Identifying and Classifying Third-party Entities in Natural Language Privacy Policies

Mitra Bokaei Hosseini & *PragyanKC*
St. Mary's University / 1 Camino Santa Maria, San Antonio, TX 78228
& *IrwinReyes*
Two Six Labs, LLC / 901 N. Stuart St., Arlington, VA 22203
& *SergeEgelman*
International Computer Science Institute / 1947 Center St, Berkeley, CA 94704

Abstract

App developers often raise revenue by contracting with third party ad networks, which serve targeted ads to end-users. To this end, a free app may collect data about its users and share it with advertising companies for targeting purposes. Regulations such as General Data Protection Regulation (GDPR) require transparency with respect to the recipients (or categories of recipients) of user data. These regulations call for app developers to have privacy policies that disclose those third party recipients of user data. Privacy policies provide users transparency into what data an app will access, collect, shared, and retain. Given the size of app marketplaces, verifying compliance with such regulations is a tedious task. This paper aims to develop an automated approach to extract and categorize third party data recipients (i.e., entities) declared in privacy policies. We analyze 100 privacy policies associated with most downloaded apps in the Google Play Store. We crowdsource the collection and annotation of app privacy policies to establish the ground truth with respect to third party entities. From this, we train various models to extract third party entities automatically. Our best model achieves average F1 score of 66% when compared to crowdsourced annotations.

1 Introduction

According to statistics from Google Play Store [1] and App Store [2] there are about 2.2 and 2.8 million applications (apps) in various categories, available for download on each platform, respectively. To generate revenue from targeted advertising, apps often collect user data such as real-time location, financial information, friends list, photos, and contact information, among others. Apps inform users of this activity through privacy policies, which detail the kinds of personal data being collected, how the data is being used, and with whom the data is shared. Consumer surveys reveal that over 90% of users accept legal terms and conditions without reading them Deloitte (2017). Moreover, statistics from a younger group with ages 18 to 34, showed that more than 97% of users never read the privacy policy and were unaware of the fact that their data has been shared to various third party companies Deloitte (2017). Such challenges arise due to the following facts. First, privacy policies are too complex and long-winded McDonald and Cranor (2008), and consumers lack the bandwidth to read and comprehend each and every privacy policy for all the services they use. Second, the asymmetric power between businesses and consumers yields "take-it-or-leave-it" attitude forced by businesses Acquisti (2010), where a consumer who doesn't like the terms being offered, is likely to be dismissed because the loss will be negligible.

Recent incidents highlight the importance of user awareness concerning third-party data collection in mobile apps. In August 2020, following a lawsuit brought about by the Los Angeles City Attorney's Office, The Weather Channel agreed to give users control over how their location data would be shared [3]. The lawsuit alleged The Weather Channel app improperly shared location data with third parties, when consumers only con-

[1] play.google.com
[2] https://www.apple.com/ios/app-store/
[3] https://apnews.com/f6a83c0b8e0a65563e4c76955c37c0ab

Proceedings of the Second Workshop on Privacy in Natural Language Processing, pages 18–27

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

sented to using it to find local weather reports. Similarly, the US Federal Trade Commission settled with children's app developer HyperBeard Inc. for allowing advertisers to collect tracking information from young users, in violation of the Children's Online Privacy Protection Act [4].

Recent regulatory frameworks, such as General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA), require app development companies to be more concrete on the data recipients listed in the privacy policies and disclose the categories of third parties with whom the app may share personally identifiable data. Such third parties, include affiliates, advertising networks, and analytics providers.

Previous works have proposed machine learning models to extract categories of data practices, such as data collection, usage, and sharing statements from privacy policies Zimmeck et al. (2017); Harkous et al. (2018). These models utilize natural language processing techniques to identify relevant features for automated extraction. However, these models fail to identify and categorize third party entities in sharing data practices of mobile apps. In this paper, we aim to analyze sharing data practices in privacy policies of mobile apps to develop an automated approach to extract and categorize third party entities. The third party entities can fall into two main categories: (1) generic third party entities, such as advertisers, partners, and affiliates; and (2) specific third party entities, such as Google Analytics, Facebook, and PayPal. For this reason, we first collect privacy policies of 100 mobile apps from Spain, which were associated with 485 most downloaded apps in Google Play. We use crowdsourcing to annotate and extract the third party categories and names from these policies. These annotations are used to train and evaluate implementations of Stanford CoreNLP CRF, scikit-learn CRF, Bi-LSTM, and Bi-LSTM CRF as named entity extraction (NER) models. Through this research, we study the feasibility of automatically extracting third party entities from privacy policy text using the crowdsourced annotated data.

To summarize, the contributions of this paper are two fold: (1) empirical evaluation of a pre-trained model to extract third party entities from privacy policies; (2) training and evaluation of various models to extract and categorize third party entities using the annotated data from 100 privacy policies.

Our results indicate poor performance of the pre-trained NER model on privacy policy corpus and the importance of training NER models for domain-specific purposes. Further, our experiment results suggest that natural language features in Stanford CoreNLP CRF model provides the best performance for extracting and categorizing third party entities. This model achieves F1 scores of 79% and 53% for generic and specific third party extraction, respectively.

This paper is organized as follows. In § 2 and § 3, we discuss background and related work. In § 4 we introduce our method in text analysis of privacy policies. In § 5, we present the evaluation and results, followed by discussion, threats to validity, and concluding remarks in § 6, § 7, and § 8.

2 Background

Named Entity Recognition (NER) The first step in most natural language processing tasks is to detect and classify all the proper names mentioned in natural language text – a task generally referred to as named entity recognition (NER) Jurafsky and Martin (2008). Generic NER models tend to focus on finding the names of people, places and organizations that are mentioned in *ordinary news texts*. In this paper, we utilize three different NER models, including Conditional Random Field, Bi-LSTM, and Bi-LSTM with CRF.

Word Embedding Distributed representation of a word as a vector in some m-dimensional space that helps learning algorithms achieve better performance by grouping similar words together Mikolov et al. (2013); Bengio et al. (2003). Each vector dimension represents some feature of the words' semantics in a corpus. In this paper, we adopt Word2Vec [5], an implementation of the Skip-gram model to construct domain-specific word embeddings, which is discussed in § 4.4.

3 Related Work

Researchers have proposed methods to ensure data transparency and compliance by analyzing data

[4] https://www.ftc.gov/news-events/press-releases/2020/06/developer-apps-popular-children-agrees-settle-ftc-allegations-it

[5] https://code.google.com/archive/p/word2vec/

practices expressed in privacy policies. For example, Breaux et al. formalized data practice requirements from privacy policies using Description Logic to automatically detect conflicting requirements and to trace data flows across policies of interacting services Breaux et al. (2009, 2014). This method relies on a small set of manually annotated privacy policies to instantiate the language. Further, initial attempts in automated analysis of privacy policy text have largely focused on readability scores Massey et al. (2013); Meiselwitz (2013); Ermakova et al. (2015).

Prior to introduction of GDPR, studies have focused on identifying policy sections related to different data practices (e.g., collection, use, and sharing) using machine learning models trained on web-based apps' privacy policies Ramanath et al. (2014); Liu et al. (2018); Wilson et al. (2016); Zimmeck et al. (2017); Harkous et al. (2018). In these studies, each segment of the privacy policy is mapped to data types from a pre-defined set of keywords, including "location", "contact", or "identifier." These studies fail to automatically extract the exact data types from the segments and identify data types that don't exist in the pre-defined set. Further, approaches that map data types to a small number of broad categories introduce inaccuracies because the mapping is based on satisficing where subtle differences between the meanings of two or more types are ignored **?**. For example, the phrase "WiFi SSID" can be construed to mean a type of location information Zimmeck et al. (2017) when location is one of a few categories to choose from (i.e., the categorization fits given the constraints). However, the term actually describes a technology that must be combined with the SSID's device information (e.g., MAC address) before it can be indirectly correlated with location; it should not be directly related to location on its own.

GDPR requires the data controllers to disclose, among others, the data recipients' categories or names (while the WP29 suggests that "the default position is that a data controller should provide information on the actual (named) recipients of the personal data"). Since 2018, majority of the privacy polices introduced updates in their terms and changes to their structure to be compliant with the new regulations. Therefore, we focus on creation of a corpus containing recently published privacy policies for our analysis.

Some studies specifically focused on automatically classifying web-based privacy policy sentences into various GDPR requirements, including "opt outs" Sathyendra et al. (2017); Tesfay et al. (2018). Researchers also utilized string searching techniques to find third-party domain names in privacy policy text Libert (2018). In contrast to these studies, our goal is to automatically extract any mentions of third party entities, including generic third party names (e.g., third parties, advertisers, affiliates, analytics services) and specific third party entities (e.g., Google Analytics, Amazon Web Services, Facebook, Paypal) from mobile apps privacy policies.

4 Privacy Policy Analysis

In this section, we first discuss our approach for constructing a corpus of 100 privacy policies. This corpus is used to train and evaluate various models to extract and classify third party entities. Second, we introduce a pilot study on five privacy policies selected from the corpus. Through the pilot study, we provide evidence on why *pre-trained* named entity recognition (NER) models on news corpus, such as Stanford CoreNLP CRF [6] cannot provide desirable results. Third, we describe our crowdsourcing approach to annotate the privacy policy corpus. Finally, we utilize the annotations toward training and evaluation of various NER models.

4.1 Privacy Policy Corpus

Our corpus of privacy policies was drawn from the "Privacy Policy" links available on Google Play Store app pages. In **June 2018**, right after the General Data Protection Regulation (GDPR) (i.e., the new privacy law for the European Union (EU)) was put into effect, we crawled the "Top Free" charts in each of the Play Store's categories to find privacy policies. We specifically targeted a European country as the location to download the privacy polices. As a result, we downloaded **100 privacy policies** associated with 485 most downloaded apps.

We used Selenium-automated headless Firefox instances to save the contents of those privacy policy links. Privacy policies were scraped by saving the HTML content of the links after they finished loading. HTML privacy policies were then preprocessed using BeautifulSoup to do a best-effort

[6] https://stanfordnlp.github.io/CoreNLP/ner.html

extraction of the policy body in plaintext. All subsequent labeling and analysis was performed on these plaintext privacy policies.

4.2 Pilot Study

Named Entity Recognition (NER) models label sequences of words in a text which are the names of things, such as person and company names. These models can be used to extract third party entities from privacy policies. For this pilot study, we utilize implementation of Conditional Random Field (CRF) model in Stanford CoreNLP. This model is trained on an annotated newswire dataset (CoNLL-2003).

To analyze if this pre-trained model can extract generic and specific third party entities from privacy policies, we randomly select five privacy policies from the privacy policy corpus. To create a gold standard on the set of five polices and identify the third party generic and specific names, we setup an annotation task on Amazon Mechanical Turk (AMT). Three privacy experts (including the first and third authors) annotated the privacy policies based on a coding frame discussed below. Finally, we compare the extracted generic and specific third party entities with the extracted results by applying pre-trained CRF model implementation in Stanford CoreNLP. Next, we discuss our annotation task setup to construct the gold standard, followed by analysis of the pilot study results.

To set up the annotation task, we first itemize selected privacy policies into paragraphs of $\sim$120 words, yielding 151 crowd worker annotation tasks and publish the tasks as Human Intelligence Tasks (HITs) on AMT. We preserve larger spans if the statements contain an anaphoric reference to a previous sentence (e.g., "we share your information with third party entities. These entities include...") or when the statement contains subparts or bullet points that depend on the context provided by the earlier sentences.

Next, we ask annotators to annotate the itemized privacy policy segments in each HIT based on the coding frame presented in Figure 1. The coding frame for this annotation task consist of two codes: *Generic Third Parties*, which we define as "categories or types of third parties with whom the application may share information with or use their services, e.g., third-party partners, affiliates, advertisers, analytics services;" and *Specific Third Parties*, which we define as "specific names of organizations that the application may share information with or use their services, such as Google Analytics, Amazon Web Services, Facebook, Paypal." Some paragraphs do not contain any third party entities, therefore, annotators were instructed to select a text box indicating "The paragraph does not contain any third-party categories or names." Finally, we compile the annotations where **two or more annotators** agreed on. We cannot use inter-rater reliability to evaluate consensus for phrase-level coding because the non-coded words dominate coded words and annotators generally agree about which phrases not to code Breaux and Schaub (2014). Figure 2 illustrates a HIT which is annotated using the codes *Generic Third Parties* and *Specific Third Parties*.

Table 1: Pilot Study Results

Model	Precision	Recall	F1
CoreNLP CRF	0.46	0.17	0.24

Through this pilot study, experts annotated a total of 148 unique third parties, including 117 and 31 generic and specific unique third party entities, respectively. To extract third party entities using a pre-trained CRF model, we applied Stanford CoreNLP implementation of CRF on itemized segments of five privacy policies. We compared the extracted entities with the experts annotations as our gold standard. Through this comparison, we calculate precision, recall, and F1 performance measures using true positives (TPs), false positives (FPs), false negatives (FNs), and Equations 1, 2, and 3. An extracted entity is a TP, if it is annotated as either generic or specific third party by the experts, otherwise the extracted entity is a FP. If an annotated generic or specific third party cannot be found in the list of extracted entities by the pre-trained CRF model, we consider the annotation as FN. Table 1 shows precision, recall and F1 score for our pilot study.

$$precision = TP/(TP + FP) \quad (1)$$

$$recall = TP/(TP + FN) \quad (2)$$

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \quad (3)$$

Based on the results, the pre-trained NER model on the news corpus cannot be effectively used on privacy policy domain. Therefore, there is a need to acquire a privacy policy corpus that contains annotations for generic and specific third

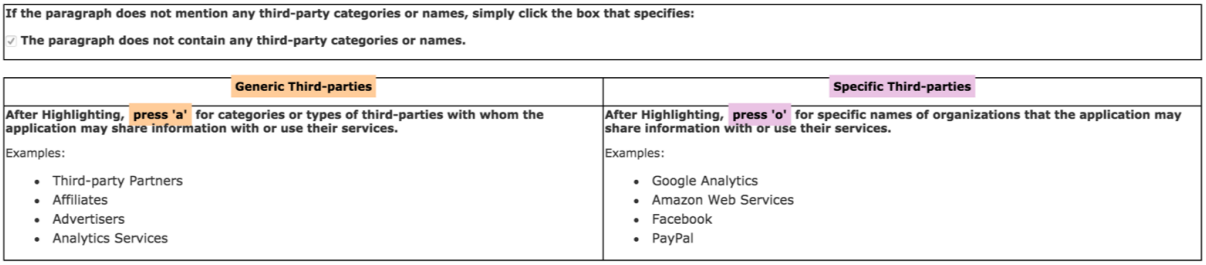

Figure 1: Third Party Annotation Coding Frame

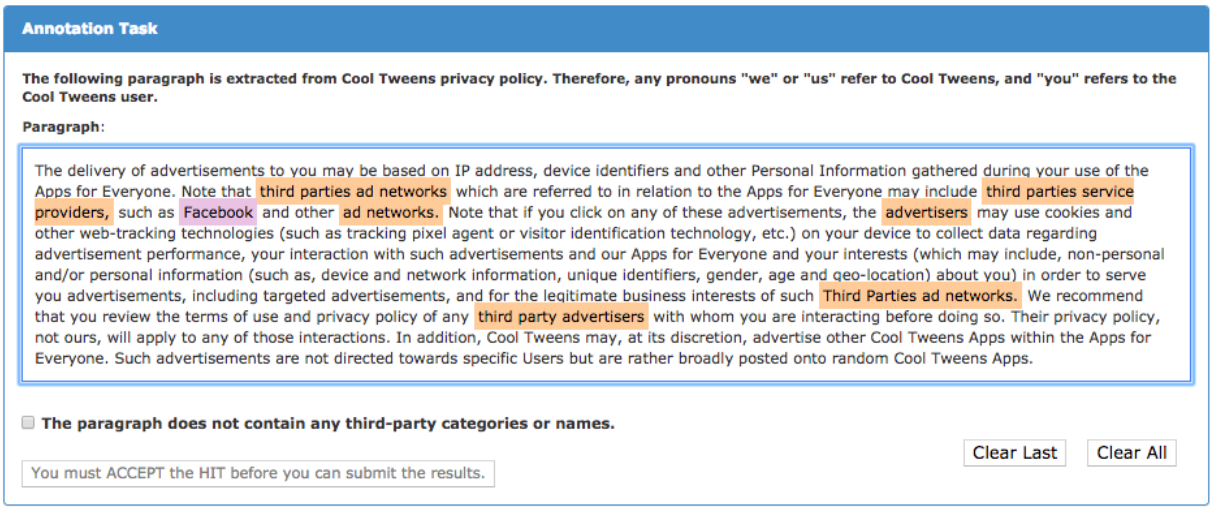

Figure 2: Annotation Example for a Privacy Policy Paragraph

party entities. This annotated corpus can be used to train various NER models. Next, we describe our efforts to construct such annotated corpus, which is then used to train and evaluate NER models.

4.3 Annotating the Corpus

We published paragraphs from 100 privacy policies on AMT and recruited workers to annotate the paragraphs based on the coding frame and instructions in Figure 1.

To recruit qualified workers for annotation, we created a qualification test on AMT. The successful workers where then granted access to annotate the privacy policies. For the qualification test, the workers were required to have HIT approval rate greater than 95 % and locate in the U.S. The qualification test HIT contains three privacy policy paragraphs that were randomly selected from the pilot study (see Section 4.2). Therefore, we had the gold standard for the annotations in the selected paragraphs. From the selected three paragraphs, one did not contain any third party entities. We used the instructions showed in Figure 1 for the qualification test. We hired 500 workers to take the qualification test. We manually analyzed the results of 500 HITs and compared the annotations with our gold standard. This analysis results in disqualifying 182 annotators due to combination of reasons, such as missing entities compared to the gold standard, marking a paragraph as not containing third parties by mistake, annotating first-party entities, annotations that spanned over English pronouns, and annotating information types.

To construct the annotated corpus, we invited all qualified workers for annotating 100 privacy polices. We follow the approach mentioned in § 4.2 to itemize the privacy policies to individual paragraphs, yielding 2,816 crowd worker annotator tasks. To achieve consensus on these tasks, each task needs to be completed by three workers. Workers are instructed to use coding frame as mentioned in Figure 1. Finally, we compile the annotations where **two or more annotators** agreed on and itemized the annotations Bhatia and Breaux (2015). The annotations resulted in 1,391 unique third party entities, including 948 generic categories and 443 specific (i.e., organization) names.

Train and Test Split Our goal is to train NER models using our annotated corpus. Therefore, we

randomly split the annotated policies into training ($n = 70$) and test ($n = 30$) sets. We use the annotated policies in the training set for developing our NER models. The test set is set aside to prevent over-fitting.

4.4 Automated Named Entity Extraction

We characterize the detection and classification of third-party entities as named entity recognition (NER) problem.

Named Entity Recognition Third party entities are categorized as either (1)*generic third party*: general types of third-parties with whom an app may share information with or use their services (e.g., partners, affiliates, advertisers, etc.); or (2) *specific third party* names of organizations that the app may share information with or use their services (e.g., Google Analytics, Amazon Web Services, Facebook, etc.). To automatically extract and categorize these entities, we train various NER models, including Conditional Random Field (CRF), Bi-LSTM, and Bi-LSTM with CRF.

Preprocessing We utilize Stanford CoreNLP to tokenize sentences in the annotated privacy policy corpus. Further, we identify the part-of-speech (POS) tags for each token. As a result, each token is associated with a POS tag, an annotation tag (i.e, 'g': **generic**, 's': **specific**, 'o': **none**), and a sentence number. The preprocessed format of the privacy policy corpus is used to train and evaluate NER models as discussed below.

Vectorization (Word Embedding) Prior to training NER models using tokenized annotated corpus, we generate vector representation of each word in our corpus. For this reason, we use two methods to create vector representation for words (i.e., word embeddings): (1) Bag-of-Words (BoW); and (2) Word2Vec.

For BoW, every word in our corpus is assigned to a unique number. Sentences in the corpus are encoded with a sequence of numbers representing the words. The sequences are then passed to Bi-LSTM and Bi-LSTM CRF models that we discuss later.

To create domain-specific word embeddings, we followed the approach by Harkous et al. Harkous et al. (2018) and trained the Word2Vec model using 77,556 English privacy policies collected from mobile applications on the Google Play Store [7]. As a result, every word in our privacy corpus can be presented using 200-dimension embedding. Common-purpose word embeddings trained on the English Wikipedia dump Pennington et al. (2014); Bojanowski et al. (2017); Ling et al. (2015) or Google News dataset Mikolov et al. (2013) exist, however, previous research has shown improvements on classification accuracy by utilizing domain-specific word embeddings.

To train Word2Vec method using privacy policies, we crawled the metadata archive for more than 1,402,894 Android apps provided by the PlayDrone project Viennot et al. (2014) from which 109,933 contained a valid link to a privacy policy. We used the BeautifulSoup library in Python to extract the text from the HTML files by stripping HTML tags associated with: head, script, URL, navigation, button, and option information. Next, we filtered non-English policy text files, yielding 77,556 privacy policies with the majority of text in English by using the DetectLang library in Python. In the next step, for each privacy policy, we tokenized the sentences and removed all non-English sentences. We also expanded the contractions (e.g., "won't" is transformed to "will not"), and removed punctuation, numbers, email addresses, URLs, and special characters. Finally, we transformed the remaining characters to lower-case. The resulting pre-processed text was used to train the Word2Vec Mikolov et al. (2013) model. We utilize embedding vectors to replace the words in our privacy corpus. Therefore, a sentence is represented with a sequence of 200-dimension vectors. We use this embedding in Bi-LSTM and Bi-LSTM CRF models.

Training Using Stanford CoreNLP, scikit-learn, TensorFlow, and Keras, we train three NER models, including CRF, Bi-LSTM, and Bi-LSTM with CRF. We utilize CRF implementation from both Stanford CoreNLP [8] and scikit-learn and train these models using our training set. In addition to POS tags as a pre-extracted feature, both implementations of CRF model extract additional **natural language (NL) features**, including word parts, simplified POS tags, lower/title/upper flags, features of nearby words.

To train Bi-LSTM NER model, we implement the model using two word embedding representations. The first implementation utilizes the word

[7] http://play.google.com
[8] https://nlp.stanford.edu/software

Table 2: Performance Results

CoreNLP CRF	Prec.	Rec.	F1
Generic	0.83	0.76	**0.79**
Specific	0.60	0.47	**0.53**
scikit-learn CRF	Prec.	Rec.	F1
Generic	0.65	0.38	0.48
Specific	0.61	0.18	0.28
Bi-LSTM with BoW	Prec.	Rec.	F1
Generic	0.63	0.49	0.55
Specific	0.54	0.20	0.30
Bi-LSTM with W2V	Prec.	Rec.	F1
Generic	0.65	0.43	0.52
Specific	0.42	0.30	0.35
Bi-LSTM+CRF with BoW	Prec.	Rec.	F1
Generic	0.63	0.49	0.55
Specific	0.41	0.26	0.33
Bi-LSTM+CRF with W2V	Prec.	Rec.	F1
Generic	0.61	0.53	0.57
Specific	0.51	0.25	0.33

embeddings generated through BoW, and the second implementation incorporates word embeddings generated through Word2Vec method. Both implementations of Bi-LSTM NER are trained using our training set with 10 epochs, batch sizes of 32, and categorical cross-entropy as loss function. These parameters perform the best on the training set. As our third model, we train Bi-LSTM with CRF layer considering two implementations of the model using BoW and Word2Vec. This model utilizes an additional CRF layer for sentence sequencing on top of Bi-LSTM.

5 Evaluation and Results

We view our work as an exploration of extracting and classifying third party entities from privacy policies. As mentioned in § 4.4, we implement six NER models that utilize natural language (NL) features, Bag-of-Words (BoW), and Word2Vec (W2V). Using the train and test split of 100 privacy policies in our corpus, we train and evaluate each model. Each token in the annotated corpus is tagged with one of three labels: **generic**, **specific**, and **none**. The criteria for this selection is described in § 4.4. Table 2 shows the results of our experiments on testing set containing 30 privacy policies. Our results suggest that the implementation of CRF in Stanford CoreNLP performs the best when compared to other models and implementations. Therefore, the NL features used in Stanford CoreNLP implementation improve the NER performance. Based on these results, representing privacy policy words using domain specific context vectors learned from W2V slightly improves F1 score. However, the NL features in CRF model still outperform Bi-LSTM models that use domain specific word vector representations.

6 Discussion

We start our discussion by analyzing results from our pilot study. Through this study, we compared the experts annotations with automated third party entities extracted using a pre-trained CRF model. However, low precision, recall, and F1 score suggest poor performance of this pre-trained model for our task. The CRF model used in our pilot study was trained on news wire corpus. The tags in this pre-trained model also does not satisfy our NER task requirements. Using the pre-defined tags, we are only able to classify third party entities in privacy policies using a label called organizations (ORG) in the CRF model implementation. Therefore, for calculating performance measures in pilot study, we defined true positives (TPs) as an automatically extracted entity that is annotated as either generic or specific by our experts. Through further analysis, the pre-trained CRF model can only extract eight unique generic third party entities, and 18 unique specific third party entities.

The results from our pilot study justifies the need for training NER models using privacy policy corpus. Therefore, through this study, we construct an annotated privacy policy corpus that contains third party entities tagged with generic and specific labels. The annotations is the result of hiring qualified workers (i.e., required to successfully pass a qualification test) on AMT. Finally, the annotated corpus is used to train and test NER models. Our experiment results suggest that the natural language (NL) features used in CRF implementation of Stanford CoreNLP produces the best results when compared to our gold standard in the test set. This model considers combinations of NL features, in-

Table 3: Distribution of Tagged Tokens in Annotated Corpus

#Tokens Tagged as Generic	#Tokens Tagged as Specific	#Tokens Tagged as None	Total
9,191	2,241	422,818	434,250

cluding current word, previous word, next word, word character ngrams (up to length six), current and surrounding POS tag sequences, additional word shape features, and presence of word in left an right windows (size 4) [9] Finkel et al. (2005).

Finally, we believe the results can be improved by increasing the size of our annotated corpus. To support this claim, we present the distribution of tagged tokens in our annotated corpus in Table 3. Based on this table, our privacy policy corpus is mainly populated with non-third party tokens and is skewed toward tokens labeled with none. As another solution to this skewed annotated corpus, we foresee to apply downsampling techniques on privacy policy segments that do not contain third party entities.

7 Threats to Validity

Construct Validity Construct validity reflects whether the measurements actually measure that which they are intended to measure Yin (2017). In this paper, we provide definitions for "generic third party" and "specific third party." These definitions are reviewed by a legal expert on our team.

Internal Validity Internal validity is the extent to which a causal relationship exists between two variables or whether the investigator's inferences about the data are valid Yin (2017). In this study, we recruited laypersons to provide us with annotations for privacy policy corpus. We set a qualification task to hire qualified participants that locate in the U.S. and have HIT approval rate greater than 95%. We consider an annotation valid if at least two qualified annotators agreed on.

External Validity External validity is the extent to which findings generalize Yin (2017). The privacy policy corpus contains 100 policies associated with 485 most downloaded apps at the time. Further analysis is required to identify the categories of these apps. Moreover, further analysis is required to evaluate the performance of trained NER models on specific app domains.

[9] https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html

8 Conclusion and Future Direction

In this paper, we aim to analyze sharing data practices of mobile apps and automatically extract and classify third party entities. For this reason, we define two categories for third party entities: (1) generic third parties, such as advertisers, partners, and affiliates; and (2) specific third party entities, including Google Analytics, Facebook, and PayPal. We first evaluate the performance of a pre-trained named entity recognition (NER) model on a set five of annotated privacy polices. The poor results of this pilot study suggest that pre-trained NER models on newswire corpus cannot be applied to extract third party entities from privacy polices. Therefore, we construct an annotated privacy policy corpus and utilize this corpus to train three different NER models. The annotated privacy policy corpus is constructed through crowdsourcing. Finally, our results shows an improvement over pilot study. Further, Stanford CoreNLP that implements CRF model using various combinations of natural language features associated with words in our corpus results in best performance when compared to the annotations.

For future, we foresee to downsample the segments of privacy polices that are not annotated as third parties. Using this strategy, we believe training and testing the Stanford CoreNLP CRF model will result in better performance. Further, regulators can utilize the results from this work to analyze the compliance of privacy policies in scale. The results of this work can also be applied to verify data practices of mobile apps. Therefore, the mobile app ecosystem can ensure a transparent environment for users.

Acknowledgments

We thank Álvaro Feal Fajardo for his participation as an expert in the pilot study. We also thank the Usable Security and Privacy Group at ICSI for their constructive feedback.

References

Alessandro Acquisti. 2010. The economics of personal data and the economics of privacy.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Jaspreet Bhatia and Travis D Breaux. 2015. Towards an information type lexicon for privacy policies. In *RELAW*, pages 19–24. 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.

Travis D Breaux, Annie I Antón, and Eugene H Spafford. 2009. A distributed requirements management framework for legal compliance and accountability. *computers & security*, 28(1-2):8–17.

Travis D Breaux, Hanan Hibshi, and Ashwini Rao. 2014. Eddy, a formal language for specifying and analyzing data flow specifications for conflicting privacy requirements. *RE*, 19(3):281–307.

Travis D Breaux and Florian Schaub. 2014. Scaling requirements extraction to the crowd: Experiments with privacy policies. In *2014 IEEE 22nd International Requirements Engineering Conference (RE)*, pages 163–172. IEEE.

Deloitte. 2017. 2017 global mobile consumer survey: Us edition.

Tatiana Ermakova, Benjamin Fabian, and Eleonora Babina. 2015. Readability of privacy policies of healthcare websites. *Wirtschaftsinformatik*, 15.

Jenny Rose Finkel, Trond Grenager, and Christopher D Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370.

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. *arXiv preprint arXiv:1802.02561*.

Daniel Jurafsky and James H Martin. 2008. Speech and language processing: An introduction to speech recognition, computational linguistics and natural language processing. *Upper Saddle River, NJ: Prentice Hall*.

Timothy Libert. 2018. An automated approach to auditing disclosure of third-party data collection in website privacy policies. In *Proceedings of the 2018 World Wide Web Conference*, pages 207–216.

Wang Ling, Chris Dyer, Alan W Black, and Isabel Trancoso. 2015. Two/too simple adaptations of word2vec for syntax problems. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1299–1304.

Frederick Liu, Shomir Wilson, Peter Story, Sebastian Zimmeck, and Norman Sadeh. 2018. Towards automatic classification of privacy policy text. *School of Computer Science Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-ISR-17-118R and CMULTI-17-010*.

Aaron K Massey, Jacob Eisenstein, Annie I Antón, and Peter P Swire. 2013. Automated text mining for requirements analysis of policy documents. In *2013 21st IEEE International Requirements Engineering Conference (RE)*, pages 4–13. IEEE.

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

Gabriele Meiselwitz. 2013. Readability assessment of policies and procedures of social networking sites. In *International Conference on Online Communities and Social Computing*, pages 67–75. Springer.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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.

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*, pages 2774–2779.

Welderufael B Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel Serna. 2018. I read but don't agree: Privacy policy benchmarking using machine learning and the eu gdpr. In *Companion Proceedings of the The Web Conference 2018*, pages 163–166.

Nicolas Viennot, Edward Garcia, and Jason Nieh. 2014. A measurement study of google play. In *The 2014 ACM international conference on Measurement and modeling of computer systems*, pages 221–233.

Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N Cameron Russell, et al. 2016. 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.

Robert K Yin. 2017. *Case study research and applications: Design and methods*. Sage publications.

Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven M. Bellovin, and Joel Reidenberg. 2017. Automated analysis of privacy requirements for mobile apps. In *NDSS*.