Calpric: Inclusive and Fine-grained Labeling of Privacy Policies with Crowdsourcing and Active Learning

Wenjun Qiu David Lie Lisa Austin
University of Toronto

Abstract

A significant challenge to training accurate deep learning models on privacy policies is the cost and difficulty of obtaining a large and comprehensive set of training data. To address these challenges, we present Calpric, which combines automatic text selection and segmentation, active learning and the use of crowdsourced annotators to generate a large, balanced training set for Android privacy policies at low cost. Automated text selection and segmentation simplifies the labeling task, enabling untrained annotators from crowdsourcing platforms, like Amazon's Mechanical Turk, to be competitive with trained annotators, such as law students, and also reduces inter-annotator agreement, which decreases labeling cost. Having reliable labels for training enables the use of active learning, which uses fewer training samples to efficiently cover the input space, further reducing cost and improving class and data category balance in the data set.

The combination of these techniques allows Calpric to produce models that are accurate over a wider range of data categories, and provide more detailed, fine-grained labels than previous work. Our crowdsourcing process enables Calpric to attain reliable labeled data at a cost of roughly $0.92-$1.71 per labeled text segment. Calpric's training process also generates a labeled data set of 16K privacy policy text segments across 9 data categories with balanced assertions and denials.

1 Introduction

Privacy policies are legal documents that disclose how a party collects, uses, and shares personally identifiable information (PII). Privacy legislation, such as the California Online Privacy Protection Act (CalOPPA) and the General Data Protection Regulation (GDPR), require digital services to use privacy policies to obtain consent for the collection and use of PII. In addition to consent, privacy policies also serve as a mechanism for transparency and accountability to ensure that digital services comply with privacy legislation [26]. With the massive growth of privacy policies, there has been significant interest in building tools to automate the analysis of privacy policies for compliance [49, 50], as well as to help end users understand them [19].

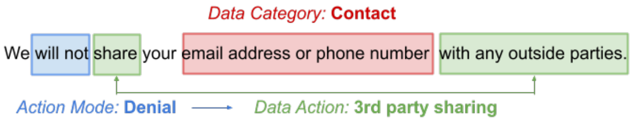

Data Category: Contact
We will not share your email address or phone number with any outside parties.
Action Mode: Denial
Data Action: 3rd party sharing

Figure 1: A fully labeled privacy policy segment with data category (contact information), data action (third party sharing), and its corresponding action mode (does not share).

However, previous attempts have been limited in inclusiveness and granularity[1]. For example, [49] and [50] are limited to only the most common categories of private data—contacts, device identifiers and location—but do not include less common data categories such as private health, financial and demographic information. Similarly, [19] trains hierarchical models that classify text segments by various privacy-related properties, such as the type of *data action* (i.e. collection, storage) and the *data category* (location, health). However, their models cannot distinguish fine-grained attributes, such as an explicit denial of a data action from the lack of a disclosure of a data action. In addition, they also cannot guarantee uniform performance on rare combinations (i.e."We do not store your health information", since explicit denials and storage are both rarer classes). PolicyLint [2] and PoliCheck [3] use a rules-based ontology to organize information types as a hierarchy, but also do not explicitly address minority data categories. These shortcomings exist because 1) the training set in previous approaches is not sufficiently large to cover less common cases, and 2) the imbalanced distribution of cases in privacy policies means that there will be insufficient training samples to provide good performance at fine-granularity.

In this paper, we address both challenges with a new training method for privacy policy classifiers, which we call

[1] We use inclusiveness to denote accurate classification for minority data categories and fine/coarse to denote the number of possible labels a model can predict at its output.

Calpric (Crowdsourcing Active Learning PRIvacy Policy Classifier). Calpric's training approach simultaneously gathers data and trains on it, synergistically using crowdsourced annotators and active learning to produce a larger and distributionally balanced dataset. Calpric overcomes a number of challenges that arise from crowdsourcing, such as untrained annotators and unreliable annotations, through a data category-specific text segmentation algorithm. The construction of previous data sets, such as OPP-115 and APP-350 [42,49], required the annotators to both select the relevant text segments in the privacy policy and label them, a complex annotation task for crowdsourced workers. Calpric automatically extracts segments comparable to the ones selected by trained annotators by determining whether a segment is relevant to the label being sought, identifying the correct segment boundary and capturing the necessary local context. However, simply acquiring more training samples would not guarantee that sufficient samples of rare categories are found. To overcome distributional imbalance, Calpric uses active learning, which uses a partially trained model to select unlabeled samples that are likely to contain information that the model currently lacks. The use of active learning itself is not without pitfalls—because active learning strategically selects samples for which it currently has little information, there tends to be little redundancy in the training samples it selects, and so an incorrect label provided by an annotator can have a large impact on model accuracy. Unfortunately, crowdsourced annotators do not provide correct labels consistently. Calpric's segmentation algorithm overcomes this challenge by increasing inter-annotator agreement and annotation accuracy, thus providing benefits to both the crowdsourcing and active learning components.

Building on these innovations, we use Calpric to build the Calpric Privacy Policy Corpus (CPPS). Training on CPPS, Calpric produces models that can provide accuracy over a greater breadth of data categories and with finer-grained labels than previous work. Calpric achieves an overall classification accuracy of 93.0% on the sentence/segment level, and 92.2% on the document level, which exceeds that of prior models trained on smaller data sets generated by trained annotators [19,42,49]. Simultaneously, Calpric's training methods enable the use of crowdsourced annotators labeled text segment for \$0.92-\$1.71 at an annotation quality similar to previous studies that used used more expensive trained annotators [49,50]. This enables Calpric to generate a training set set of 16856 labeled text segments, which is 3× larger than the largest previous human-annotated data set and contains significantly more information on minority data categories and classes.

In summary, this paper makes the following contributions:

1. We present Calpric, which synergistically combines crowdsourcing and active learning to produce Calpric Privacy Policy Corpus (CPPS), which contains 16856 labeled text segments extracted from 52K Android privacy policies covering 9 data categories and 3 data actions. CPPS is balanced across data categories and action modes. As far as we are aware, CPPS is the largest corpus of privacy policy text segments to date. The Calpric project is openly available https://github.com/dlgroupuoft/Calpric.

2. While creating CPPS, Calpric trains models that provide more inclusive and finer-grained classification performance on privacy policies. Calpric's models cover 9 data categories and can differentiate denials (i.e. explicit denial of a data action) from the absence of disclosure about a data action.

3. We perform measurements and find that Calpric's automated text segmentation increases the number of usable labels produced by 65%, enables the use of crowdsourced annotators, which reduces labeling costs by 9×, and results in a model that is more accurate than prior work, especially on minority data categories and classes.

4. We conduct a study on rare data categories and actions in 1,800 Android app privacy policies. We find that some categories are not as rare as previously believed, and that with the exception of the most popular applications, the number of explicit denials and use of controllable data actions increases with application popularity.

2 Challenges

In this section we better characterize the challenges of inclusive and fine-grained privacy policy classification and the availability of training data, which covers data imbalance, labeling cost, and segmentation challenges.

2.1 Inclusive and Fine-grained Classification

While previous work has demonstrated that it is possible to train models that achieve high classification accuracy on privacy policy text, that classification has been limited in inclusiveness and granularity. We summarize the limitations of previous approaches as compared to Calpric in Table 1. Limitations in breadth mean that model predication accuracy may suffer for minority (rare) classes, as machine trained models tend to optimize for accuracy in majority (common) classes. [49] and [50] only cover 3 common categories and do not cover rare data categories such as health, financial, demographic, social media, personal identifier, or survey data. [19] covers 11[2] rare categories, but has low accuracy in them due to the limited access to training data in those categories.

To alleviate some of these issues [19] combines different data actions when training models to predict certain private

[2] Polisis covers non-mobile privacy policies, we only train Calpric on categories that are applicable to mobile privacy policies.

	Granularity of labels	Categories
Polisis [19]	Relevant / ¬Relevant (2)	Common (3) + Rare (11)
MAPS [49]/ [50]	Assertion / Denial (2)	Common (3)
PolicyLint [2]	Assertion / Denial (2)	All[3]
Calpric	Assertion / Denial / Non Mentioned / Choice / Ambiguous (5)	Common (3) + Rare (6)

Table 1: Comparison of automated privacy policy analysis tools

attributes. For example, a single "information type" attribute model is trained across all data actions, combining actions such as 1st party (common) and 3rd party (rare) collection. As a result, they can only report prediction accuracy overall, and cannot guarantee accurate prediction for 3rd party collection.

Insufficient data and biased distribution also lead to coarser-grained categories, which provide less information to the end user. Previous machine learning approaches generally cannot identify phrases with explicit denials ("We do not collect your device identifier information") [19, 50]. While [49] and PolicyLint [2] can identify explicit denials, they do so using explicit rules to either generate synthetic denial samples, or check for negative modifiers in a dependency parse tree. In addition, none can extract indications of choice or opt-out options ("You may opt-out of this collection"), or ambiguity ("We may not collect your contact information"). In contrast, Calpric is able to differentiate between statements that (i) assert collection; (ii) explicitly deny collection; (iii) provide an opt-out option; or (iv) are ambiguous.

Calpric's training approach gathers data at lower cost through crowdsourcing and produces a dataset with a balanced distribution across target categories via crowdsourcing and active learning. However, the use of these two approaches themselves create new challenges, which we detail below.

2.2 Cost and Imbalance in Training Data

We detail the challenges of distributional imbalance in training data and the cost of acquiring labels for training.

2.2.1 Class Imbalance

OPP-115 [50] and APP-350 [49] are so far the most used privacy policy data sets for machine learning-based models, yet they both suffer from data imbalance. That is, the distribution of *data categories* collected by digital applications is not uniform, and as a result, privacy policies do not uniformly cover various categories of data. Imbalance exists across both data categories and positive/negative classes. For example, there are many more privacy policy text segments disclosing the collection of location information than health information.

	OPP-115	APP-350	CPPS
Contact	8.46% | 1217	22.5% | 1744	48.07% | 2852
Location	2.45% | 449	22.1% | 971	51.29% | 2318
Device	0.57% | 700	14.9% | 2062	50.57% | 2828
Demogra.	1.35% | 408	17.7% | 481	48.44% | 3552
Financial	5.06% | 435	-	51.73% | 2339
Health	2.37% | 211	-	48.60% | 1790
Survey	4.76% | 84	-	45.13% | 421
Personal id	11.3% | 106	-	38.76% | 387
Soc. media	1.35% | 74	-	47.15% | 369
Average	4.19% | 3684	19.3% | 5258	47.75% | 16856

Table 2: The number of labels across data categories and percentage of denials in previous datasets compared to CPPS[4]

Similarly, the privacy policy segments are heavily skewed towards assertions (i.e. a *positive label*) of data collection than denials (i.e. a *negative label*).

To better characterize the challenge, we tabulate both the number of labeled segments in each category and the percentage of denials for the two data sets and CPPS in Table 2. We can see that data for certain categories is extremely sparse: OPP-115 contains thousands of labels on contact data but only 44 labels on health data. Consequently, Polisis [19], which is trained on OPP-115, struggles with a low accuracy in minority classes such as health. We also see that the data sets exhibit class imbalance across categories, with only 4.19% of the samples in the OPP-115 having a denial label. APP-350 has a slightly better class balance, but this is due to the addition of synthetic denials, which they accomplished by manually changing a positive sample into a denial one. For instance, a positive sample: *"Our App collects your location data"* might be converted to *"Our App does not collect your location data"*. We note that while this approach mitigates the class imbalance to some extent, it cannot be used to improve data category imbalance. Furthermore, there is no assurance that such synthetic labels are representative of real denials in the wild. In comparison, after active learning, the CPPS training set generated by Calpric has almost perfect class balance. As we can see, Calpric is able to find and label many more samples in the 6 rarer data categories.

2.2.2 Labeling Cost

Previous data sets, such as the OPP-115 [42] and the APP-350 [49] data sets, are generated with trained human annotators, usually composed of law students. Even with their prior legal knowledge, it still takes considerable time to read through a privacy policy and label the specific text segments that indicate, for example, whether personal information is collected and if so, what type of information is collected. For instance, OPP-115, which contains 115 privacy policies, was labeled by 10 law students, spending an average of 72 minutes

[3] Technically PolicyLint and PoliCheck are able to handle all categories, but they rely on name-entity recognition, which does not take into account context like our language-model based extraction.

[4] Each privacy policy in OPP-115 was annotated by multiple annotators, thus the number of labels are not equivalent to the number of segments. 0.75 is the middle value of overlap threshold provided by the authors to try to consolidate multiple labels on similar text segments

on each privacy policy. The speed, availability and cost of trained annotators limits the size of the data set. Recognizing this challenge, we propose a crowdsourcing solution reduces the cost of labeling by 9×.

One obvious question that arises with the use of crowdsourced annotators for privacy policies is whether such untrained annotators can sufficiently understand privacy policy language to produce accurate labels [43]. We note however, that since privacy policies are the primary method of obtaining consent from individuals for the collection of information, to obtain meaningful consent, they must be understandable by a representative target individual from whom the information collection would likely happen. For example, Canada's privacy legislation, PIPEDA, states that *"the consent of an individual is only valid if it is reasonable to expect that an individual to whom the organization's activities are directed would understand"* and the GDPR defines consent as *"any freely given, specific, informed and unambiguous indication of the data subject's wishes"*—in both cases, the user must "understand" or be "informed." While it is true that many privacy policy documents fall short of this idealized level of transparency, we argue that the interpretation that an average person has for a privacy policy intended for them (were they to actually read the policy) is still relevant, and is what Calpric tries to predict.

Nevertheless, one cannot simply use crowdsourced annotators and expect to achieve accurate classification, especially in combination with active learning. First, crowdsourced annotators acquired through platforms such as Amazon's mTurk service, generally do best on short, simple tasks, such as single-sentence translation [1] and sentiment analysis [7]. Further, reducing the complexity and length of tasks is beneficial for crowdsourced annotators [8, 23]. Second, active learning methods generally assume reliable oracles that always return correct labels, which is not the case for crowdsourced annotators. Previous privacy policy studies presented several crowdsourced annotators with the entire privacy policy, asked them each to assign a label, and then attempt to increase label reliability by only accepting the label if the inter-annotator agreement is above some threshold [43]. A drawback to this is that different annotators may select different text segments to label, and the text segments themselves may not align perfectly, decreasing inter-annotator agreement.

Calpric addresses these challenges by extracting relevant text segments from a privacy policy and presenting these to the crowdsourced annotators so that all of them label the same segments. This both decreases the time they must spend on the task and increases inter-annotator agreement rate, which both decrease the cost of acquiring labels. Calpric re-requests labels for text segments that did not achieve an inter-annotator agreement above an acceptable threshold—a mechanism called *relabeling*. However, some segments may suffer from chronically poor agreement—that is they do not achieve sufficient agreement even after several attempts. There can be several reasons for the chronically poor agreement for a segment. First, Calpric may mistakenly extract a segment that is not relevant to the labeling task. Our crowdsourced task also has workers annotate whether a segment is relevant or not, which is then used as a label to retrain the model used to classify if a text segment contains relevant information about a data category. Second, the text segment may be relevant but inherently ambiguous—that is the text is not sufficiently clear about whether collection is occurring or not [32]. For example, in

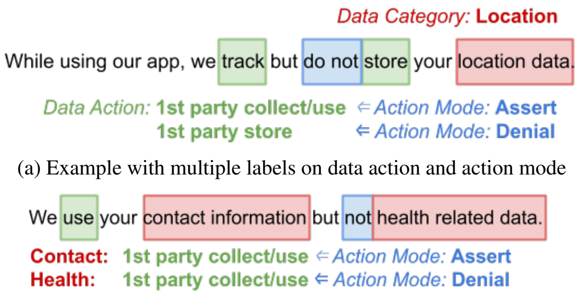
> *"if you log-in using a third party social media account, we may not collect any personal or account information from that social media provider"*

the "may" makes it ambiguous whether information is collected or not, and users have no control over the usage of private data. In addition, using the ambiguous term "personal information" without further details makes it impossible to infer which data category the segment refers to and undermines the purpose and value of privacy policies [32]. To prevent wasting more crowdsourcer resources on such text segments, Calpric stops the relabeling attempt after a preset number of iterations, and labels such segments as *ambiguous* and does not use them for training. While classifying all text segments that do not achieve an acceptable level of inter-annotator agreement as ambiguous necessarily over-approximates the true set of ambiguous text segments, we show that Calpric is still able to achieve high classification accuracy despite the loss of potentially usable text segments due to this over-approximation.

Our current implementation of Calpric uses crowdsourced annotators from Amazon's mTurk service to provide labels used for training. While mTurk captures a large portion of the mobile app user population, who are the target individuals of mobile app privacy policies in our data set, we note that mTurk cannot capture every subgroup of mobile app user. For example, children are not permitted to use the mTurk service, so our current Calpric models cannot predict how children might interpret privacy policies. Nonetheless, this is a limitation of mTurk, not Calpric—we believe that our current implementation of Calpric can be extended in a straightforward manner to predict for any subgroup of individuals so long as annotators from that subgroup can be recruited to provide labels from which Calpric can learn.

2.3 Segmentation

As mentioned earlier, extracting text segments is crucial to Calpric's ability to provide accurate, broad classification of privacy policies at a lower annotation cost. While previous work, such as Polisis, used existing off-the-shelf segmentation tools [17], Calpric requires text segments for annotation, not just classification. For accurate annotation by crowdsourced workers, Calpric must extract text segments that not only semantically coherent to a topic, but relevant to the specific data


Data Category: Location
While using our app, we track but do not store your location data.
Data Action: 1st party collect/use ⇐ Action Mode: Assert
1st party store ⇐ Action Mode: Denial
We use your contact information but not health related data.
Contact: 1st party collect/use ⇐ Action Mode: Assert
Health: 1st party collect/use ⇐ Action Mode: Denial

(a) Example with multiple labels on data action and action mode

(b) Example with multiple labels on data category

Figure 2: Privacy Policy segments with multiple labels

category for which an annotation is being sought. As a result, we build custom data category-specific *Segmenters* in Calpric.

A Calpric Segmenter first identifies sentences that are relevant to a data category, and then uses standard topic similarity metrics to *contextualize* the sentence into a coherent text segment. Finally, in contrast to previous work, we also evaluate the safety of using text segment level classification to assign global labels to a privacy policy (Section 5.1.1). Specifically, we use our Segmenters to examine how often privacy policies may have multiple contradictory text segments, and whether these represent real contradictions where segment-level labels may not be consistent with a privacy policy-level label.

3 Design Overview

The goal of Calpric is to train a hierarchical multi-label classification model that extracts the *action mode* associated with a *data action* for a set of *data categories* from privacy policy text. Calpric accomplishes by first extracting relevant text segments from a privacy policy and then labeling those segments as shown in Figure 1. At the highest level, Calpric labels text segments into 9 data categories: *contact, device information, location, financial, health, demographic, survey, personal identifier* and *social media data*. The next level identifies 3 data actions: *1st party data collection/use, 3rd party sharing* and *data storage*; and then 5 possible action modes are labeled for each data action: *assert* (action is performed), *choice* (action is performed but the user may opt-out), *denial* (denial of the action being performed), *not mentioned* (irrelevant to data practice), and *ambiguous* (identify policy texts that are poorly described, which aligns with interests of recent legal studies [26, 32]). Note that a single text segment may contain information about multiple data categories and multiple data actions, as shown in Figure 2, making this a fine-grained multi-label classification task.

3.1 The Calpric Pipeline

Calpric is implemented as a pipeline that processes the input in 3 major stages as illustrated in Figure 3. First, a *Data Preparation* stage ingests the raw privacy policies and performs basic pre-processing to prepare them for labeling by the crowdsourced annotators (5 per segment in our study) and subsequent training. Second, a *Segmenter* stage extracts text segments relevant to each data category from the processed privacy policies and simultaneously trains and updates the Segmenter models with active learning. The final *Action Model Training* also uses active learning to train models that produce multi-class action mode labels for each data action/data category combination.

Calpric's pipeline effectively decomposes the learning task into two separate models. The Segmenter includes a *Category Model* that selects text segments that are relevant to one of the data categories. After this, *Action Models* at the next stage only have to predict action mode labels for each data action. This decomposition has two benefits. First, the decomposition of a single task into two simpler sub-tasks fits the requirements of crowdsourcing, which requires short, simple tasks. Second, active learning requires an even distribution of bootstrap data, ideally in all combinations, which becomes intractable as the label dimension grows [44]. Decomposing the labeling task reduces its dimensionality. We now describe each of the pipeline stages in more detail.

3.2 Data Preparation

The Privacy Policy Scraper downloads metadata of 375K Android applications from the Play Store and filters down to a final dataset of 51,781 usable privacy policies in the plain text format. The processing details are included in Appendix A.

Calpric then prepares a privacy-specific contextualized embedding named PriBERT. Specifically, we perform additional pretraining on the *bert-base-uncased* model using the 52K privacy policies collected by the Privacy Policy Scraper, with a maximum sequence length of 128 and a batch size of 8. Each training checkpoint captures the BERT activations from the last 4 hidden layers of the previous transformer checkpoint. We also apply a truncation rate of 2% to provide robustness to non-sentential inputs.

3.3 Segmenter

We create data category-specific specific Segmenters by training a model that identifies relevant sentences, and then using standard NLP tools to identify the appropriate segment boundaries. We begin by using the NLTK [5] Sentence Tokenizer to split entire privacy policies individual sentences. Bullet lists are converted into a sentence heuristically by concatenating each list item with the text just before the list items.

The individual sentences are passed to a Category Model, which classifies if the sentence is relevant to a data category. There are 9 separate Category Models in Calpric, each responsible for one data category Calpric currently supports and a sentence can be classified as being relevant to more than

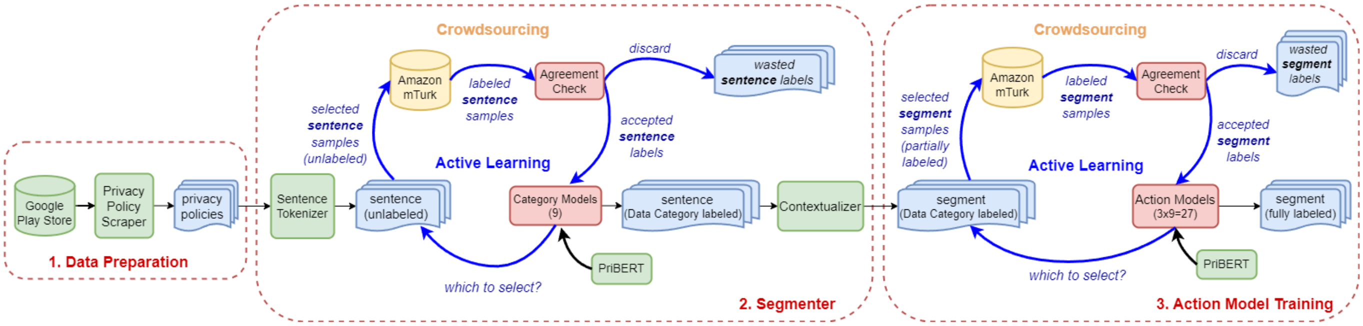


Figure 3: A simplified overview of the active learning system

one data category. Finally, relevant sentences will be *contextualized* by adding additional sentences into a *text segment*, which provides the context necessary for the annotators to understand and apply the correct label. For example, consider the text segment:

> *Personal information is data that can be used to uniquely identify or contact a single person. When you visit, download or upgrade our app or our products, we do not use this information explicitly. However, we may collect personal information to improve our services and deliver a better experience.*

All sentences in the segment contribute to its meaning: without the first sentence, it is not clear that personal information refers to contact information, but if the segment were to only contain the first two sentences, annotators may be misled to provide a denial label to the segment. At the same time, including extraneous sentences in the segment increases the time and effort for annotators, which may increase labeling costs. Sentences are contextualized by the *Contextualizer* component. We describe the Category Model and Contextualizer that make up the Segmenter in more detail below.

3.3.1 Category Models

For each data category, Calpric trains a binary Category Model to identify relevant sentences: it projects each sentence into a vector representation using the PriBERT embedding mentioned in Section 3.2. These vectors are then fed into a set of binary classification models fine-tuned on top of BERT. We set the batch size to 32 and the number of epochs to 48. We use leaky ReLUs [29] and a dropout value [39] of 0.1 to prevent over-fitting while sustaining the weight updates to avoid vanishing gradient problems in the propagation process. We apply a softmax activation function to the model output and use cross entropy loss and ADAM optimizer [22] to update model weights.

Each Category Model is initialized using a bootstrap training set of 200 sentences to be labeled. The bootstrap set is prepared in two steps: first, Calpric randomly selects a number of privacy policies from the unlabeled pool, and crowdsourced workers are asked if the target data category is mentioned in these given documents. While this requires the annotators to read through the full documents, they do not need to give detailed annotations. The privacy policies that are annotated as mentioning a target data category will be tokenized into sentences, and 200 sentences are randomly selected as the bootstrap set for each of the 9 Category Models. After this, active learning is applied to select more sentences for labeling. It is possible that the bootstrap set does not cover all classes and the model will fail to converge, but we did not encounter this situation in all our experiments. If such a situation arises, it would be necessary to expand the size of the bootstrap set until all classes are represented.

3.3.2 Contextualizer

The Contextualizer starts with a relevant sentence selected by the Category Model and iteratively expands it into a text segment by examining the sentences immediately before and after to see if either should be added to the current segment. To do this, the Contextualizer computes the similarity of the sentences to the current text segment using the Word Mover's Distance (WMD) [24] and compares it against a threshold calculated from the WMD mean and standard deviation of sentences within the current privacy policy [12]. This process continues until both the previous and subsequent sentences are not similar enough to be added to the current text segment.

3.4 Action Models

Action Models are the final stage of the Calpric pipeline. Similar to Category Models, Action Models are designed with three fine-tuned BERT-based classifiers, each responsible for labeling a data action for each data category (9 data categories with 3 data actions for a total of 27 Action Models). Unlike the binary Category Models, Action Models are multi-classed, as shown in Figure 4. Together with the relevancy label obtained from the Category Model, the input segment is assigned labels on action modes for each data action. These labels result in a complete segment-level label. We bootstrap Action Models by sending segments output from Category Models to mTurkers to label until there are at least 20 labels for each of the 5 action

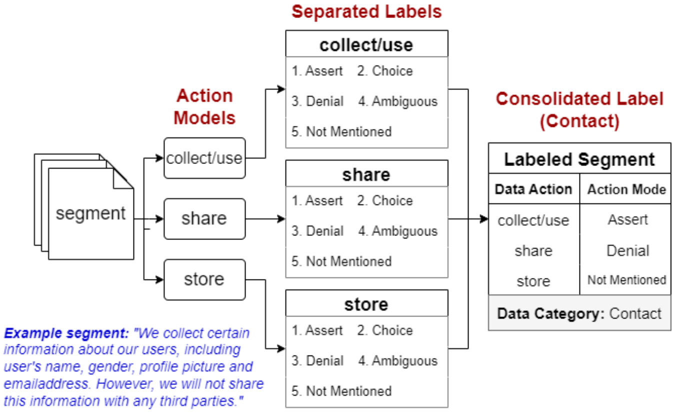


Figure 4: The action classification process to produce a label on the given *contact* sample segment

mode options for the 3 data actions, resulting in at least 300 labels per data category.

4 Active Learning and Crowdsourcing

4.1 Crowdsourcing Training Data

It is well-known that crowdsourced workers may have varying degrees of skill and may not pay full attention when performing tasks [9], which is especially dangerous for Calpric which relies on active learning to obtain training labels. To mitigate these challenges, Calpric follows best practices of qualifying mTurk workers as annotators, including specific requirements and qualification tests specific to data category, and designing a simple question survey to perform the labeling task. Details of this are provided in Appendix B. *The use of human annotators was approved by our institutional review board (IRB).*

After obtaining labels from crowdsourcers, Calpric carefully examines them to ensure label reliability. Only labels with an inter-annotator agreement greater or equal to the *Acceptance Threshold* (80%) will be accepted as training samples. All accepted labels are used in the next training iteration of the appropriate Category Model or Action Model, while unaccepted labels are discarded, referring as the *Wasted Requests* as they still cost money but confer no benefit to the training of Calpric's models. Calpric's Segmenter increases the agreement rate thus decreasing waste, directly leading to reduced labeling costs (Section 5.2).

4.2 Active Querying Strategy

As mentioned in Section 3.1, Calpric uses two types of classifiers, the Category Models and the Action Models, both of which are trained with active learning. During active learning, the Category Model selects sentences from the entire unlabeled pool of privacy policy sentences, while the Action Model selects text segments from the pool currently labeled by the Category Model as relevant to the target data category. As a result, the sentence selected by the Category Model may not be in the text segment selected by the Action Model. To consolidate the selections and simultaneously train both models, Calpric initially prioritizes the Category Model, otherwise the Action Model may select too many irrelevant text segments for labeling. Thus, we initially allow the Category Model to pick sentences for labeling until it reaches an F1 threshold of 85%, and then give the Action Model priority in selecting segments for labeling requests and continue training on both models.

To select samples to label, Calpric uses pool-based sampling as this method is best suited to settings such as ours that have a large unlabeled pool [35]. Our implementation builds on top of the *modAL* active learning framework, which is built upon the study of Cardoso et al. [6]. Further details are presented in Appendix D. We use Uncertainty-based Sampling [25], which selects the samples with the greatest uncertainty for labeling, and modify it for our situation, which has a very large pool of unlabeled samples. Specifically, we compute not only the uncertainty for the samples in our pool, but also record the history of uncertainty scores. Samples that achieve a low-level of uncertainty over a period of time are likely never to be selected since uncertainty scores generally decrease as the model learns more about the input space. As a result, we remove these from the pool to save having to recompute the uncertainty for these on each active learning iteration, as the cost of computing uncertainty for every unlabeled sample is significant due to their large number.

Note that the active learning process is designed to be online, but as a prototype, we still perform manual checks throughout the process to minimize wasted mTurker funds in case of program bugs. For most iterations, Calpric only updates the model with new labels from mTurkers, which takes <1 min. To prevent catastrophic forgetting, we retrain all labels after 10 iterations. The main bottleneck is waiting for mTurkers to accept HITs and provide annotations.

4.3 Relabeling Strategies

Recall that text segments whose inter-annotator agreement fails to cross the agreement threshold are not accepted for training and waste labeling resources. Calpric currently discards these segments, a policy, we call *Label and Discard*. Label and Discard protects the trained model from training on incorrectly labeled samples, which can lead to lower accuracy, as shown in our Evaluation in the *Acceptance Threshold* section in Appendix C. Calpric aims to add 30 new labels each active learning iteration before selecting another set of samples to label. Since, we know all unaccepted labels will be wasted, we can estimate the number of label requests we need to make to attain 30 accepted labels. We have measured the acceptance rate for our survey to be approximately 73%, which requires Calpric to request 42 surveys each iteration.

Another option, rather than wasting the unaccepted labels, is to re-request them to see if the agreement rate for a larger number of annotators can be acceptable. We call this policy *Incremental Relabeling*, which we implemented based on that of Zhao et al. [47]. Incremental Relabeling trades off the possibility of recovering some of the wasted labeling resources for the risk of wasting even more resources. We put a cap of 3 on the number of times any text segment can be requested for labeling for a total of up to 15 annotations and keep the agreement threshold at 80% (i.e 4/5, 8/10 or 12/15). We evaluate these two policies in the *Relabeling* section in Appendix C.

5 Evaluation and Measurements

We begin by evaluating the accuracy of Calpric's Category Models (9) and Action Models (27) against a test set of segments whose labels have been provided by law students. We then evaluate the accuracy of the Segmenter against segments that were labeled by those trained annotators, as well as the Segmenter's contribution to reducing annotation cost. We also evaluate Calpric's use of active learning to reduce annotation costs and mitigate data imbalance. Finally, we compare the cost of Calpric's use of crowdsourced annotators versus the cost of using trained annotators. Measurements of Calpric's sensitivity to the acceptance threshold, relabeling strategy and size of the unlabeled pool are given in Appendix C.

5.1 Accuracy of Category and Action Models

We measure the absolute accuracy of Calpric's predictions against the law student-labeled test set. We then show the benefits of training on the CPPS dataset by comparing Calpric's Category and Action Models against a baseline classifier trained on OPP-115 and APP-350.

5.1.1 Absolute Accuracy

As APP-350 and OPP-115 do not have labels for "choice" or "ambiguous" we set aside 30 labeled segments per data action in each data category for a total of 810 test segments. We then train Calpric without the test segments and tabulate Calpric's accuracy in Table 3. To save space, we show the accuracy of each action mode, averaged across all three data actions in each data category, as well as an overall accuracy for each data category.

The evaluation shows that Calpric can achieve good accuracy across both common and rare data categories, as well as good accuracy in all the fine-grained action model labels. Specifically, it achieves an average f1 of 0.86 and an overall accuracy of 0.90, with the highest f1 in *Assert* (0.92) and *Not Mentioned* (0.91), and the lowest in *Ambiguous* (0.78). We suspect this is because *Assert* statements are usually described clearly as per the legal requirements, while *Not Mentioned* labels are very different from labels of the remaining 4 action modes which all have some contextualized relationship with the target data category. Therefore intuitively it is easier for Calpric to correctly predict on *Not Mentioned* labels. On the other hand, its accuracy in predicting *Ambiguous* labels is slightly lower, as there are many factors leading to ambiguity in privacy policies, e.g. long sentences, terms being too general, confusing wordings, incomplete information, and so on. However, we believe that an F1 score of 0.78 is still very promising, given the fact of the potential noise within the ambiguous labels.

data category	**denial**	**assert**	**choice**	**amb.**	**NM**	**avg. f1**
contact	0.80	0.93	0.83	0.78	0.94	0.86
location	0.79	0.85	0.77	0.67	0.94	0.80
device	0.89	0.92	0.83	0.82	0.96	0.88
demographic	0.81	0.88	0.86	0.80	0.86	0.84
financial	0.88	0.96	0.80	0.82	0.86	0.86
health	0.94	0.95	0.82	0.89	0.96	0.91
survey	0.82	0.90	0.80	0.78	0.91	0.84
personal id	1.00	1.00	0.97	0.82	0.97	0.95
social media	0.81	0.90	0.93	0.67	0.81	0.82
average	0.86	0.92	0.85	0.78	0.91	0.86

Table 3: F1 score on the CPPS test set. We show the f1 for each of the denials, assertions, choice, ambiguous (amb.) and not-mentioned (NM) labels, as well as the average f1 value.

5.1.2 Benefits of CPPS

As explained in Section 2.1, previous tools classify at a coarser granularity than Calpric, so we cannot directly compare the accuracy of labels produced by Calpric against theirs. As previously shown in Table 1, previous tools label privacy policy text segments into two classes, while Calpric provides 5. However, all previous tools were trained exclusively on the APP-350 and OPP-115 data sets, while Calpric is trained on the larger and balanced CPPS dataset that was produced by Calpric's training approach. It is therefore unfair to compare Calpric against existing tools. Instead, we train baseline models with identical architectures as those of Calpric's on data from APP-350 and OPP-115. We randomly select some segments from the the APP-350 and OPP-115 as the test set and tabulate the F1 result in Table 4. Calpric outperforms the baseline models on fine-grained classification tasks as well as the overall average across all data categories.

5.2 Segmenter

In previous tools, text segments used for training were extracted by human annotators while Calpric's training approach automatically extracts text segments. We thus measure how closely Calpric's segments extracted by Calpric's Segmenter cover the same segments extracted by human annotators. To show that the Segmenter increases inter-annotator agreement,

		contact	device	location	demogr.	financial	health	survey	pers. id	soc. media	average
Category	Calpric	**0.94**	0.90	**0.89**	**0.97**	**0.94**	**0.89**	**0.87**	**0.78**	**0.82**	**0.89**
	baseline	0.90	0.79	0.85	0.78	0.92	0.43	0.80	0.63	0.75	0.76
Action	Calpric	**0.97**	**0.91**	**0.92**	**0.90**	**0.89**	**0.98**	**0.96**	**0.84**	**0.89**	**0.92**
	baseline	0.91	0.68	0.62	0.65	0.63	0.54	0.60	0.57	0.57	0.64

Table 4: F1 comparison between Calpric Category and Action models vs baseline model

we compare the level of agreement with and without automated text segmentation. Finally, we investigate the frequency and impact of conflicting text segments in privacy policies.

5.2.1 Segment Coverage

To examine the ability to capture useful segments from privacy policies, we compare the outputs of Calpric's Segmenter against the text segments extracted by human annotators in APP-350. We measure the number of labeled sentences in APP-350 in the relevant data categories that are covered by (i.e. appear entirely in) some segment extracted by our Segmenter. We run our Segmenter on all the original privacy policies provided by APP-350 and find that 97% of the sentences in APP-350 are covered by our Segmenter. We examine the uncovered sentences and find that the vast majority are either ambiguous or irrelevant sentences. For example, in APP-350, the following segment is labeled as collecting contact information:

> *"Anonymous Data AI Factory may collect and use anonymous data, via Google Analytics. We cannot identify you from this data. Such data is pooled to give us general statistics on our audience and usage of our apps."*

but is not labeled as collecting contact information by our Segmenter.

5.2.2 Increasing Inter-annotator Agreement

We now evaluate the benefit of having annotators label text segments instead of existing approaches such as full-text or paragraph-level labeling. We randomly select 3 privacy policies from APP-350 and conduct an experiment where mTurk annotators are asked to either label the full document, paragraphs produced by existing preprocessors, or text segments extracted by Calpric. For full document labeling (FDL), we present the full document to annotators and ask them to label 20 sentences that denote any of the data actions and action modes on any of the relevant data categories. For paragraph labeling (PL), we present paragraphs for annotators to select and label. We generate paragraphs using a modified version of the PolicyLint preprocessing script. We exclude outlier paragraphs falling out of the middle two quartiles (i.e. 25-75%) of the distribution in terms of length, as they might be excessively long or so short as to contain no useful information. For segment labeling (SL), we have annotators label 20 segments per privacy policy using the Calpric's survey. To align with our labeling scheme (i.e. includes more classes), which differs from the original APP-350, selected samples are reviewed and re-labeled by law students. We calculate label accuracy (Acc.) by measuring the percentages of raw labels that are correctly labeled by mTurkers. A label is considered accurate if and only if it covers one of the labeled sentences in APP-350 and the labels in data category, data action and action mode all match.

The results are tabulated in Table 5. We found that SL produces the most efficient results: annotators needed roughly 4 times and 2 times as long to produce the same number of labels using FDL and PL, respectively. That is, 21 minutes for SL, 46.5 minutes for PL, and 90 minutes for FDL (per policy). As a result, to maintain the same hourly pay rate, FDL and PL require 4× and 2× the payment to obtain the same amount of labels as using SL. We find that SL gives the highest accuracy. Closer inspection of FDL shows that some annotators incorrectly label segments such as *"We do not track your location unless we have your opt-in consent."* as "No" instead of "Choice", as they truncated the phrase after "location", thus missing the later part that makes it "Choice". SL also outperforms PL by 15%. We found that this was mainly because text critical to understanding a data action may span a paragraph break.

We also observe that the agreement rate (AR) for FDL is significantly lower, as annotators often disagree on text boundaries for their labels. This result aligns with what previous scholars observed [42], even when labels are created by trained annotators. On the other hand, ARs for PL and SL are similar, with PL being slightly higher (1.3%).

We compute the raw cost per label by dividing the total payments by the number of unique labeled segments, the effective cost after discarding unaccepted segments, and the time taken to produce each raw label. FDL is both more expensive (4× for raw labels, and 5.2× for accepted labels), and requires 4.3× as much time to produce raw labels than SL. Qualitatively, we also found that because annotators are free to select the text they wish to label, they tended to select easier segments that could be found by simply searching the text for keywords (i.e. "demographic" or "contact") instead of carefully reading the complete privacy policy. As a result, this may exacerbate class imbalance and lead to poorer coverage of the privacy policy text. PL has a better performance than FDL, yet is still 2.2× more expensive than SL.

	Acc.	AR	c_raw	c_accept	t_raw
FDL	55%	6.7%	$0.5	$4.55	4.5 mins
PL	63.33%	73%	$0.28	$1.88	2.32 mins
SL	78.33%	71.67%	$0.125	$0.87	1.05 mins

Table 5: Comparison between Full-Document Labeling (FDL), Paragraph Labeling (PL), and Segment Labeling (SL) using crowdsourcing: label accuracy(Acc.), agreement rate (AR), the average cost per raw label (c_raw), average cost per accepted label (c_accept), and the average time took to produce one raw label (t_raw).

	n_nonAl	n_Al	Al_save	m_start	m_end
contact	1596	942	40.98%	4.92%	49.68%
location	840	632	24.76%	7.04%	50.00%
device	998	610	38.88%	4.20%	46.07%
health	732	529	27.73%	5.56%	42.53%
finanical	982	793	19.25%	6.80%	49.94%
demogra.	1428	1149	19.54%	7.55%	58.40%
average	1096	776	28.52%	6.03%	49.44%

Table 6: Comparison between active vs. non-active Category Models: *n* is the number of training labels required to achieve F1=0.85; A higher *AL_save* indicates a larger improvement in the learning progress; *m* is the percentage of samples in a minority class, recorded at the bootstrap (*m_start*) stage and the completion of training (*m_end*).

5.2.3 Intra-document conflicts

One potential risk of segment-level labeling is that it becomes unclear how to consolidate contradictory labels if more than one relevant segment for a data category and data action appear in the same privacy policy. We examined 115 privacy policies and find that 12% have contradictory labels for a data category and data action. Of these, 8% have conflicts that take the form of a denial that they collect a certain type of data, followed by a statement that they will share that data if legally required to do so. We also compare contradictions found by Calpric with PolicyLint [2], on OPP-115. PolicyLint finds 20 of the privacy policies have contradictions, 8 of which are also found by Calpric. Of the 12 not found by Calpric, PolicyLint considers 5 contradictions only because it lacks a "Choice" label, 2 are due to PolicyLint and Calpric having different subsumptive relationships (i.e. PolicyLint considers demographic information as a type of personal information while Calpric considers them separate classes), 2 because of differences in the annotation scheme and 3 due to differences in the data categories covered. Calpric also finds 6 contradictions that PolicyLint does not find. We manually verified that they are real contradictions and were missed by PolicyLint as it is unable to differentiate between choices and explicit denials.

5.3 Active Learning

The following evaluations demonstrate how active learning enables Calpric to 1) achieve lower labeling costs by reducing the number of labeled training samples, and 2) improve class balance to achieve higher accuracy.

5.3.1 Labeling Savings

We evaluate the reduction in the amount of labeled training data that active learning provides to Calpric. To do this, we set up an experiment that evaluates how many training samples are required to achieve a certain Category Model accuracy across all 6 data categories. We prepare a set of balanced test and training sets for each data category by using samples from APP-350 (and OPP-115 if APP-350 does not label such categories). Text segments that are labeled as in a data category are assertions, and segments not in the category are denials. We simulate a non-active training procedure by randomly selecting samples from this pool and compare this to a training procedure that uses active learning to select samples. We record the number of training samples required to achieve an F1 score of 0.85 in the non-active learning and active learning cases in the n_nonAL and n_AL columns respectively, and the percentage of samples saved in the active learning case in the AL_save column in Table 6. We did not include *survey*, *personal identifier* or *social media* because the non-AL model is not able to find enough training samples to achieve the required accuracy. For the other categories, we observe that the average savings in the number of samples required is 28.52%. The savings are higher in categories where more samples are required in the non-active learning case, suggesting that there are many similar samples that the random selection in the non-active learning case selects over and over again. However, this trend is weaker in financial and demographic categories. Upon further investigation, we observe that these two categories exhibited longer samples with more complex features. As an example, device-related sentences such as

> *"Our app may access your unique device identifier to tailor ads."*

are usually simple and concise, whereas a demographic-related sentence like

> *"App A provides us with access to certain information about such User as is stored in the User's App A account, namely, the User's public profile (i.e., User's name, profile picture, email address, gender and other public information) and his/her list of friends and/or any other information which is detailed and displayed to the User in the notice which appears during the log in process."*

is not. This complexity means that a good number of samples are required to achieve higher accuracy regardless of how the samples are selected, which makes the advantages of active learning in these cases, though still significant, somewhat lower than in the cases with simpler samples. The contact and device categories exhibit the greatest savings. We believe this is a result of those categories having a heavy-tailed distribution, the tails of which active learning is much better able to target than random selection.

5.3.2 Class Balance

To verify that Calpric mitigates class imbalance, we measured class balance as a percentage of samples in a minority class, recorded at the bootstrap phase and at the end of model training, respectively. Table 6 also shows the starting (m_start) and ending (m_end) percentages of minority samples in all data categories. We see that in all categories, the minority class initially starts of low, but eventually approaches close to making up half of the training set (50%) used to train the model. We also observe an inverse correlation between the increase in minority percentage and the AL saving percentage. This is intuitively reasonable because boosting from a lower minority percentage to a higher minority percentage naturally requires more training samples. We performed a similar evaluation on Action Models, and found similar results in the improvement of class balance. In these cases, the increases in the minor classes were 29.6% in the health, 19.4% in the financial, and 24.0% in the demographic categories.

We also compare Calpric with standard upsampling methods to mitigate to class imbalance, such as duplicating or synthesizing minority samples [21]. To do this, we randomly select 800 labeled text segments and sort them by whether they are in a minority or majority class. We then duplicate, synthesize or select minority labels with active learning until the training sets are balanced, train our models and compare accuracy. To duplicate samples, we randomly select samples from the minority set and repeat them in the final training set. To synthesize minority samples we randomly select majority samples and negate them using the method described in [4], implemented with Stanza[5]. Finally, we use active learning to select enough additional minority samples to balance the training set. We then train a fine-tuned BERT model with a batch size of 32 for 48 epoches and evaluate the resulting model on a test set with a naturally occurring distribution of majority and minority samples. We repeat the experiment 5 times and tabulate the loss, accuracy, F1 values for majority and minority test sets, weighted F1, and balanced F1 for the 3 data set balancing methods, as well as training with no balancing method in Table 7. As we can see, Calpric with active learning achieves better results except for loss. Using no balancing method (None) achieves the lowest loss, but as expected, suffers from low accuracy in the minority class, as it has simply optimized for the majority class.

Balance alt.	acc.	F1 (maj/min)	wgt. F1	bal. F1
None	0.79	0.88, 0.08	0.69	0.44
Duplication	0.78	0.87, 0.29	0.73	0.58
Negation	0.79	0.88, 0.32	0.75	0.62
Calpric	**0.80**	**0.89**, **0.56**	**0.81**	**0.73**

Table 7: Performance comparison against other alternatives, including the non-AL baseline method (Random), upsampling with Duplication and Negation. Metrics include accuracy, F1 for majority and minority class, weighted F1, and balance F1.

	AR	avg_agree	Acc.	cost
Law students	0.83	0.91	0.99	$8.20
mTurkers	0.81	0.88	0.98	$0.92

Table 8: Comparison between labels created by mTurkers vs. law students: acceptance rate (AR), average agreement rate for all questions among all annotators (avg_agree), label accuracy (Acc.), and the average cost to produce one accepted label (cost).

5.4 mTurkers Vs. Law students

While trained annotators, such as law students, might cost more than crowdsourced annotators, such as mTurk workers, they might also be more effective at generating usable labels. To investigate this, we evaluate how law students compare to mTurkers at performing the surveys used in Calpric label requests. We prepare 600 surveys from segments that are randomly selected from OPP-115 and evenly distributed for each data category (we use OPP-115 instead of APP-350 because APP-350 does not cover as many data categories). We assign the task to 10 mTurkers and 10 law students, which is similar to the annotation setup of OPP-115 [42]. We pay the law students $31/hr[6] and use an acceptance threshold of 8/10. We tabulate the results in Table 8, which shows that the acceptance rate of the law students is slightly higher but still quite similar to that of the mTurkers (83% vs 81%). Similarly the average inter-annotator agreement rate is slightly higher for the law students but still essentially the same. As a result, law students and mTurkers produce almost the same number of accepted labels per label request, resulting in a cost per accepted label disparity of roughly 9× in favor of the mTurkers. To evaluate accuracy, we treat the labels in the OPP-115 set as the correct labels and compute the accuracy of accepted labels for the two types of annotators. We find that the both mTurkers and law students have almost perfect labeling accuracy (98% vs 99%).

6 Android App Privacy Policy Analysis

With Calpric's capabilities, we study Android App privacy policies by posing three questions as described below. We limit our study to the top 5 app categories (education, business, entertainment, music & audio, tools and lifestyle) as tracked by AppBrain[7], and add 3 minority app categories: lifestyle,

[5] https://stanfordnlp.github.io/stanza/

[6] We initially advertised a pay rate of $20/hr but increased it after we were only able to recruit 3 law students over a 4 week period. A quick search on the Internet confirmed that $31/hr is not unreasonable for law student interns.

[7] https://www.appbrain.com/stats/android-market-app-categories

shopping and health & fitness. We selected these minority categories as we believe they are more likely to use some of the minority data categories included in Calpric currently, such as social media, financial and health.

Q1: Is there a correlation between explicit denials and app popularity? We use the number of downloads and app rating as indicators for popularity and tabulate the average number of explicit denials per application in Tables 9 and 10 (in Appendix E). When considering rating, we remove apps with fewer than 500 reviews and 50K downloads as some apps with few users can have very high ratings. Regardless of whether we infer popularity by rating or downloads, we see that generally apps that are more popular have more denials. However, this trend does not hold for the most popular apps. Upon closer inspection, we find that apps in the most popular category are generally well-known apps supported by large companies and organizations, and as such support a wide range of functionality, resulting in more handling of private information. For instance, Facebook, a representative example of such an app, contains social media functionality, and has functions associated with personal and demographic information. As such, they cannot have explicit denials of collecting and using this information.

Q2: Which data actions have more of the rare explicit denial and choice labels? We tabulate the frequency of assertions, denials and choice labels by data action in Table 11. From this, we can see that overall, assertions are the most frequent, followed by explicit denials, and lastly choice labels. This is reasonable as the primary function of privacy policies is to disclose how they handle private user data. We also find that sharing has the highest number of denial and controllable statements, suggesting that app developers may feel that users are most concerned about this data action. Sharing is often associated with non-critical functionality, such as advertising, analytics or social media, and thus can be excluded more or made controllable by the user to make a potentially useful app more privacy protective. Curiously, there are more denials of storing data than assertions of it, suggesting that storage may be often implied as opposed to explicitly stated.

Q3: Are rarer data categories uniformly distributed among app categories? How does the distribution of rare data categories compare against previous data sets? We tabulate the normalized frequency of data category disclosures for each app category in Table 12 in Appendix E. In summary, common data categories (contact, device, location) are used more or less uniformly across all app categories. However, the distribution of rare data categories is less even. For example, health data is disproportionately used by health & fitness apps, and shopping apps are more likely to use financial data. As a result, we conclude that rarer data categories are more specialized and tend to be used only by specialized apps.

Second, we find that data categories that did not receive as much attention as they were perceived to be rare, are actually not so rare. We see that the use of social media data, financial data and demographic data is fairly common. We believe that lack of attention to these other categories may have been partially caused by some of the early privacy policy datasets being drawn from web pages instead of mobile apps (such as OPP-115), which resulted in under-representation and poor accuracy for those data categories [19].

7 Limitations and Threats to Validity

Because Calpric relies on crowdsourced labels, it cannot account for policy jargon having inconsistent meanings across annotators. For example, sharing has specific meanings in CCPA[8], which may differ across jurisdictions and that crowdsourced annotators may not be aware of. We note that our evaluations against OPP-115 and APP-115 do not take into account changes in laws that have taken place between the time those datasets were collected and the time we developed Calpric. As Calpric is trained on segment texts, it does not take into consideration definitions appearing in policy headers. Similarly, it cannot take into account text that is not in the privacy policy, such as consent requests in the application itself. While Calpric currently only works on Android privacy policies, we believe the same approach can be used on policies from other sources, such as websites and IoT devices. Finally, Calpric does not have a hierarchy of data categories like PolicyLint/PoliCheck [2, 3] does, causing some contradictory labels (i.e. collection of phone number and denial of collecting emails, which are both contact information). We believe Calpric can support hierarchical categories and leave this for future work.

8 Related Work

Privacy Policy Analysis: In contrast to previous studies, we present a system that automatically collects diverse training samples before labeling and support precise information extraction and summarization on a broad range of categories. Most existing automated systems analyze privacy policies in a coarse granularity, focusing on one-level categorical taxonomy. Watanabe et al. [41] used keyword-based search model to identify non-compliance between mobile apps and their privacy policies. Costante et al. [9] presented a solution to automatically assess the completeness of a policy using more complex algorithms such as Linear Support Vector Machines (LSVM). Wilson et al. [42] created OPP-115, a data set of 115 website privacy policies labeled by legal experts. Frederick et al. [27] presented a performance comparison among Logistic Regression (LR), Support Vector Machine (SVM), and Convolutional Neural Network (CNN) models trained on OPP-115, focusing on the categorical levels only. Zimmeck et al. [50]

[8] https://oag.ca.gov/privacy/ccpa

developed a document-level compliance checking tool that focuses on 9 privacy requirements, covering 3 data categories, 2 data actions and 2 notices. Recent works began to explore sentence or segment-level classification in privacy policies. Harkous et al. [19] developed a multi-label classifier using a CNN to label policies using the OPP-115 taxonomy, which covers both the top-level categories and the second-level attributes. However, they suffer from low accuracy for minority classes such as health due to lack of training samples. On the other hand, Zimmeck et al. [49] created a labeled mobile app-specific privacy policy corpus, APP-350, and trained SVM classifiers to analyze privacy requirements. To compensate the rarity of denial annotation labels, they created synthetic data by manually changing assertions into denials. In contrast, we overcome class imbalance by mining a large unlabeled dataset for potential denial samples and getting them labeled with active learning.

Wilson et al. [43] investigated and confirmed the viability of extracting labels from privacy policies through crowdsourcing. While they focus solely on crowdsourcing methodologies, we extend our work to the classification of privacy policy. Despite crowdsourcing being a cheaper option, on average, they reported a cost of $60 to label each privacy policy. Our system is able to further reduce the average cost of $13.5. Zimmeck et al. [48] also explored crowdsourcing, but did not train their ML models on these labels. They did, however, bring up an important issue of privacy policies collected via crowdsourcing: the low inter-annotator agreement among labels. In both prior works, workers are given an entire privacy policy. To create a label, they first need to read through the document, select and highlight a segment of texts among all texts, and then do detailed annotation based on the specific taxonomy. Because workers may select texts differently based on their own interpretations, it is difficult to measure the inter-annotator agreement and filter out labels with low confidence. To overcome this issue and further reduce the labeling cost, we simplify the crowdsourcing tasks by designing a novel segmentation algorithm and survey-based tasks. PolicyLint [2] and PoliCheck [3] use an ontology to organize information types as a hierarchy and identify conflicts caused by coarse-grained labels, similar to what we discussed in Section 5.2.3. However, instead of supervised deep learning, they employ statistical name-entity recognition.

Active Learning: In general, there are three active learning scenarios: membership querying synthesis, stream-based, and pool-based selective sampling. Pool-based selective sampling has been the most well-studied scenario, especially for text classification [40] and information extraction [36]. The major advantage of this method is that it evaluates and ranks the entire set of unlabeled training points before selecting the next one to label [35]. We select pool-based sampling because it is applicable to Calpric's scenario as we have the entire unlabeled training set up front and it is the most effective and widely used sampling mode for text classification.

To our knowledge, Calpric is the first deep active learning classifier for privacy policies. There are related studies using similar learning approaches in areas such as image analysis and classification [16, 45]. Zhang et al. [46] implemented active learning strategies on top of CNNs for sentiment analysis, whereas Shen et al. [38] investigated uncertainty-based active learning heuristic for sequence tagging on a newly proposed CNN-CNN-LSTM architecture. We are also the first paper that explores active learning on multi-label BERT models [11]. Although there are recent papers studying active learning on the, they focus solely on single-label problems [15, 18], where each data is restricted to have one label.

Unreliable Oracles: Most existing solutions on unreliable oracles in active learning require a confidence score for the oracle and also assume there exist at least some reliable oracles to use (though perhaps at a higher cost) [13]. In other words, the assumption is still impractical especially in the setup where annotation tasks are complex and time-consuming. A more recent study realizes the fact that the noise in human oracles may be non-uniformly distributed, yet the solution still relies on the confidence level [14]. In privacy policy classification, when using crowdsourced labels, it is difficult to derive an accurate estimation of the confidence score. There are also studies on crowdsourcer reliability and crowdsourced label quality [14, 28, 33]. However, they either do not involve machine learning or do not apply to an active querying setup.

9 Conclusion

In this work, we simultaneously address both problems of cost and data imbalance through a combination of automatic text selection and segmentation, crowdsourcing and active learning. In analyzing the properties of our solution, Calpric, we find all three components are essential to its success. Automatic text selection and segmentation are crucial to be able to reliably use crowdsourced annotators, as shown by our comparison of automatically segmented labeling tasks with other alternatives used by previous work (Section 5.2.2). Enabling the use of crowdsourced annotators gives a 9× saving in cost over trained annotators such as law students (Section 5.4). Finally, having cheaper, reliable on-demand annotators enables the use of active learning, which further contributes to decreasing the cost by selecting a more optimal set of segments for labeling to achieve greater accuracy and improved balance across data categories and action modes.

Acknowledgments

We thank the shepherd and anonymous reviewers for their feedback. This research was supported by a Tier 1 Canada Research Chair in Secure and Reliable Systems, Telus Corp, and NSERC Grants RGPIN-2018-05931 and CRDPJ 535902-18. Wenjun Qiu is partially supported by an SRI fellowship.

References

[1] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. Active Learning and Crowd-Sourcing for Machine Translation. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May 2010. European Language Resources Association (ELRA).

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

[3] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. Actions speak louder than words: Entity-Sensitive privacy policy and data flow analysis with PoliCheck. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 985–1002. USENIX Association, August 2020.

[4] Yonatan Bilu, Daniel Hershcovich, and Noam Slonim. Automatic claim negation: Why, how and when. pages 84–93, 01 2015.

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python*. O'Reilly, Beijing ; Cambridge [Mass.], 1st ed edition, 2009. OCLC: ocn301885973.

[6] Thiago N.C. Cardoso, Rodrigo M. Silva, Sérgio Canuto, Mirella M. Moro, and Marcos A. Gonçalves. Ranked batch-mode active learning. *Information Sciences*, 379:313–337, February 2017.

[7] Shayok Chakraborty. Asking the Right Questions to the Right Users: Active Learning with Imperfect Oracles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3365–3372, April 2020.

[8] Lydia B. Chilton, Greg Little, Darren Edge, Daniel S. Weld, and James A. Landay. Cascade: crowdsourcing taxonomy creation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1999–2008, Paris France, April 2013. ACM.

[9] Elisa Costante, Yuanhao Sun, Milan Petković, and Jerry den Hartog. A machine learning solution to assess privacy policy completeness: (short paper). In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society - WPES '12*, page 91, Raleigh, North Carolina, USA, 2012. ACM Press.

[10] Aron Culotta and Andrew Mccallum. Reducing labeling effort for structured prediction tasks. In *Proceedings of the National Conference on Artificial Intelligence*, volume 2, pages 746–751, 01 2005.

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

[12] Gaël Dias, Elsa Alves, and José Gabriel Pereira Lopes. Topic segmentation algorithms for text summarization and passage retrieval: An exhaustive evaluation. In *AAAI*, 2007.

[13] Pinar Donmez and Jaime G. Carbonell. Proactive learning: cost-sensitive active learning with multiple imperfect oracles. In *Proceeding of the 17th ACM conference on Information and knowledge mining - CIKM '08*, page 619, Napa Valley, California, USA, 2008. ACM Press.

[14] Jun Du and Charles X. Ling. Active learning with human-like noisy oracle. In *2010 IEEE International Conference on Data Mining*, pages 797–802, 2010.

[15] Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online, 2020. Association for Computational Linguistics.

[16] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. *arXiv:1703.02910 [cs, stat]*, March 2017. arXiv: 1703.02910.

[17] Goran Glavaš, Federico Nanni, and Simone Paolo Ponzetto. Unsupervised Text Segmentation Using Semantic Relatedness Graphs. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 125–130, Berlin, Germany, 2016. Association for Computational Linguistics.

[18] Daniel Grießhaber, Johannes Maucher, and Ngoc Thang Vu. Fine-tuning BERT for Low-Resource Natural Language Understanding via Active Learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1158–1171, Barcelona, Spain (Online), 2020. International Committee on Computational Linguistics.

[19] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. *arXiv:1802.02561 [cs]*, June 2018. arXiv: 1802.02561.

[20] Henry Hosseini, Martin Degeling, Christine Utz, and Thomas Hupperich. Unifying privacy policy detection. *Proceedings on Privacy Enhancing Technologies*, 2021(4):480–499, 2021.

[21] Justin M. Johnson and Taghi M. Khoshgoftaar. Survey on deep learning with class imbalance. *Journal of Big Data*, 6(1):27, December 2019.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980 [cs]*, January 2017. arXiv: 1412.6980.

[23] Aniket Kittur, Boris Smus, and Robert Kraut. CrowdForge: crowdsourcing complex work. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*, page 1801, Vancouver, BC, Canada, 2011. ACM Press.

[24] Matt J Kusner, Yu Sun, Nicholas I Kolkin, and Kilian Q Weinberger. From word embeddings to document distances. *Proceedings on International Conference on Machine Learning*, page 10, 2015.

[25] David D. Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pages 148–156. Elsevier, 1994.

[26] David Lie, Lisa M. Austin, Peter Yi Ping Sun, and Wenjun Qiu. Automating accountability? privacy policies, data transparency, and the third party problem. *University of Toronto Law Journal*, page e20200136, December 2021.

[27] Frederick Liu, Shomir Wilson, Peter Story, Sebastian Zimmeck, and Norman Sadeh. Towards automatic classification of privacy policy text. page 5, 2018.

[28] Matt Lovett, Saleh Bajaba, Myra Lovett, and Marcia J. Simmering. Data quality from crowdsourced surveys: A mixed method inquiry into perceptions of Amazon's Mechanical Turk Masters. *Applied Psychology*, 67(2):339–366, April 2018.

[29] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, 2013.

[30] Michal.Danilk Nakatani Shuyo. langdetect: Language detection library ported from Google's language-detection.

[31] Matthew E. Peters and Dan Lecocq. Content extraction using diverse feature sets. In *Proceedings of the 22nd International Conference on World Wide Web - WWW '13 Companion*, pages 89–90, Rio de Janeiro, Brazil, 2013. ACM Press.

[32] Joel R. Reidenberg, Jaspreet Bhatia, Travis D. Breaux, and Thomas B. Norton. Ambiguity in Privacy Policies and the Impact of Regulation. *The Journal of Legal Studies*, 45(S2):S163–S190, June 2016.

[33] Steven V. Rouse. A reliability analysis of Mechanical Turk data. *Computers in Human Behavior*, 43:304–307, February 2015.

[34] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden Markov models for information extraction. In G. Goos, J. Hartmanis, J. van Leeuwen, Frank Hoffmann, David J. Hand, Niall Adams, Douglas Fisher, and Gabriela Guimaraes, editors, *Advances in Intelligent Data Analysis*, volume 2189, pages 309–318. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[35] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, June 2012.

[36] Burr Settles, Mark Craven, and Lewis Friedland. Active learning with real annotation costs. *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 01 2008.

[37] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, January 2001.

[38] Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada, 2017. Association for Computational Linguistics.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 06 2014.

[40] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, March 2002.

[41] Takuya Watanabe, Mitsuaki Akiyama, Tetsuya Sakai, Hironori Washizaki, and Tatsuya Mori. Understanding the inconsistency between behaviors and descriptions of mobile apps. *IEICE Transactions on Information and Systems*, E101.D(11):2584–2599, November 2018.

[42] 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. 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, Berlin, Germany, 2016. Association for Computational Linguistics.

[43] Shomir Wilson, Florian Schaub, Rohan Ramanath, Norman Sadeh, Fei Liu, Noah A. Smith, and Frederick Liu. Crowdsourcing annotations for websites' privacy policies: Can it really work? In *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, pages 133–143, Montreal, Quebec, Canada, 2016. ACM Press.

[44] Bishan Yang, Jian-Tao Sun, Tengjiao Wang, and Zheng Chen. Effective multi-label active learning for text classification. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 917, Paris, France, 2009. ACM Press.

[45] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z. Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation, 2017.

[46] Ye Zhang, Matthew Lease, and Byron C. Wallace. Active discriminative text representation learning. *arXiv:1606.04212 [cs]*, December 2016. arXiv: 1606.04212.

[47] L. Zhao, G. Sukthankar, and R. Sukthankar. Incremental relabeling for active learning with noisy crowdsourced annotations. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 728–733, 2011.

[48] Sebastian Zimmeck and Steven M. Bellovin. Privee: An architecture for automatically analyzing web privacy policies. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 1–16, San Diego, CA, August 2014. USENIX Association.

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

[50] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven M. Bellovin, and Joel Reidenberg. Automated Analysis of Privacy Requirements for Mobile Apps. In *Proceedings 2017 Network and Distributed System Security Symposium*, San Diego, CA, 2017. Internet Society.

A Privacy Policy Scraper

Calpric requires a small set of labeled privacy policy samples to bootstrap the active learning process and a large set of unlabeled privacy policy samples as the potential training set. The Scraper downloads metadata of 375K Android applications from the Play Store, eliminates broken links by checking the HTTP return status, and filters out non-legal documents with keyword checking heuristics [19, 20]. It then detects and excludes non-English documents using Langdetect [30]. Finally, it discards duplicates and trivially short privacy policies. Building on top of the Dragnet model [31], the Scraper sanitizes HTML policies by removing irrelevant elements such as HTML tags, advertisement, and UI features (e.g., navigation bar). For now, Calpric considers privacy policies in HTML format only, though we could have easily extended the preprocessing to handle less common formats such as PDF or raw text. The process results in a total of 51,781 usable privacy policies in plain text format.

B MTurker Selection and Survey

MTurker Selection: Calpric selects mTurkers as survey participants based on requirements and tests. All participants should have an approval rate > 90 and the number of HITs approved > 500 while being an English speaker and an Android mobile user. The qualification test determines whether an mTurker has the capability to label privacy policies, in which mTurkers read through detailed annotation instructions and answer qualification questions. Calpric also ensures no worker answers the same question repeatedly by recording and checking their worker IDs before assigning HITs.

MTurk Survey: Calpric sends HITs[9] as batches of multi-question surveys to Amazon mTurk. Each survey queries a worker for labels on data category relevance and action modes for each data action for a single text segment. The format of the survey has gone through several iterations of revision to enhance clarity. The version with the highest inter-annotator agreement among 5 mTurk annotators was eventually selected. To control labeling cost, we perform some calibration to identify the optimal payment for completing a HIT. We prepare sample HITs based on labeled text segments from the APP-350 and OPP-115 datasets to evaluate how worker wages affect the inter-annotator agreement rate and accuracy of the labels. We observe a significantly low agreement rate (31.9%) for hourly payments below \$1[10], while the agreement rate fails to increase significantly for payments higher than \$5.2 per hour (76.0% for \$5.2 vs. 77.0% for \$13). Similarly, the accuracy did not increase for payments above this threshold either. Based on this, each HIT consists of roughly 40 surveys and the payment varied $\$0.16-0.25$ per survey. We also

[9] A **H**uman **I**ntelligence **T**ask is the term that the Amazon mTurk uses to describe a task.

[10] All amounts are in USD.

	50K	10K-50K	1K-10K	1K
collect/use	0.21	0.38	0.34	0.31
share	0.21	0.31	0.38	0.28
store	0.08	0.19	0.15	0.24
overall average	0.50	0.89	0.87	0.83

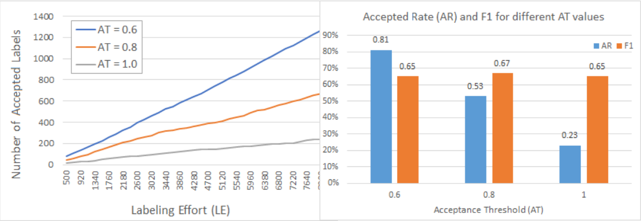
Table 9: Average # of explicit denials per privacy policy by # of downloads

	4.5	(4.0, 4.5]	(3.0, 4.0]	3.0	None
collect/use	0.26	0.35	0.33	0.33	0.40
share	0.32	0.40	0.39	0.24	0.32
store	0.14	0.24	0.04	0.10	0.19
overall average	0.72	0.99	0.76	0.67	0.91

Table 10: Average # of explicit denials per privacy policy by rating

embed an honesty checker in each survey for mTurkers to indicate whether they paid close attention and answer questions with their best effort. MTurkers receive full payments regardless of what they answer but Calpric discards all labels with "No" and re-publishes the same samples to be labeled by other qualified mTurkers. Manual inspection shows that 63% of these are indeed incorrect, while the remaining false positives may still be random guesses that fall into the correct categories by chance. The conclusion aligns with the previous study [33].

C Sensitivity study



(a) (b)

Figure 5: (a) an example result of Accepted labels vs. Labeling Requests at different Acceptance Thresholds (AT) (data category: contact); (b) average Acceptance Rate (AR) and F1 performance for different ATs

Acceptance Threshold (AT): Calpric uses an AT of $4/5$, which the inter-annotator agreement rate has to meet or exceed for the labels from the annotators to be accepted. We investigate the effect different ATs have on the acceptance rate and accuracy of the resulting model. We have the active learning select 1600 segments to be labeled, and have each of them labeled by 5 crowdsourced annotators for a total of 8000 labels. We then apply 3 different ATs: 3/5, 4/5 and 5/5 and compute the number of accepted labels, average accept rate and resulting accuracy of the trained model. Figure 5-a shows the number of accepted labels by AT. As expected, the lower the threshold, the larger the number of accepted labels, though we do observe that the variability in the number of accepted labels does increase with AT. Similarly, Figure 5-b shows that the average acceptance rate also decreases inversely with AT. However, we observe an interesting trend in the accuracy of the trained model, which peaks slightly at the 4/5 AT. We surmise that the 3/5 threshold likely has lower accuracy due to a higher rate of incorrect labels due to the more permissive AT, while the 5/5 AT has too few training points to enable the model to generalize. Interestingly, combined with our results from Section 5.1, it shows that despite discarding some number of samples that don't meet the acceptance rate, Calpric is still able to find enough representative samples to achieve higher classification accuracy than previous works that use all samples from trained annotators.

Relabeling: We evaluate the trade-off between Label & Discard and Incremental Relabeling, as described in Section 4.3. We introduce a measure, the Relabeling Success Rate (RSR), which we define as the fraction of wasted labels that are recovered as accepted after relabeling. We take 100 wasted labeling requests and republish them on mTurk to compute the RSR. Out of the 100 relabeled segments, only 2 passed the AT after $N = 2$ and only 1 passed the AT after $N = 3$, resulting in an overall RSR of 3% for 3 relabeling attempts. This result heavily suggests that Incremental Relabeling is not an effective use of labeling resources.

This result is consistent with our earlier measurements, which suggest that the labels produced by an agreement rate of 4/5 across 5 mTurkers are reliable, as they also largely agree with labels produced by trained annotators. It is not necessary, and does not help, to gather labels from a larger set of mTurkers. Indeed, further investigation suggests that the wasted labeling requests may be on text segments that are inherently ambiguous. We also observe that the segments that end up wasting labeling resources tend to be longer and more complex, which contributes to a higher possibility of multiple interpretations. Specifically, we find that the average length of accepted segments is 380.4 characters while that of wasted segments is 439.2, a difference of roughly 15%. One of the wasted text segments is:

> *"Health Plan may also partner with your Group Health Plan, if you obtain benefits through such a Plan, in which the Group Health Plan is responsible for some of the information being provided through the Health Plan site."*

	assert	denial	choice
collect/use	5.06	0.90	0.63
share	1.41	1.07	0.89
store	0.33	0.49	0.13
overall freq.	6.80	2.46	1.65

Table 11: Average # of labels per privacy policy by data actions and action modes

app category data category	education	entertainment	music & audio	tools	shopping	health & fitness	business	average
contact	0.93	0.92	0.77	0.63	1.00	1.00	0.57	0.82
device	0.26	0.59	0.32	0.55	0.23	0.82	0.37	0.42
location	0.16	0.30	0.30	0.52	0.85	0.47	0.12	0.38
demographic	0.69	0.14	0.13	0.05	0.52	0.73	0.05	0.31
health	0.03	0.02	0.00	0.03	0.00	1.13	0.22	0.18
financial	0.31	0.65	0.65	0.18	0.93	0.32	0.08	0.40
survey	0.08	0.02	0.02	0.02	0.02	0.05	0.03	0.03
personal identifier	0.13	0.08	0.02	0.00	0.08	0.03	0.02	0.05
social media	0.54	0.79	0.70	0.17	0.67	0.32	0.23	0.45

Table 12: Average number of text segments per app for each data category by Android app category

Both the grammar and meaning of this sentence are very ambiguous. We find it notable that roughly 27% of segments selected by Calpric fall into this ambiguous category and believe investigating this is interesting future work.

Unlabeled pool: We study how the size of an unlabeled training pool affects the training process. We experiment with Calpric using two different pool sizes: one that contains the entire pool of all 3311K segments from 52K privacy policies and another that contains only 12K randomly selected segments. Here we focus on contact, device and location data categories for the *collect/use* data action, which results in approximately 4K segments for each category. Otherwise, the two models share the exact same structure and hyperparameters. We train each model with an early stop when the number of training samples equals to 2000. We repeat each experiment five times and calculate the average accuracy. We observe a lower accuracy for all models trained using a smaller unlabeled pool: an average F1 of 86.7% for models using the full-size unlabeled pool versus an average F1 of 90.2% for models using the 12K unlabeled pool. Recall that the majority class is the assertion class (i.e. we are collecting) and the minority class is the denial class (i.e. we do not collect). We found that the accuracy difference is mostly in the minority class, where the model with the smaller unlabeled pool is more likely to misclassify true minorities as false majorities. Figure 6 compares the percentage of minority samples of the device data category as samples for the training set are selected. In both the full and 12K small pool, we initially observe a fast increase in minority samples as the active learning selects the minority samples for labeling. However, for the smaller pool, the minority percentage soon starts to decrease as active learning exhausts all the minority samples in the pool and is forced to select those in the majority class. As a result, the model does not generalize as well for the minority class, resulting in lower accuracy. For the large pool, the makeup remains balanced at approximately half in both classes. This is also confirmed by the final minority percentage as shown in Table 6. All ending percentages of models trained with the full-size unlabeled pool converge to approximately 50%, achieving a balance between the majority and minority classes.

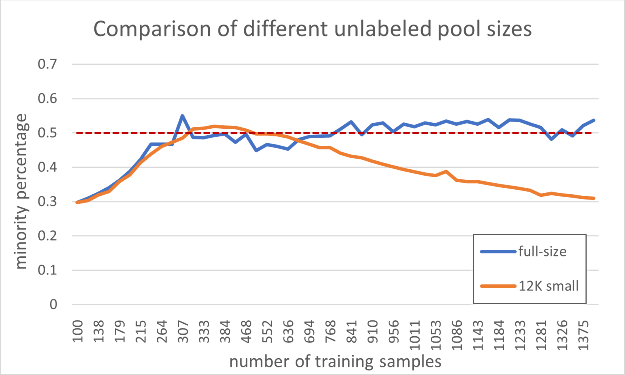


Figure 6: Comparison between Contact Category Models trained on different sizes of the unlabeled training pool. The red dotted line shows the balance line of 50% minority sample percentage

D Active Learning Querying Strategies

Basic Uncertainty selects the least certain instance from the unlabeled set and requests for it to be labeled [10]. The confidence probability is calculated using:

$$\underset{y\in Y}{max}[1-\mathbb{P}(y\in Y|\mathbf{x})] \tag{1}$$

Margin Sampling selects instances where the difference between the first most likely and second most likely classes are the smallest, minimizing the classification error [34]. **Entropy Sampling** chooses samples with the largest entropy in class probabilities, minimizing the log loss [37]. For binary classification, both are reduced to the basic uncertainty strategy. The three methods will query instances with a class posterior closest to 0.5, the most *ambiguous* segments [35]. We confirmed this conclusion using our CPPS dataset.

E Android App Analysis

Table 9, Table 10 and Table 12 show the frequency of explicit denials by number of downloads, app rating, and app category, respectively. Table 11 summarizes the number of labels by data actions and action modes.