Label] purpose.analytics-research [Text] analyzing your usage of our product and service
	✗	[Label] purpose.advertising-marketing [Text] advertisement optimization
	✗	[Label] purpose.advertising-marketing [Text] provide you with more targeted advertisements instead of general advertisements
	✗	[Label] purpose.advertising-marketing [Text] based on your information

Table 12: Sample RoBERTa and BART predictions of Type-II slots. (✓) and (✗) indicates correct and incorrect predictions, respectively. Precision (P) and recall (R) score is reported for each example in the left column.