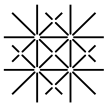

University of Basel

Privacy Policy Analysis with Large Language Models

Master thesis

Natural Science Faculty of the University of Basel
Department of Mathematics and Computer Science
Research Group Privacy-Enhancing Technologies
https://pet.dmi.unibas.ch/en/

Examiner: Prof. Dr. Isabel Wagner
Supervisor: Shiva Parsarad, PhD candidate

Maryem Fatima
maryem.fatima@stud.unibas.ch
21-067-079

17th July, 2025

Acknowledgments

I want to start by thanking Prof. Isabel Wagner for giving me the opportunity to work under her supervision on my Master's thesis. Her expertise and encouragement throughout this thesis helped me tackle the challenging problems in our research. I am grateful for the positive feedback I received during the group meetings, which always motivated me to push forward.

I am also deeply thankful to my supervisor, Shiva Parsarad. From the start till the end, she helped me pave the path for my thesis research. Her detailed feedback on my thesis drafts really helped me improve my writing skills. Overall, her guidance throughout the thesis taught me the right way of conducting research and made this journey much smoother.

Lastly, I want to thank my family and friends for being there for me throughout this time. Their support and understanding helped me get through the hard times.

Abstract

Privacy policies are complex documents that explain the collection and sharing of user data with third parties. Analysis of these documents is challenging and often impossible for a user with little to no knowledge of legal terms. However, they are of significant importance as they explain the collection and sharing of user data with third parties. In this thesis, we investigate the utility of large-language models for the comprehensive analysis of these privacy policy documents on a large scale.

We examined five well-known LLMs on two labeled datasets, OPP-115 and MAPP. The models involved in the research are DeepSeek-R1-Distill-Qwen-32B, Mixtral-8x7B, LLaMA-3.1-8B, Grok 3 Beta, and Gemini 2.5 Flash. We evaluated the performance of each model on 12 different types of user personal data in both single-attribute (per-attribute) and multi-attribute (joint) inference scenarios. We discussed how the role of structured output during the inference pipeline, temperature adjustment, and few-shot learning collectively impact the model's accuracy.

The results show that structured prompts and conservative inference parameters (such as a temperature of 0) improve model performance. The LLM performed better with categories that had clear and specific terms, such as "cookies and tracking elements," but struggled with broader categories, like "personal information," which used more vague or overlapping language. Fine-tuning further enhances performance for high-priority labels, but prompt design remains the most effective method to optimize performance.

Table of Contents

1 Introduction

1.1 Problem Statement

According to The General Data Protection Regulations (GDPR) [1], all businesses that collect or process personal data of individuals in the EU are required to inform consumers about how their data is collected, processed, stored, and shared. Typically, these disclosures are provided to the user during the sign-up process for websites, mobile applications, and online services.

Users generally ignore these privacy policies, even though these are a crucial way to understand how their data is used. One survey indicated that 74% of participants didn't read the privacy policy at all, and 97% consented to the conditions without reading them [2]. Even people who attempted to read only spent an average of 73 seconds on a policy that should have taken around 30 minutes to read at the average adult reading speed. Most of the people who participated didn't notice the terms that were intentionally included, which mentioned exchanging data with employers or even offering a firstborn kid as payment [2]. This shows how often people agree to privacy agreements without really reading and understanding them.

The complicated language of privacy policies often discourages people from reading them. In one study, where Flesch-Kincaid Grade score was calculated for a subset of privacy policies of famous websites. This score was greater than 12 which means that the degree required to understand these policies is college or university. [3]. This makes privacy policies hard for many people to read and goes against informed consent principles. [4]. The fundamental challenge stems from the gap between how important privacy policies are for user protection and how difficult they are to understand in practice. Privacy policies are typically buried in long, unclear, and opaque text that lacks user-friendly formatting or structure. The terminology is often unclear, overly technical, and incomprehensible to most users, making it nearly impossible to understand key information about how personal data is collected, used, or shared [3].

This readability problem poses significant risks beyond mere inconvenience. When consumers cannot comprehend privacy regulations, they are more likely to agree to harmful data practices unknowingly, making them vulnerable to data breaches, unauthorized sharing, or misuse of their personal information [5]. Consequently, this accessibility gap creates

several serious problems for both users and the broader digital ecosystem:

1. **Breach of users' data:** Users don't know exactly their data is gathered or shared with third parties. Applications that have privacy policies that are hard to read are far more inclined to share a lot of data with third parties [6].

2. **Problems with regulators:** It is hard for regulators to check that companies are following privacy regulations or enforce them because privacy policies use complicated and unclear wording. Privacy policies that use imprecise language are harder to enforce and harder to keep an eye on [7].

The General Data Protection Regulation (GDPR) in Europe [1] and the California Consumer Privacy Act (CCPA) in the United States [8] both require that companies should be transparent about how they handle data. However, even with these legal frameworks in place, users continue to struggle with understanding privacy policies. Reading these policies is not only time-consuming for most people but also ineffective for informed decision-making. In another study if users were to read every privacy policy they encounter, it would take approximately 244 hours annually per person [9], making manual review entirely impractical. In particular, determining if and how personal data is shared with third parties remains one of the most critical and unresolved challenges in digital privacy, as users often cannot identify these practices from complex policy language [10].
Our research investigates how large language models perform in extracting important information from privacy policies, specifically focusing on third-party data sharing practices. We explore whether LLMs can bridge the gap between legal policy documents and everyday digital services that users use by automatically extracting and simplifying critical disclosures in privacy policies, particularly those concerning data sharing with third parties. This approach aims to analyze the capability of LLMs in terms of privacy policy data.
For this analysis, we employed several state-of-the-art large language models, including DeepSeek-r1-distil-32b, Mixtral, Gemini, Llama, and Grok, to analyze privacy policies and extract third-party data-sharing information. Our methodology involved developing prompting strategies and evaluating different frameworks to assess the models' accuracy in identifying data collection, usage, and sharing practices. The results demonstrated that LLMs can help us in extracting third-party data information compared to traditional NLP approaches, with Grok and Gemini showing better quality analysis in identifying complex sharing relationships and data categories. However, our findings also revealed challenges in handling ambiguous policy language and ensuring consistent performance across different policy structures and domains.

1.2 Research Objectives

The goal of the thesis is to investigate how well large language models DeepSeek, Mixtral, Gemini, Llama, and Grok can read and understand privacy policies. We assessed the effectiveness of LLM in determining whether a privacy policy explicitly or implicitly states that users' data will be shared with third parties. To achieve this, we create prompts that includes 12 distinct personal information types, including contact and demographic details,

financial data, and biometric identifiers. We used two privacy policy corpora, MAPP and OPP_115 [11, 12] in our thesis. The main goals of the research are as follows.

1. Investigate how well LLMs like Gemini, DeepSeek-R1-Distill-Qwen-32B, Grok, llama, and Mixtral can extract information from privacy policies regarding the exchange of data with third parties . LLMs that are part of this study are explained in Section 3.3.

2. To find out which configurations for each LLM work better, for example, Joint inference (looking at all sorts of information at once) or individual inference (looking at one type of information at a time). Different information types are explained in Section 3.5.

3. Investigate which prompt engineering methods work well for the analysis and how LLMs behave when different numbers of few-shot examples are used.

4. To find out which kind of personal information is easiest to find and which is harder for LLM-based analysis to find. Is there any change in LLM response quality with the change in personal type information? For instance, does LLM behave the same for all the information types or perform differently with a different information type?

1.3 Significance of the Research

This study addresses a critical gap in understanding how different large language models perform across varied datasets and configurations for privacy policy analysis, particularly for third-party data sharing extraction. The significance of this research lies in its comprehensive evaluation of LLM capabilities under different operational conditions:

Comparative LLM Performance Analysis: Our research systematically evaluates multiple state-of-the-art language models(see section 3.3) across two distinct privacy policy datasets with varying characteristics and complexity levels. This comparative analysis reveals how model architecture, size, and training approaches influence performance on legal document analysis tasks, providing crucial insights for selecting appropriate models for privacy policy applications

Configuration Sensitivity Assessment: We investigate different prompting strategies, for instance, the effect of temperature variation on the analysis, and the effect of few-shot examples in the prompt. We also analyzed the effect of in-context definitions on the overall efficiency of the results. This analysis demonstrates the sensitivity of LLMs to configuration choices and provides guidance for optimizing the model configuration for inference in real-world privacy policy analysis scenarios [13, 14].

Dataset Evaluation:We investigate MAPP and OPP_115 datasets for third-party data collection analysis. The analysis revealed that the datasets were imbalanced with respect to third-party data sharing and collection labels, with significantly fewer examples of third-party disclosures compared to first-party disclosures. The broader implications of this research extend to multiple stakeholder groups:

For users: There is a need for a process that would make it easier for users to understand the essence of privacy policy [4]. Our study is in the direction of making it possible to automatically look at privacy policies.

For regulators, LLM-based methods can also help in finding sensitive data sharing practices on a large scale and keeping an eye on compliance more effectively, as suggested by prior work on automated policy analysis [15].
For website operators: Knowing how automated systems read their privacy policies could help them write policies that are easier to understand and more open.

1.4 Approach Overview

In this thesis, we extended the foundational work [16] done on privacy policy analysis on a large scale. This foundational research established a comprehensive framework for evaluating large language models' capabilities in privacy policy analysis by systematically comparing multiple LLM architectures GPT-4 Turbo, Llama2 with SVC classifiers across the MAPP and OPP-115 datasets for automated extraction of personal information collection practices. The study demonstrated that LLMs could effectively identify and categorize data collection statements with significant improvements over traditional NLP approaches. Their comparative evaluations resulted in an F1 score of 84.1% for ChatGPT and 86% for the SVC classifier. Even though SVC classifiers are more accurate and high in F1 score. But ChatGPT resulted in a comparable level of performance. ChatGPT offered a comparable performance with the convenience of usability that makes it viable in similar tasks. This foundational work was primarily focused on data collection and practices of information type withoutthe explicit mention of First party and third party, leaving a gap in understanding how LLMs perform in analyzing the third-party data-sharing relationships, data transfers to external entities, and multi-party privacy obligations that characterize modern digital ecosystems. Their study was also only focused on Llama 2 and and GPT-4 Turbo.
For this thesis, we used two benchmark datasets: MAPP (Multilingual Annotated Privacy Policies) [11] and OPP_115 (Online Privacy Policies) [12]. Both of these datasets have privacy rules that have been marked for third-party data sharing, which makes it possible to evaluate LLMs in this respect. In our thesis, we run an inference pipeline on our models to check how different LLMs predict the sharing and collection of personal information with third-party entities.

2
Literature Review

Privacy policy analysis has undergone significant evolution over the past two decades, driven by the exponential growth of digital platforms and online services [17] and the increasing complexity of data protection regulations [18, 19]. This chapter examines the progression from manual review processes of privacy policies to sophisticated automated systems powered by natural language processing (NLP), support vector machine classifiers, and large language models (LLMs). The review traces key developments in methodology, dataset creation, and technological advancements that have shaped the current landscape of privacy policy analysis.

2.1 Privacy Policy Analysis: Evolution and Challenges

The rapid growth of digital platforms in the late 1990s and early 2000s led to a surge in the collection and processing of personal data, raising concerns about user privacy and transparency. This prompted researchers across disciplines, including law, computer science, health, e-commerce, privacy engineering, etc., to study privacy policies as a means of understanding data handling practices [20, 21]. Early studies on privacy policies focused on evaluating the readability and accessibility of these documents, highlighting that most policies required college-level reading skills and were rarely understood by typical internet users without any idea of the legal terms of privacy policies. [9]. As the volume and complexity of privacy policies increased, manual analysis by researchers and legal professionals became time-consuming and error-prone, prompting the development of automated methods [22]. This shift toward automation necessitated the adoption of computational approaches, particularly natural language processing techniques, to extract structured information from policy texts[23].

An overview of the evolution of privacy analytics tools is provided in the Table 2.1. In the following, we explain these methods in more detail.

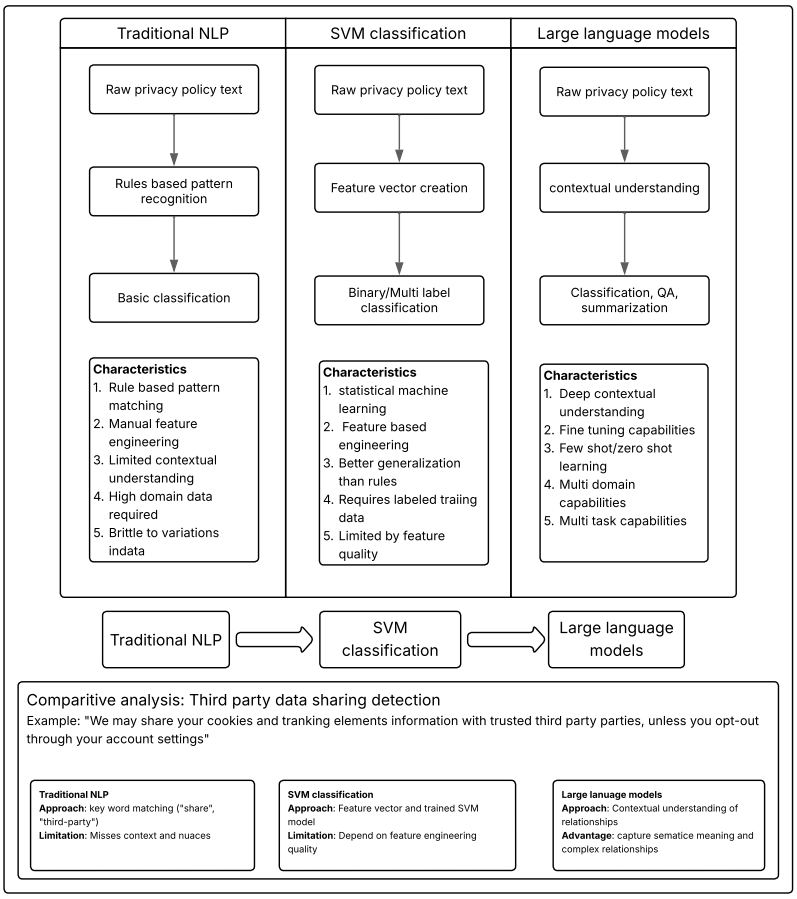

Figure 2.1: Evolution in the analysis of privacy policy tools

2.1.1 Traditional NLP-Based Approaches and Limitations

The manual analysis of the privacy policy was time-consuming. This problem led to the creation of automated systems that can help users to summarize the text of privacy policies. [24, 25]. These automated systems based on NLP for the analysis of privacy policies used keyword extraction and rule-based algorithms to find out how data was collected and shared [22]. These methods couldn't scale as the language used in privacy policies is sometimes vague and ambiguous, which made these systems struggle when they encountered such vague keywords specifically designed to hide the data usage statements. [26].

In another tool, ATPChecker [10], a compliance analysis tool, was built to verify whether Android applications' data collection and usage behaviors match the practices disclosed in their privacy policies. It uses NLP tools for the privacy policy analysis and static analysis of bytecode of Android apps to analyze the interaction of apps with third-party libraries. The research discovered that almost 31% of third-party libraries do not provide privacy policies. and 39% of third-party libraries provide privacy policies on how they conceal data usage. The crux was that over 65% of host apps violate the regulation requirements for clearly disclosing data interactions with third-party libraries (TPLs). In the limitations of their study, they have mentioned that because of NLP analysis of privacy policy, there is a

possibility that the system would fail to detect the correct usage if the privacy policies include some patterns that adversaries have used to hide their data usage statements. This means that NLP systems struggled with the complex, multi-layered relationships in third-party data sharing, motivating the need for more advanced language understanding capabilities that could handle contextual nuances and semantic relationships beyond keyword matching and rule-based systems.

2.1.2 Support Vector Machine Classifiers

To address the limitations of NLP tools, supervised machine learning approaches like support vector machines were employed. These supervised machine learning approaches needed a vast dataset with labels to classify the policy segments [27].

Polisis is a big step forward in automated privacy policy analysis. It uses hierarchical neural-network classifiers to label policy segments with 10 high-level and 122 fine-grained privacy classifications [28]. The framework shows how it may be used in real life by using automatic privacy icon assignment (which gets 88.4% of the responses right) and PriBot a similar question-answering system that gives the right answers to 89% of user questions concerning privacy regulations. Polisis does, however, have some problems. For example, it relies on predefined taxonomies that may not include new privacy practices, it sometimes misclassifies things and needs confidence scoring mechanisms, and it often fails to analyse policies that are made to trick automated classifiers. These problems show how hard it is to make strong automated privacy policy tools. This gap encourages future research into more advanced LLM-based methods.

In one of the comparative studies done on machine learning algorithms for privacy policy analysis, they evaluated twelve traditional machine learning algorithms for privacy policy classification. [29] evaluated twelve traditional machine learning algorithms for privacy policy classification on the OPP-115 dataset. The study found that Support Vector Machines achieved the best performance with 79% accuracy and 0.79 F1-score, followed closely by Logistic Regression with 78% accuracy and 0.78 F1-score. The authors concluded that traditional ML approaches alone were insufficient to exceed 81% accuracy on the OPP-115 dataset due to its small size and class imbalance, suggesting the need to combine these methods with more advanced techniques such as language models or deep learning.

2.1.3 Large Language Models and their Applications in Privacy Policy Analysis

The development of transformer-based models fundamentally changed the landscape of natural language processing and privacy policy analysis. Unlike traditional NLP approaches that relied on feature engineering and domain-specific rules, large language models introduced robust architectures capable of learning complex linguistic patterns from vast amounts of text data [30, 31].

BERT (Bidirectional Encoder Representations from Transformers) represented a significant advancement with its encoder-only architecture, enabling bidirectional context understanding that proved particularly valuable for privacy policy classification tasks [30]. BERT's ability to capture nuanced meanings enhanced tasks like classifying opt-out clauses and

third-party data sharing practices [32]. In contrast, GPT models employed a decoder-only architecture optimized for text generation, making them suitable for tasks requiring policy summarization and explanation generation [31]. The key advantages of LLMs over traditional NLP or SVM classifiers approaches include: (1) contextual understanding that goes beyond keyword matching through bidirectional attention mechanisms [30, 33], (2) transfer learning capabilities that reduce dependence on large domain-specific datasets by leveraging pre-trained representations[34, 35], (3) few-shot and zero-shot learning abilities that enable adaptation to new policy types without extensive retraining [13, 31], and (4) multilingual capabilities that address the cross-lingual challenges identified in earlier corpus development efforts [36, 37]. However, LLMs also introduced new challenges, including high computational requirements [38], potential biases in training data [39, 40], and interpretability concerns, particularly relevant for legal and regulatory applications [41, 42].

Adapting large language models to legal texts required specialized fine-tuning approaches using datasets like OPP-115 and MAPP. While these corpora enabled promising results when used with support vector machines, particularly in tasks like segmentation and label prediction, they remain limited in scope and coverage [43]. This highlights both the potential of LLMs for scalable policy analysis and the need for richer, more comprehensive datasets to overcome the shortcomings of earlier NLP systems.

Beyond traditional analysis tasks, large language models have enabled new applications aimed at improving user comprehension of privacy policies. One such approach is PrivCAPTCHA, which leverages few-shot prompting with LLMs to convert complex policy text into concise, interactive chunks optimized for mobile devices [44]. Inspired by CAPTCHAs, the design requires users to engage with the transformed policy content by clicking on labeled segments, effectively verifying their understanding. This demonstrates how LLMs can bridge the gap between technical policy analysis and meaningful user interaction.

Recognizing the dual nature of LLMs as both privacy analysis tools and potential privacy risks, the CLEAR system addresses user awareness when interacting with LLM-powered applications [45] Through co-design workshops, researchers identified that users often remain unaware of privacy risks when sharing information with conversational AI interfaces. CLEAR provides contextual, just-in-time privacy risk assessment by leveraging LLMs to analyze privacy policies and generate personalized risk warnings, exemplifying the paradoxical role of LLMs in privacy contexts.

Extending real-time assistance approaches, PRISMe represents the first comprehensive qualitative evaluation of LLM-driven privacy policy assessment in web browsing contexts [46] This browser extension combines automated policy analysis with interactive user interfaces, providing both dashboard visualizations and conversational LLM chat interfaces. The mixed-methods user study (N=22) demonstrated that LLM-powered real-time policy assessment can make complex privacy documents accessible to users without specialized legal knowledge, directly addressing the readability challenges identified in early privacy policy research [46] .

These user-centered approaches collectively demonstrate that LLM applications in privacy policy analysis extend far beyond traditional automated extraction and classification. They position LLMs as mediating technologies that can transform how users interact with and

understand privacy policies, addressing fundamental challenges in privacy communication that have persisted despite advances in automated analysis techniques.

2.2 Contemporary LLM Architectures used in our thesis

The comparative analysis of foundational transformer architectures with the NLP and SVM-based tools in privacy policy analysis applications has driven continued innovation in the field of privacy policy analysis using large language models. This results in more extensive experiments in this field. These large language models represent diverse approaches to balancing performance, efficiency, and deployment flexibility for privacy policy analysis.
In the following paragraph, we have explained all the large language models in detail that we have used in our thesis research. We have examined three open-source models and two proprietary cloud-based models. In some cases, we use distilled models to reduce both the memory required to load the model (memory footprint) and inference time. In others, we employ Mixture of Experts (MoE) models, which selectively activate only a subset of specialized sub-models during inference to improve efficiency and scalability. A detailed analysis of these models is as follows.

Distilled Dense Models: DeepSeek-R1-Distill-Qwen-32B represents a significant advancement in model distillation techniques, where a smaller model (student) is trained to replicate the behavior of a larger model (teacher), reducing computational requirements while preserving performance [47]. According to its Hugging Face model card, it "outperforms OpenAI-o1-mini across various benchmarks," delivering state-of-the-art results for dense models, making it valuable for international privacy policy analysis [48]. Its requires significantly lower VRAM compared to base deepseek-r1 model, enabling deployment on more accessible hardware configurations [49].

Sparse Mixture-of-Experts Architecture : Mixtral-8x7B-v0.1 exemplifies the MoE approach with 46.7 billion total parameters but only 12.9 billion active per token [50, 51]. The MoE architecture consists of multiple specialized subnetworks (experts), where a gating mechanism dynamically selects a subset of experts for each input, enhancing efficiency by activating only a fraction of parameters. This model excels in logic-intensive tasks such as coding and mathematical reasoning, crucial capabilities for parsing complex legal language and logical relationships in privacy policies [50]. Its open-source nature enables local deployment and fine-tuning capabilities essential for specialized legal text analysis.

Compact Dense Models : Llama-3.1-8B represents the trend toward efficient, high-performing models that balance accuracy with computational feasibility [52]. Developed by Meta AI for text generation, reasoning, and multilingual tasks in eight or more languages.

Proprietary Cloud-Based Models : Grok 3 Beta, developed by xAI, focuses on real-time data access and reasoning capabilities, providing insights into how proprietary models handle dynamic legal content and real-time policy updates [53]. Similarly, Gemini 2.5 Flash,

developed by Google, represents the state-of-the-art in multimodal large language models, supporting vast context windows (up to 1M tokens) optimized for text processing with powerful multilingual capabilities [54]. It is available via API on Vertex AI and allows batch processing of privacy policies.

2.3 Datasets and Annotation Frameworks

To perform all the experiments using NLP tools and machine learning approaches, there was a need for extensive annotated data that could be fed to these algorithms. Especially that the quality of both approaches depends on the quality of the datasets used. This requirement led to the creation of annotated privacy policy corpora. The OPP-115 corpus [12] marked a significant step forward, consisting of 115 English-language privacy policies annotated by law students under expert supervision. It identifies text segments related to data practices across 10 categories, such as data security and third-party sharing. While widely used, OPP-115 focuses on English policies and is limited in size, restricting its applicability. To address cross-lingual needs, the MAPP corpus [11] was introduced, comprising 64 English and 91 German policies, each annotated with 23 attribute types. These corpora enabled more accurate NLP models and, critically, provided the foundation for training more sophisticated language models. However, the limited scale and domain coverage of these datasets highlighted the need for models capable of better generalization and transfer learning across diverse policy contexts and languages.

3
Methodology

This chapter outlines the complete methodology that we followed to evaluate the ability of large language models (LLMs) in the identification of third-party data sharing practices from privacy policy documents. From dataset selection to preprocessing and model selection, every decision was guided by the goal of making this research not only accurate but also practically meaningful.

3.1 Dataset Selection

Selecting an appropriate dataset is a crucial step in evaluating the performance of a large language model, as it directly influences the quality, relevance, and reliability of the analysis. For our experiments that focused on analyzing privacy policies using large language models, we selected two well-established corpora, the OPP_115 corpus [12] and the MAPP [11] dataset. These datasets were chosen because they provide structured and annotated privacy policy texts, which are essential for evaluating how accurately and consistently a model can extract or reason over policy content.

3.1.1 OPP_115 Corpus

This corpus comprises 115 files from various websites. The dataset contains the top 5 most trending sites at the time of creation. Three legal experts annotated each policy in the dataset, and then released their overlapping results on the basis of different thresholds.
The main structure of OPP_115 is as follows.

- **Annotations**: This includes labels created by all three annotators who worked on the privacy policy.
- **Consolidations**: Each of the three subdirectories in the consolidation contains the results of the consolidation algorithm with a different convergence threshold on the annotation folder:
  - **Strict (1.0)**: Requires complete (100%) agreement among annotators.
  - **Moderate (0.75)**: Requires at least 75% agreement.

  - **Lenient (0.5)**: Requires a minimum of 50% agreement.

- **Original_policies**: This folder has the original policies without any sanitation technique. It contains the original HTML documents.

- **Sanitized policies**: This contains the text of the privacy policies of all the websites separated by "!!!" delimiters.

All policies in the dataset were extensively annotated. In cases where specific practices were not annotated, we treated them as absent, assuming that the policy does not mention the collection of that specific attribute.

3.1.2 MAPP Corpus

MAPP dataset is a bilingual privacy policy corpus having privacy policy annotations in both English and German language. Just like OPP_115, the MAPP corpus has also been annotated by multiple annotators. But the annotations available for the MAPP corpus are less fine-grained than those of OPP_115.
For instance, in OPP_115, when the annotator provided an annotation, they mentioned the specific selection of text that had the annotations. But in the MAPP dataset, this type of information is not present.
The main structure of the MAPP dataset is as follows. Apart from this, this dataset folder also has a readme and documentation, which contains the details of the annotation scheme. The breif overview of the folder structure is as follows.

- **English sanitized policies**: In this table, we have sanitized policies from different website in text file format. Each file has an ID and name of website concatenated together in the form of the file name.

- **English Consolidations**: This folder includes CSV files, which are the consolidation efforts of multiple annotators during the annotation process. For each file in the English sanitized policies folder, we have a corresponding consolidation file in this folder with the same name. The structure of this file is mentioned in Table 3.2.

- **German sanitized policies**: Similarly, this folder also contains sanitized privacy policies, but in the German language.

- **German Consolidations**: This folder also contains the consolidation efforts. But it has the consolidation of German language policies.

For the MAPP dataset, in the following table (Table 3.1), we calculated the total samples present for third-party data sharing. We calculated the count of all the types of information separately that are shared with third parties.
During the exploratory study on the MAPP dataset, we realized that the dataset is heavily unbalanced with respect to third-party information. For example, in the case of political or religious views, there is no single label for third-party data sharing, and overall, the occurance of third party labels is far less compared to first-party or other categories.

Information Type	TRAIN	TEST
Contact information	29	9
Cookies and tracking elements	68	9
Demographic data	13	2
Financial	14	6
Generic personal information	147	35
Health, genetic, or biometric data	5	3
IP address and device IDs	42	9
Location	21	5
Other	34	9
Personal identifier	8	1
Social media data	11	2
Unspecified	144	31
User online activities	62	17
Computer information	22	2

Table 3.1: Third-Party Sharing/Collection – Information Type Distribution (TRAIN vs TEST) for MAPP dataset

3.2 Data Preprocessing

After the dataset is selected for our experiments, the next step in the pipeline is preprocessing. Preprocessing helps us to format data in a desired form, which can then be fed to the models to get the output. In our case, both datasets required some level of preprocessing before they could be used for our experimentation. In the following, we first explain preprocessing steps for MAPP datasets, and then in the next section, the preprocessing steps for the OPP_115 dataset will be discussed.

3.2.1 Preprocessing for MAPP dataset

In this section, we will discuss the preprocessing steps we took for the MAPP dataset. Our main 2 folders in the MAPP dataset are as follows.

- English sanitized policies
- English Consolidations

As our focus was on extracting third-party labels. This relevant information is present in "Category Name" column of the consolidation file. The structure of the consolidation file for MAPP dataset is shown in the table 3.2.

3.2.1.1 Data extraction

Once we analyse this column and it has a Third-party label, then we move to the next column of "Attribute Name". If the attribute name mentions that it includes the information types in the privacy policy, then we check the exact information types involved in the privacy policy by analyzing the "Value Name" column. So, mainly, to extract information related to our research, we used the following columns in the consolidation files.

- **Category Name** This contains infromation category, whether its first-party or third-party or both

Table 3.2: Privacy Policy MAPP Dataset Sample

Policy ID	**Segment ID**	**Category Name**	**Attribute Name**	**Value Name**	**Policy Type**
1	1	First Party, Third-Party	Information Type, Collection Process	Collection Process_Shared by first party with a third-party, Collection Process_Collected on first-party website/app	TRAIN
1	17	First Party, Third-Party	Information Type, Purpose, Collection Process, Does/Does Not (opt), Third-Party Entity	Information Type_Generic personal information, Collection Process_Shared by first party with a third-party, Collection Process_Collected on first-party website/app	TRAIN

- **Attribute Name** This column contains the data whether the privacy policy mentions about information type, collection purpose, choice type etc. For our use case, we stayed focus or Infromation type and collection purpose. If the collection process is "Shared by first party with a third-party", and it also includes information type, then we mark that information type as being shared with third party.
- **Value Name** This contains the information type mentioned in the privacy policy along with the proces by which it is collected or shared.

3.2.1.2 Label Assignment Methodology

For the label assignment, we used the above three columns mentioned in the previous pa-pragraph. We did a step by step evaluation of whether the label exists or not. The steps we took are as follows.

Step-by-step process

1. **Filter for third-party contexts:** Check if the *Category Name* contains "Third-Party" to identify rows relevant to third-party data practices. 3.2
2. **Identify sharing activities:** Within the *Value Name* field, look for indicators of third-party sharing such as "Collection Process_Shared by first party with a third-party" or similar sharing-related processes.

3. **Extract information types:** From rows that meet both criteria above, identify the specific information types mentioned in the *Value Name* field (e.g., "Information Type_Generic personal information").

4. **Assign final labels:** Use only the extracted information types as the end labels, excluding process descriptions or other metadata. The structure of the label for policy ID 1 and segment ID 17 in table 3.2 is as follows.

```
{
 'Financial': 'NO',
 'Health_genetic_or_biometric_data': 'NO',
 'Contact_information': 'NO',
 'Location': 'NO',
 'Demographic_data': 'NO',
 'Personal_identifier': 'NO',
 'User_online_activities': 'NO',
 'Social_media_data': 'NO',
 'IP_address_and_device_IDs': 'NO',
 'Cookies_and_tracking_elements': 'NO',
 'Computer_information': 'NO',
 'Generic_personal_information': 'YES',
 'Political_religious_or_philosophical_belief': 'NO',
 'Other': 'NO',
 'Unspecified': 'NO'
}
```

After we had extracted the data and gathered all the third-party labels, we parsed through the English sanitized policies files and merged the policy segment along with its labels in the CSV file format for easy analysis during the inference pipeline. The final CSV file is formatted as shown in Table 3.3.

Table 3.3: MAPP preprocessing results

Column Name	**Description**
policy_id	Unique identifier for each privacy policy document.
segment_id	Identifier for each text segment (usually a paragraph or sentence).
label	An object that holds all the information types with a YES or NO label indicating presence or absence.
segment_text	The actual text content of the segment from the privacy policy.

3.2.1.3 Data splitting strategy:

For the MAPP dataset, the data set already has column "Policy type" as we can see in the table 3.2 which defines wherther the policy belongs to train or test. So, that is why we do not need an explicit functionality for train and test split. We just used these labels and created our test split CSV. At the end of the preprocessing pipeline for MAPP data, the test data file structure is the same as mentioned in the table 3.3
For our experiments we have used this file for prompt construction and inferance analysis.

3.2.2 Preprocessing for OPP_115 dataset

Similar to the MAPP dataset, we developed a dedicated preprocessing pipeline on the OPP_115 dataset. The main components of the pipeline are described below. As mentioned earlier, For our experiments, we selected the strictest threshold (1.0) to ensure the highest quality annotations. Since privacy policies have serious implications for user privacy, our approach prioritized precision by relying only on the most confidently annotated segments.

3.2.2.1 Data Extraction

Our focus was on third-party data collection/use. The relevant information was extracted from the CSV files. These CSV files in the consolidation folder lack headers. According to the dataset documentation:

- **Column 5** This contains the policy segment text.
- **Column 6** This includes the category type.
- **Column 7** This contains the associated annotation in JSON format.

The annotation column (column 7) against the privacy policy segment is a JSON object with multiple key-value pairs, each describing a specific aspect of the policy segment. For third-party sharing and collection, the attributes that are relevant are explained in Table 3.4.

Table 3.4: OPP_115 label structure for third party category

Attribute	**Description**
Third-Party Entity	The third-party involved in the data practice.
Does/Does Not	Indicates whether the policy explicitly states the practice is not performed. Defaults to *Does*.
Action Third-Party	Describes how the third party receives, collects, or accesses user data.
Identifiability	States whether the data is linked to the user's identity. Optional; defaults to *not selected*.
Personal Information Type	The type of information shared with or collected by the third-party.
Purpose	The stated reason for third-party collection or sharing.
User Type	Specifies if the practice applies to account holders or anonymous users. Optional.
Choice Type	Indicates if the user has choices related to the practice. Optional.
Choice Scope	Specifies the scope of user control, even if limited or unclear. Optional.

A label is marked `YES` only when the policy explicitly states that the data type is collected. If there is no explicit mention, we default to `NO`.

For accurate detection of personal information practices, we examined two specific keys in the annotation object:

- **Personal Information Type**
- **Does/Does Not**

This dual-check is necessary because some policies explicitly mention not collecting certain data types. In such cases, the annotators still record the type of personal information, but the associated value under `Does/Does Not` is `Does Not`. An example is provided below:

```
{
  "Personal Information Type": {
    "selectedText": "web beacons or store in Web Storage",
    "value": "Cookies and tracking elements"
  },
  "Does/Does Not": {
    "selectedText": "do not",
    "value": "Does Not"
  }
}
```

In the above example, we can say that the policy talks about the 'Cookies and tracking elements'. But if we check the Does/Does Not key, it clearly states that it doesn't collect this information. So, both key value pairs are required to correctly identify the presence of any personal information in the privacy policy segments.
As our goal was to identify personal information that is being shared with third-parties. The dataset defines 15 types of personal information. The frequency distribution of all the data types is shown below: in the table 3.5

Table 3.5: Frequency of Personal Information Types Across YES and NO Labels

Personal Information Type	**YES**	**NO**
Computer_information	46 (1.21%)	3745 (98.79%)
Contact	121 (3.19%)	3670 (96.81%)
Cookies_and_tracking_elements	165 (4.35%)	3626 (95.65%)
Demographic	38 (1.00%)	3753 (99.00%)
Financial	62 (1.64%)	3729 (98.36%)
Generic_personal_information	428 (11.29%)	3363 (88.71%)
Health	43 (1.13%)	3748 (98.87%)
IP_address_and_device_IDs	60 (1.58%)	3731 (98.42%)
Location	41 (1.08%)	3750 (98.92%)
Other	122 (3.22%)	3669 (96.78%)
Personal_identifier	13 (0.34%)	3778 (99.66%)
Survey_data	11 (0.29%)	3780 (99.71%)
Unspecified	573 (15.11%)	3218 (84.89%)
User_Profile	70 (1.85%)	3721 (98.15%)
User_online_activities	184 (4.85%)	3607 (95.15%)

3.2.2.2 Merging Segments

In OPP_115, multiple rows hold the annotation for one privacy policy. Against each single annotation, annotators have used a separate row. For instance, if a policy segment collects email address and contact information, both will be mentioned separately. This means that the labels for each policy could span multiple rows. This was causing redundancy, so, in order to fix the issue, we merged all rows with the same `segment_id`, combining their annotations into a single entry. This reduced duplication and helped us in streamlining the training and inference processes.

3.2.2.3 Data Splitting Strategy

Since the dataset is multi-label, each instance can have up to 15 binary labels. As the dataset does not come with predefined training, validation, and test splits, we partitioned it using a stratified sampling approach:

- **Training set:** 60%
- **Validation set:** 20%
- **Test set:** 20%

In this way, we made sure that the label distribution remained consistent across all splits (Table 3.6).

Table 3.6: Distribution of YES and NO labels for personal information types across training (60%), validation (20%), and test (20%) splits, with counts shown as YES/NO.

Personal Information Type	**Training (YES/NO)**	**Validation (YES/NO)**	**Test (YES/NO)**
Computer information	28/2247	9/749	9/749
Contact	73/2202	24/734	24/734
Cookies and tracking elements	99/2176	33/725	33/725
Demographic	23/2252	8/751	8/751
Financial	37/2237	12/746	12/746
Generic personal information	257/2018	86/673	86/673
Health	26/2249	9/750	9/750
IP address and device IDs	36/2239	12/746	12/746
Location	25/2250	8/750	8/750
Other	73/2201	24/734	24/734
Personal identifier	8/2267	3/756	3/756
Survey data	7/2268	2/756	2/756
Unspecified	344/1931	115/644	115/644
User Profile	42/2233	14/744	14/744
User online activities	110/2164	37/721	37/721

3.2.2.4 Privacy Policy Text Preparation

The privacy policy text is present in sanitized HTML files. Each privacy segment is separated by the `!!!` delimiter. In order to extract the privacy policy segments, we used this delimiter and extracted all the segments from all HTML files. Then we parsed the consolidation files and matched each segment with its corresponding row, ensuring alignment between policy text and metadata.

The final preprocessed files for training, validation, and test sets include the columns presented in Table 3.7.

Table 3.7: Dataset Schema and Variable Descriptions for Privacy Policy Analysis

Column Name	**Description**
policy id	Unique identifier for each privacy policy document.
segment id	Identifier for each text segment (usually a paragraph or sentence).
third party sharing	Indicates whether the segment discusses third-party sharing or collection.
Computer_information	Refers to technical data like browser type, OS, screen resolution, etc.
Contact	Includes email address, phone number, or physical address.
Cookies and tracking elements	Refers to tracking technologies such as cookies, beacons, or pixels.
Demographic	Information such as age, gender, income, or education level.
Financial	Includes credit card numbers, banking information, or payment details.
Generic personal information	Broad personal information not falling into more specific categories.
Health	Any health-related data or medical history.
IP address and device IDs	IP addresses, MAC addresses, or device identifiers.
Location	Geographical location data, GPS coordinates, or city/state.
Other	Information that does not clearly fall under other predefined categories.
Personal identifier	unique identifiers, such as username, account ID, or national ID.
Survey_data	User-provided responses to surveys or forms.
Unspecified	Used when the type of personal information is not clearly identified.
User_Profile	Data about the user's preferences, interests, or profile settings.
User_online_activities	Tracks user behavior such as browsing history, clicks, or page views.
segment_text	The actual text content of the segment from the privacy policy.

3.3 Large Language Model Selection

To evaluate different LLMs for the analysis of privacy policies for third-party data sharing and collection, we selected three open-source models and two commercial ones. The list, along with their exact versions, is as follows.

- **DeepSeek-R1-Distill-Qwen-32B** [47]
- **Mixtral-8x7B-v0.1** [55]
- **Llama-3.1-8B** [56]
- **Grok 3 Beta** [53]
- **Gemini 2.5 Flash** [54]

The Table 3.8 summarizes five Large Language Models (LLMs) evaluated for third-party data collection: DeepSeek-R1-Distill-Qwen-32B, Mixtral, Llama, Grok, and Gemini, based on fine-tuning capabilities, context size, open-source status, and deployment requirements. It is worth mentioning that DeepSeek-R1-Distill-Qwen-32B, Mixtral, and Llama are hosted on Hugging Face. Gemini uses Google Cloud's Vertex AI. Grok is by xAI.

Table 3.8: Comparison of Large Language Models: Technical Specifications and Licensing

Model	**Fine-Tuning**	**Context Size**	**Open Source**	**Parameters**
DeepSeek-R1-Distill-Qwen-32B	Yes (LoRA, QLoRA)	128K tokens	Yes (MIT, with distillation restrictions)	32B
Mixtral-8x7B-v0.1	Yes (LoRA, QLoRA)	32K tokens	Yes (Apache 2.0)	12.9B active / 46.7B total
Llama-3.1-8B	Yes (LoRA, research use)	128K tokens	No (Meta license, non-commercial)	8B
Grok 3 Beta	No	8K tokens (standard)	No (proprietary, xAI)	Not disclosed (estimated 60B+)
Gemini 2.5 Flash	Yes (Google Cloud)	1M tokens	No (proprietary, Google)	Estimated 1.6T (Mixture of Experts, sparse activation)

3.4 Implementation Overview

This section describes the technical implementation of a batch inference pipeline for analyzing privacy policies using the OPP-115 and MAPP datasets. We have analyzed 15 different personal data types for our research analysis (e.g., financial, health, contact). The list of the data types that we have used for our experiments is mentioned in the Table: 3.1.

3.4.1 Inference pipeline

In our inference pipeline, we experimented with two main prompt designs. In our first prompt design, we simultaneously analyzed all 15 categories, and in the second design, we analyzed each information type separately. The prompts follow a standardized structure with task definitions, category descriptions, strict YES/NO criteria, a few-shot examples, policy text, and JSON output format, ensuring consistent responses [31]. For the model selection, we ran exploratory analysis on the models mentioned in this table 4.1, and after analyzing their performance and cost comparison, we picked a total of 5 models. 3 of these models are open source, and 2 are proprietary-based models. Their detailed description is as follows.

3.4.1.1 Open-Source Models

The 3 open source models we used in our researh are as follows. In this section, we have mentioned the deployment settings we have used to load these models

DeepSeek-R1-Distill-Qwen-32B This model was deployed on 2 × NVIDIA A100-80GB GPUs due to its larger computational requirements. We used vLLM for the inference pipeline. This model was also loaded in 16-bit precision. The detailed analysis of 16-bit precision vs 32-bit precision is mentioned in this table. 5.1. This model supports structured outputs, which means we can define the object in which we expect the model to respond.

Mixtral 8x7B The Mixtral model was deployed on 2 × NVIDIA A100 GPUs. In order to load this model for inference, we used the same components, vLLM with FP16 precision mode for efficiency. As an open-source sparse Mixture-of-Experts (MoE) model, Mixtral also supports structured outputs.

Llama 3.1-8B Llama was also deployed using 2 × A100 GPUs. This is small in parameter size as compared to the distilled DeepSeek model and Mixtral. That's why it required fewer computational resources as compared to the other two models. Like the 2 above, Llama3.1 also supports structured output format.

3.4.1.2 Commercial Models

Our two commercial models, including Gemini and Grok, are accessed via their respective REST APIs, allowing seamless integration with external services.

Gemini Model The Gemini model requires a cloud storage bucket for managing input and output data. It accepts files in JSONL format [57], which supports batch processing of prompts for enhanced throughput.

Grok Model Grok has published their inferance APIs. but the APIs does not support batch processing yet [53]. To overcome this limitation, we developed a custom service that leverages Grok's completion API [53] to process prompts individually and store the responses for further analysis.

3.4.1.3 Libraries used during inference pipeline

The following libraries facilitate key tasks in the inference pipeline:

- **vLLM** [58]: A high-performance library for serving and performing inference with large language models, optimizing memory usage and throughput for efficient processing on local hardware.
- **Pydantic** [59]: A data validation and parsing library used to define and enforce structured schemas for model inputs and outputs, ensuring consistent data formatting in JSON or similar formats.
- **Datasets** [60]: A Hugging Face library for efficiently loading, preprocessing, and managing large datasets, enabling streamlined data handling for model inference.
- **PyTorch** [61]: A deep learning framework that provides the backend for model inference, supporting tensor operations and GPU acceleration for efficient computation.
- **Pandas** [62]: A data manipulation library used for preprocessing input data, grouping samples, and managing batch processing, as well as storing inference results with associated metadata.

- **Regex (re)** [63]: A Python module for regular expression operations, used to parse and validate model outputs, particularly for extracting structured data from text responses.
- **OpenAI (xAI API Client)** [64]: A client library for interacting with xAI's REST API, facilitating prompt submission and response retrieval for proprietary models like Grok.
- **Scikit-learn** [65]: A machine learning library used to compute evaluation metrics such as accuracy, precision, recall, and F1 score, enabling performance analysis of model outputs.

3.4.2 Evaluation Metrics

The outputs of the models were evaluated using a confusion matrix, which categorizes predictions into true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [66]. These components were used to compute the following metrics:

- **Precision**: The ratio of correctly identified positive instances to all instances predicted as positive, defined as
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$
- **Recall**: The ratio of correctly identified positive instances to all actual positive instances, defined as
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$
- **Accuracy**: The proportion of correct predictions, defined as
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$
- **F1 Score**: The harmonic mean of precision and recall, defined as
$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The F1 score was selected as the primary metric due to the class imbalance in privacy policy datasets, such as OPP-115 [12] and MAPP [11], where certain data practices (e.g., third-party sharing) are less frequent. The F1 score balances precision and recall, ensuring robust evaluation in imbalanced settings, which is critical for accurately identifying privacy practices.[12, 66].

4

MAPP Experimentation

4.1 Experimental Setup

We started our experimentation by focusing on the MAPP dataset with the aim of analysing how well the selected large language models behave when they encounter a privacy policy. And to evaluate if they accurately predict the collection or sharing of data with third parties. We passed the data to the LLMs through a pre-processing pipeline that was used to clean the data and format it in a way that would make the language model's performance better. We evaluated the following models for the inference pipeline (Table 4.1) using the MAPP dataset for third-party information detection.

Table 4.1: List of Evaluated Models and Inference Setup

Model	**GPU Used**
Claude (Anthropic)	Proprietary (Cloud API)
Gemini [54]	Proprietary (Cloud API)
Grok [53]	Proprietary (Cloud API)
DeepSeek-R1-Distill-Qwen-32B [47]	2× A100-80GB
Llama 2 70B [67]	2× A100-80GB
Mixtral [55]	A100
Phi-4 [68]	A100

4.1.1 Inference pipeline

For inference, we used Hugging Face's Transformers library. We used AutoTokenizer and AutoModelForCausalLM to load the model and tokenizer, then employed the built-in text-generation pipeline for generating responses on the MAPP test data. After the pipeline completed inference, we calculated the confusion matrix by calculating true positives, false positives, true negatives, and false negatives. This setup provided a foundation to evaluate models behavior on the MAPP dataset.

4.1.2 Exploratory Inference on the MAPP Dataset

Using the above mentioned setup, we conducted preliminary inference on the MAPP dataset to gain initial insights into model behavior. The main purpose of this experiment was to assess the feasibility of using MAPP for our main experiments, particularly for third-party data practices.

We experimented with two prompt styles: one requesting all data types at once and another querying each category separately. The following prompts shown below **??** have the structure of our prompt that we used for the single attribute inference.

Privacy Policy Analysis Prompt

You are a helpful assistant trained to analyze privacy policies. Always respond with YES or NO as instructed.
The following content between the double quotation marks is a privacy policy.
"Privacy policy text"
Task:
Determine whether the privacy policy explicitly affirms that any of the following personal data types are collected by or shared with third parties. Only return "YES" if the policy clearly states or directly implies that the specific data type is collected by or shared with a third-party. If the data type is not mentioned or the policy is unclear, return "NO."
Data Types:
"Financial": Financial information, such as credit/debit card data, other payment information, credit scores, etc.
Output Format Instruction:
Please format your answer as follows:
Data: Answer
where Data is the data type above, and Answer must be only YES or NO. Strictly follow the output format. Do not add anything extra in the response.

Figure 4.1: Prompt Template for Privacy Policy Analysis Task for one infomation type

The prompt that we used to run the inference experiment for all the information types at once is as follows in the given table 4.2

Privacy Policy Analysis Prompt

You are a helpful assistant trained to analyze privacy policies. Always respond with YES or NO as instructed.
The following content between the double quotation marks is a privacy policy.
"Affiliates Press Contact Support Terms Privacy Site Notice"
Task:
Determine whether the privacy policy explicitly affirms that any of the following personal data types are collected by or shared with third parties. Only return "YES" if the policy clearly states or directly implies that the specific data type is collected by or shared with a third-party. If the data type is not mentioned or the policy is unclear, return "NO."
Data Types:

- "Financial": Financial information, such as credit/debit card data, other payment information, credit scores, etc.
- "Health_genetic_or_biometric_data": Information about a person's health, genome, or biometric markers.
- "Contact_information": Contact information, such as name, email address, phone number, street address, etc.
- "Location": Geo-location information (e.g., user's current location) regardless of granularity, i.e., could be exact location, ZIP code, city level.
- "Demographic_data": Demographic information, e.g., gender, sexual orientation, race, ethnicity, age, occupation, education, etc.
- "Personal_identifier": Identifiers that uniquely identify a person, e.g., SSN, ID card number, driver's license number, etc.
- "User_online_activities": The user's online activities on the first-party websites/apps or other (third-party) websites/apps, e.g., user profiles, pages visited, time spent on pages, general user behavior online, etc.
- "Social_media_data": User profile and data from a social media website/app or other third-party service to which the user gave the First_Party access, e.g., by connecting with Facebook, Twitter, or other services. Exchanged data may include user profile, photos, comments, friends, etc.
- "IP_address_and_device_IDs": Permanent (e.g., device IDs, MAC address) or temporary (e.g., IP address) identifiers needed to establish a connection for the current browsing session.
- "Cookies_and_tracking_elements": Identifiers locally stored on the user's device by the company/organization or third parties, including cookies, beacons, or similar that are commonly used to identify users uniquely but are not essential to establish a connection with the user's device or to provide a service.
- "Computer_information": The type of operating system (OS) or web browser that the user uses, or similar computer or device information.
- "Generic_personal_information": No specific type of information is mentioned, but the policy talks about 'personal information' or 'personally identifiable information' in general.
- "Political_religious_or_philosophical_belief": Any data that describes political, religious, or philosophical beliefs of individuals.
- "Other": A specific type of information not covered by other values for this attribute.
- "Unspecified": The type of information is not explicitly stated or unclear (e.g., refers to 'information' very generically).

Output Format Instruction:
Please format your answer as follows:
Data: Answer
where Data is the data type above, and Answer must be only YES or NO. Strictly follow the output format. Do not add anything extra in the response.

Figure 4.2: Comprehensive Privacy Policy Analysis Prompt with Multiple Data Types

4.2 Results on the MAPP Dataset

We evaluated MAPP dataset using the models mentioned in Table 4.1. We analyzed these models to predict whether the privacy policy mentions that it shares the user's personal information type with a third party or not. The results obtained from these models are explained in the subsections below follows. We focused only on the F1 score as the nature of our dataset is unbalanced.

We tested each model at temperatures 0 and 1. This was to test at which extreme the model performs better. In this phase, we were following the prompt design mentioned in [69]. All the experiments with the MAPP dataset were based on these prompts with some level of tweaking. But the base structure was the same.

4.2.1 Claude (Anthropic) on MAPP

Based on the results of experiments Claude on the MAPP dataset (Table 4.2), the F1 scores for Claude (Anthropic) are generally low across all temperature settings, indicating limited effectiveness in detecting third-party information in the MAPP dataset. However, the Haiku variant, especially when combined with a different prompt strategy, shows improvements, suggesting that both model variant and prompt design can significantly impact performance.

Table 4.2: F1 Score Comparison by Temperature: Claude (Anthropic) and Gemini 1.5 Flash on MAPP Dataset

Model	Temperature	F1 Score
Claude (Anthropic)	0	0.0622
Claude (Anthropic)	0.2	0.0542
Claude (Anthropic)	1	0.0485
Claude (Anthropic, Haiku)	1	0.2576
Claude (Anthropic, Haiku, different prompt strategy)	1	0.4095
Gemini 1.5 Flash	1	0.3243

4.2.2 Grok

Grok models were evaluated on third-party information types with various prompt engineering and temperature settings (Table 4.3). Notably, adding a clear third-party definition to the prompt significantly improved F1 scores. The Grok 3 Beta model with definitions achieves the best results among the Grok variants. This highlights the importance of prompt clarity and the model's ability to leverage explicit instructions for better third-party information detection.

Table 4.3: F1 Scores for Grok on Third-Party Information Types

Setting	Temperature	Definitions	F1 Score
Grok 3 Mini	0	NO	0.1506
Grok 3 Mini	1	NO	0.1601
Grok 3 Mini	0	YES	0.2687
Grok 3 Mini	1	YES	0.2709
Grok 3 Beta	1	YES	0.2904

Mixtral and Phi-4 Results (Third-Party)

Mixtral and Phi-4 were evaluated on third-party information types at a temperature of 0 (Table 4.4. Both models struggled with class imbalance, with Mixtral achieving a very low F1 score and Phi-4 performing slightly better.

Table 4.4: F1 Scores for Mixtral and Phi-4 (Third-Party, Temp 0)

Model	F1 Score
Mixtral (Temp 0)	0.037
Phi-4 (Temp 0)	0.118

Both Mixtral and Phi-4 struggle with the third-party detection task, as indicated by their low F1 scores. Phi-4 performs slightly better than Mixtral, but overall the perfomance of the model using prompt 4.2 was bad.

4.2.3 DeepSeek-R1-Distill-Qwen-32B (Temperature 1 and 0)

DeepSeek-R1-Distill-Qwen-32B was evaluated on third-party information types, both as a base model and with fine-tuning. Based on the results in Table 4.5, the model struggled with class imbalance, but prompt engineering and fine-tuning improved F1 scores in some settings.

DeepSeek-R1-Distill-Qwen-32B's base model performs poorly when evaluated on all attributes together, with F1 scores close to zero. However, when evaluated on single attributes, the model achieves much higher F1 scores for certain categories (e.g., Demographic, Financial, Personal Identifier), indicating that DeepSeek-R1-Distill-Qwen-32B can be effective for specific information types when the class imbalance is less severe or the attribute is more clearly defined. Nevertheless, performance remains inconsistent across categories, and the model struggles with others, especially under class imbalance.

Table 4.5: F1 Scores for DeepSeek on Third-Party Information Types

Setting	F1 Score
Single Attribute, Temp 0	
Contact Information	0.4211
Health	0.0000
Social Media	0.0800
Demographic	0.8000
Location	0.2500
User Online Activities	0.4091
Financial	0.5455
Personal Identifier	0.6667
Generic Personal Information	0.5349
Political and Religious	N/A
Single Attribute, Temp 1	
Contact Information	0.0588
Location	0.0000
Cookies	0.1429
Personal Identifier	0.0263
Demographic Data	0.0294
Health (Generic)	0.0000
Social Media Data	0.0455
User Online Activities	0.1345
Generic Personal Information	0.1720
Political or Religious Information	0.0000

4.2.4 Gemini on MAPP

For Gemini, we tested both prompt styles: querying all personal information attributes together and querying each attribute separately. The results are reported in Table 4.6.

Table 4.6: F1 Score for Gemini 1.5 Flash on MAPP Dataset (All Attributes Combined)

Model	F1 Score
Gemini 1.5 Flash	32.43%

Gemini 1.5 Flash achieves a moderate F1 score when evaluated on all attributes combined, outperforming several other models. 4.6. When we tested the model with single attribute inference approach, Gemini demonstrated strong performance for some categories at temperature = 0 (e.g., Demographic), but F1 scores are zero for health. 4.7This suggest that if done correctly, with prompt engineering, the results from Gemini can be improved.

Table 4.7: F1 Score for Gemini Single Attribute Results

Information Type	**F1 Score**
Computer Information	0.2000
Location	0.2000
Contact Info	0.2759
Other	0.1333
Cookies and Tracking Elements	0.3448
Personal Identifier	0.0000
Demographic Data	1.0000
Philosophical Data	0.0000
Financial	0.1739
Social Media Data	0.0588
Generic Personal Info	0.3243
Unspecified	0.2535
Biometric	0.0000
User Online Activities	0.3077
IP Address and Device IDs	0.3636

4.2.5 Llama 2 70B on MAPP

Llama 2 70B, does not support structured outputs. When prompted with an unstructured output format, the model often produced irrelevant, verbose, or off-format responses, making automated evaluation challenging. This issue was consistent across all prompt styles (full, shortened, and few-shot). For example, the model sometimes outputs long, rambling, or repetitive text, or simply repeats the prompt or instruction instead of providing answers in the required format. These results indicate that Llama 2 70B behaves differently with respect to different information types, as shown in the table 4.8.

Table 4.8: F1 Scores for Llama 2 70B by Information Type

Category	**F1 Score**
IP Address and Device IDs	0.57
Contact Information	0.26
Generic Personal Information	0.25
Personal Identifier	0.10
Cookies and Tracking Elements	0.42
Computer Information	0.49
User Online Activities	0.43
Social Media Data	0.20
Financial	0.44
Location	0.35
Health, Genetic, or Biometric Data	0.38
Demographic Data	0.40

Llama 2 70B shows highly variable performance across information types, with some categories achieving moderate F1 scores (e.g., IP Address and Device IDs, Cookies and Tracking Elements), while others remain low. The overall F1 score is modest, and the model's tendency to produce unstructured or verbose outputs further limits its utility for automated evaluation in this context.

4.2.6 Observations from Visual Comparison

Considering All Attributes Together

- **Overall Low Performance:** When models are tasked with predicting all attributes at once, most achieve low F1 scores (Figure 4.3). This suggests that the multi-label, multi-class nature of the task is challenging for current LLMs, especially in the presence of class imbalance.
- **Model Differences:** Based on Figure 4.3, Gemini stands out with a notably higher F1 score compared to other models, indicating better generalization or adaptation to the complex prompt. Claude (Anthropic) and Grok show moderate performance, while DeepSeek-R1-Distill-Qwen-32B, Mixtral, and Phi-4 struggle, possibly due to their architectures or training data limitations.
- **Impact of Prompt Complexity:** The drop in performance for most models in this setting (considering all attributes together) highlights the importance of prompt design and the limitations of LLMs in handling complex, multi-faceted queries without additional guidance or structure.

Single Attribute/Category

- **Improved and Varied Performance:** Based on the Figure 4.4, and in comparison with Figure 4.3 F1 scores generally improve when the task is simplified to single-attribute prompts, especially for certain categories. This demonstrates that models are better at focused, binary classification tasks than at handling multiple labels simultaneously.
- **Category Sensitivity:** Some categories (e.g., Cookies and tracking elements, Ip address and device info) show much higher F1 scores (Figure 4.4), suggesting that these are either more clearly defined in the data or more easily recognized by the models. Other categories (e.g., Social Media, Financial) remain challenging, possibly due to ambiguity or fewer positive examples.
- **Model Strengths:** DeepSeek-R1-Distill-Qwen-32B and Gemini show strong performance in several categories (Figure 4.4),indicating their potential for targeted information extraction as compared to the joint inferance for all information type at once in a single prompt. However, performance is inconsistent across categories, highlighting the need for further fine-tuning or data balancing.

General Insights

- **Prompt Engineering Matters:** The clear difference in performance between the two prompt strategies explains the importance of prompt engineering. Clear, focused prompts enable models to perform better as they help them to understand the problem in a better way.

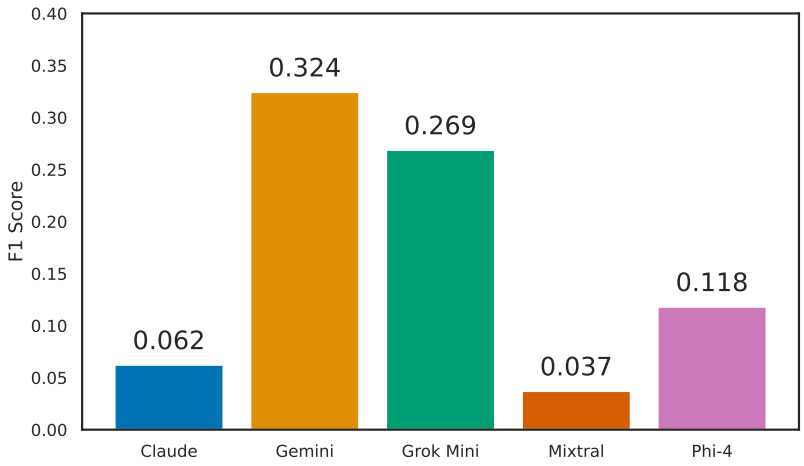

Figure 4.3: F1 comparison of different models for all information types for MAPP dataset - temperature = 0

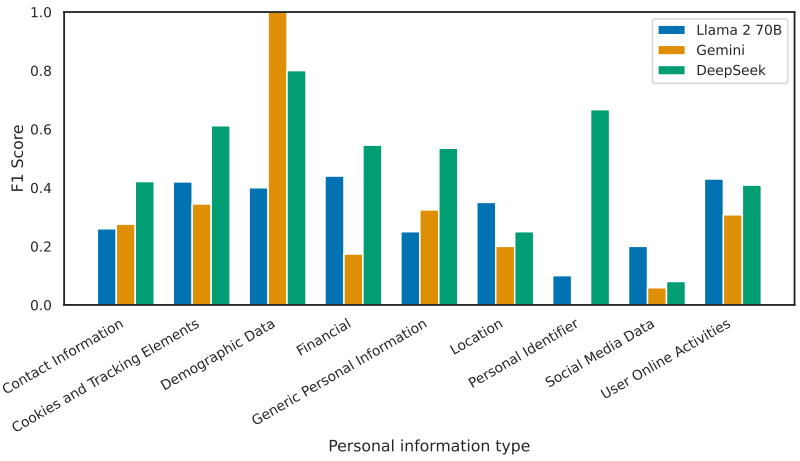

Figure 4.4: F1 comparison of different models for each information types separately for MAPP dataset - temperature = 0

- **Class Imbalance Remains a Challenge:** Even with single-attribute prompts, categories with fewer positive examples still yield low F1 scores, indicating that class imbalance is a persistent issue.
- **No Universal Winner:** No single model consistently outperforms others across all settings and categories.

4.2.7 Motivation for Choosing the OPP_115 Dataset

Given the limitations of the MAPP dataset, most notably the imbalance in key label categories, we shifted our focus to the OPP_115 dataset for subsequent experiments. OPP_115 offers a more balanced label distribution, particularly across first-party and third-party data practices, enabling more robust and meaningful model evaluation.

This decision was taken as we needed a dataset that supports both evaluation and training characteristics, that is, more balanced labels for all classes. As demonstrated in later sections, OPP_115 facilitated a more consistent experimental pipeline and more reliable results. Although it is also imbalanced, it still improved the overall inference and fine-tuning pipeline.

5

OPP_115 Experimentation

5.1 Introduction

In this section, we evaluate the performance of large language models using the OPP_115 dataset. The experiment focused on analyzing the ability of the model to correctly answer whether the personal information type is being shared with a third-party or not using the OPP_115 dataset. The experiments are more elaborate and extensive as compared to the MAPP dataset analysis, as we learned some lessons and shortcomings from our exploratory research and implemented them in the best possible way for the large language models using OPP_115. In the analysis, we performed an ablation study. By adding or removing different prompt components, we analyzed the capacity of different LLMs to answer privacy prompts related to the third party data sharing.

As mentioned earlier, for performance evaluation, we used confusion matrices to calculate accuracy, precision, recall, and F1-score. Among these, our primary focus was on the F1 score, as it provides a balanced measure of both precision and recall. It is especially important in scenarios with class imbalance, which is present in our privacy policy corpus. High precision ensures that predicted attributes are correct, while high recall ensures that the most relevant attributes are captured. The F1 score, as their harmonic mean, effectively captures the trade-off between them. We have reported these metrics individually for each attribute to assess model performance.

5.1.1 Experimental Setup

Our experimental setup consisted of the following components.

- **Inference Pipeline with vLLM:** For our OPP_115 analysis, we used vLLM for our open source models. vLLM is also an open-source library. The main focus of vLLM library is to optimize the deployment and inference of large language models (LLMs). It helps in the performance of inference pipeline by minimizing memory overhead and utilizing continuous batching strategies. This results in significantly higher throughput and efficiency compared to traditional inference approaches.
- **Local Model Deployment:** Due to the lack of internet connectivity on our GPU

machines, we pre-downloaded the model and loaded it locally prior to running the inference. This ensured a smooth and uninterrupted inference workflow.

- **Half-Precision Model Loading (FP16):** When we load the model for inference, we have to mention the precision with which the model should be loaded. For our experimentation, we loaded the models in half precision (16-bit floating point) format. We took this decision after comparatively experimenting with both precisions. The evaluation showed negligible differences in accuracy metrics between the two, while half precision substantially reduced the memory footprint, and it made the inference pipeline faster as well. Notably, loading the model in FP16 required only half the GPU resources compared to FP32, allowing us to optimize hardware usage effectively.

  We validated this choice by running an internal benchmark comparing both precisions. The results showed minimal performance difference: the 32-bit model achieved an F1 score of 0.6122, while the 16-bit model scored 0.6118. Given the faster inference time and smaller memory footprint, we adopted 16-bit precision for all experiments. The results for our benchmark are mentioned in the table below 5.1. We only tested it with 2 models and generalize it for the Mixtral as well.

Table 5.1: Comparison of model performance, efficiency, and resource usage at different precisions

Model	**Precision**	**F1 Score**	**GPU Usage**
DeepSeek_R1_Distill_Qwen_32B	32-bit	0.6122	Higher
DeepSeek_R1_Distill_Qwen_32B	16-bit	0.6118	Lower
Llama 3 8B	32-bit	0.5000	Higher
Llama 3 8B	16-bit	0.5143	Lower

- **Structure outputs** We used the structured output for our output configuration. This helped us in the most profitable way, as the response before implementing structured output was uncertain. This structured output increased the response quality to a high level. For instance, the response before and after the implementation of structured output is as follows.

- **Ablation study** To understand the impact of prompt design, we conducted the same ablation study. We systematically added or removed various sections from the prompt and calculated the F1 score using the test dataset.

  The ablation configuration was as follows.

  - **Effect of temperature variation**
  - **Effect of top p variation**
  - **Effect of different numbers of few-shot prompts**
  - **Effect of inclusion and exclusion of third-party definition**
  - **Effect of prompt with all the attributes together**

5.1.2 Performance Analysis of DeepSeek-R1-Distill-Qwen-32B for Privacy Policy Classification

This section evaluates the performance of the DeepSeek-R1-Distill-Qwen-32B model using OPP_115 dataset. In this analysis, we have examined the impact of temperature, a few-shot prompting, the presence and absence of third-party (TP) definitions, and the presence or absence of instructions on the performance of DeepSeek-R1-Distill-Qwen-32B

5.1.2.1 Prompt Selection

Prompt engineering has a significant effect when we test LLM's performance in our privacy policy analysis case. We tested various prompt styles, and the best that worked for us is mentioned here. 5.1.2.1

Base Prompt

Task: Analyze this privacy policy for third-party computer information disclosures. Respond ONLY with "YES" or "NO".

Definitions

Computer_information:
"The type of operating system (OS) or web browser that the user uses, or similar computer or device information."
third-party:
"Natural or legal person, public authority, agency, or body other than the data subject, controller, processor and persons who, under the direct authority of the controller or processor, are authorised to."

Analysis Criteria

- Only mark as YES if the policy clearly states that computer information is shared with third parties
- If the policy is unclear or doesn't explicitly mention sharing, mark as NO
- Be very strict in your analysis | require clear evidence

Examples

[YES]:
"We share your operating system and browser type with Google for analytics purposes"
"Our advertising partners receive device information, such as browser version, for targeted ads"
[NO]:
"We collect device information for site functionality"
"Third parties may access data"
"Operating system data is used to improve our services"

Policy: "[text]"
Answer: Computer_information: [YES/NO]

5.1.2.2 Temperature Variations

The temperature parameter influences the randomness or creativity of large language models (LLMs) during inference. To understand its effect in the context of privacy policy analysis, we evaluated three different temperature settings across all attributes of the type of personal information.

The table shows that the model performs the best at temperature 0 for most of the personal information types. However, in some cases, like Financial and others, the model performed better at a temperature of 0.2. In these cases when the context is not very distinctive,

the model benefited from its creativity. From this we can say that in order to identify privacy policies, we need the model to be in its deterministic form. High temperature value introduces variability in the model response, which tends to degrade the performance for most of the information types. This comes with an exception of few information types, where the model in creative mode improves the F1 score.
In summary, for this domain-specific task, if we load the model in its most conservative form, it returns the most reliable results in most of the case. This means reduced creativity leads to more accurate and consistent performance on our specific domain. However, there is no fit for all. It's better to test the information types at various temperatures and try to create a balance between recall and precision as clearly suggested by the results in the table. 5.2
For detailed visual analysis of temperature effects across all attributes, see Figure B.1 in Appendix B.

Table 5.2: F1 scores for all attributes across temperature settings (0, 0.1, 0.2). The highest score in each row is bolded.

Attribute	**Temp 0**	**Temp 0.1**	**Temp 0.2**
Contact	**0.2647**	0.2571	0.2329
Cookies and Tracking	**0.6118**	0.6000	0.5977
Demographic	**0.3448**	0.2581	0.2162
Financial	0.4444	0.4167	**0.4651**
Computer Information	0.2264	**0.2308**	0.2069
Generic Personal	**0.4495**	0.4482	0.4078
Health	**0.2500**	0.1250	0.1111
IP Address and Device IDs	**0.5714**	0.5455	0.5455
Location	**0.6154**	0.5714	0.5333
Other	0.1798	0.1978	**0.2391**
Personal Identifier	0.0190	**0.0206**	0.0204
Survey Data	**0.0606**	0.0571	0.0571
Unspecified	**0.4103**	0.4051	0.3975
User Online Activities	**0.4754**	**0.4754**	0.4567

5.1.2.3 Few-Shot Prompting

A few shots prompting involves providing examples with their expected output. We experimented with 0, 1, 2, and 3-shot prompting techniques. Few-shot prompting is done to guide the model's output. The results show mixed effects depending on the label. For Cookies_and_tracking_elements, performance improves with more shots, reaching an F1 score of 0.6118 with 3 shots, compared to 0.4956 with 0 shots. In contrast, Computer_information performs better with 0 shots (F1 score 0.1839) than with 1 or 2 shots (0.1791 and 0.2034, respectively), suggesting that examples may introduce noise for certain categories.
The health information section was the hardest for deepseek analysis. Even when we tried to help the model by giving it more targeted few-shot examples that were very similar to the test cases, it still didn't predict them correclty. This suggests that the model may not be able to generalize well in the case of health.

For comprehensive analysis of few-shot prompting effects across all attributes, see Figure B.2 in Appendix B.

Table 5.3: F1 scores for all attributes across few-shot prompt counts (0, 1, 2, 3 shots). Missing values indicate unavailable data. Highest values in each row are bolded.

Attribute	0-shot	1-shot	2-shot	3-shot
Computer Information	0.1839	0.1791	0.2034	**0.2308**
Contact	0.2647	0.2647	0.2647	**0.3238**
Cookies and Tracking	0.4956	0.5455	0.5217	**0.6118**
Demographic	0.2182	0.2857	0.2857	**0.3448**
Financial	0.2973	0.4151	**0.5116**	0.4444
Generic Personal	0.4580	**0.4663**	0.4459	0.4495
Health	0.1111	**0.2500**	**0.2500**	**0.2500**
IP Address and Device IDs	0.2571	**0.5714**	**0.5714**	**0.5714**
Location	0.3200	0.5000	**0.6154**	**0.6154**
Personal Identifier	**0.0309**	0.0198	0.0179	0.0206
Survey Data	**0.0741**	0.0588	0.0488	0.0606
Unspecified	0.2778	0.4000	**0.4103**	0.3333
User Online Activities	0.4328	0.4234	0.4328	**0.4754**

5.1.2.4 Third-Party Definitions

DeepSeek-R1-Distill-Qwen-32B is a Chain-of-Thought (CoT) based model, which generally performs better when provided with clear reasoning steps or well-written instructions. In our previous experiments, we used a base prompt that included the definition of the "Third-Party". To understand the impact of this additional information, we tested how the model would perform if we removed the "Third-Party" definition from the prompt.

We found out that removing this part actually improved the F1 scores across all attributes. One possible explanation is that the "Third-Party" definition may have caused confusion or distraction for the model, making it harder for it to focus on the main task. Without that extra context, and allowing the model to rely on its pre-trained knowledge seemed to have a better effect on the overall performance of the model.

This suggests that sometimes giving too much information in the prompt especially if it is not directly useful, can hurt the model's performance. For models like DeepSeek-R1-Distill-Qwen-32B that rely on CoT reasoning, keeping the prompt simple and focused might lead to better results.

For detailed analysis of third-party definition effects, see Figure B.3 in Appendix B.

Table 5.4: F1 scores for all attributes across third-party definition settings. Highest values in each row are bolded.

Attribute	Without TP	With TP
Computer Information	**0.5000**	0.2308
Contact	**0.2917**	0.2647
Cookies and Tracking	**0.6176**	0.6118
Demographic	**0.4000**	0.3448
Financial	**0.5263**	0.5116
Generic Personal	0.4468	**0.4663**
Health	**0.4000**	0.2500
IP Address and Device IDs	0.5455	**0.5714**
Location	**0.6667**	0.6154
Personal Identifier	**0.0440**	0.0190
Survey Data	**0.1333**	0.0741
User Online Activities	**0.5047**	0.4754

5.1.2.5 Effect of Instructions

The removal of instruction has a negative effect on the performance of the model. as the F1 score drops by 50 percent for cookies and tracking elements. So, we generalize it for all other information types as well.

Table 5.5: F1 scores for attributes with and without instructions. Highest values in each row are bolded.

Attribute	With Instructions	Without Instructions
Cookies and Tracking	**0.6118**	0.3587
Health	**0.2500**	0.2222

5.1.3 Performance Analysis of Mixtral for Privacy Policy Classification

In this section, we evaluate the performance of the Mixtral 8x7B model model using OPP_115 dataset. In this analysis, we have examined the impact of temperature, a few-shot prompting, the presence and absence of information type definitions on the performance of Mixtral8x7B.

5.1.3.1 Prompt Engineering

After testing multiple prompts with DeepSeekR1DistillQwen32B, the plan was to use the same prompt and test Mixtral and all other models in our list. To our surprise, the same prompt that worked good for DeepSeek-R1-Distill-Qwen-32B performed really badly for Mixtral. The results that we obtained using the same prompt as Mixtral are mentioned in the following table 5.6

Table 5.6: F1 scores for attributes when Mixtral was tested using DeepSeek-R1-Distill-Qwen-32B prompt.

Attribute	**F1 Score (DeepSeek Prompt)**
Computer Information	0.1194
Demographic	0.0253
Health	0.0000
IP Address and Device IDs	0.0318
Location	0.0223
Other	0.0680
Survey Data	0.0400

So, in order to find the right type of prompt that will work for mixtral, we started doing prompt engineering again. We tried out many prompts from the simplest to the complex, and found out that the following prompt 5.1.3.1 worked better for Mixtral.

Instruction-Style Prompt

```
<s>[INST]
You are a privacy policy analysis assistant.  Respond ONLY
with "YES" or "NO" to indicate whether the
privacy policy mentions third-party [Personal_Information_Type]
sharing.
Example 1:
Privacy Policy:  "We may use third-party cookies to track
your behavior across websites."
Answer:  [[ [Personal_Information_Type]:  YES ]]
Example 2:
Privacy Policy:  "We only use internal analytics tools."
Answer:  [[ [Personal_Information_Type]:  NO ]]
Now analyze the following:
Privacy Policy:  [text]
[/INST]
```

Then we kept this prompt as a base prompt, and performed an ablation study as we did for DeepSeek-R1-Distill-Qwen-32B. We added and removed components from the prompt to see how each component of the prompt effects the model's output. As mentioned earlier, we tested mixtral with the same prompt as of deepseek and calculated the F1 score for it as well. Following are all the experiments that we performed on mixtral along with their F1 score.

5.1.3.2 Temperature Variations

Temperature settings influence output determinism, with lower values (e.g., 0) producing more predictable results. Table 5.2 presents F1 scores for all attributes across different temperature settings (0, 0.1, 0.2). *Cookies and Tracking Elements* achieves the highest F1 score (0.6000 at temperature 0), which again shows strong performance with deterministic outputs. *Financial* shows a decline from 0.3529 (temp 0) to 0.1008 (temp 0.2), driven by increased false positives and reduced precision. Less distinctive categories like *Personal Identifier*, *Survey Data*, and *User Profile* consistently yield F1 scores of 0.0000, indicating challenges when the information type is vague or does not have clear definition. *Health* and *Location* exhibit moderate performance (0.3636 and 0.2222 at temp 0).
For detailed visual analysis of temperature effects across all attributes, see Figure B.4 in Appendix B.

Table 5.7: F1 scores for all attributes across temperature settings (0, 0.1, 0.2) for Mixtral. All values are derived from confusion matrices and performance metrics. Highest values in each row are bolded.

Attribute	**Temp 0**	**Temp 0.1**	**Temp 0.2**
Computer Information	**0.1818**	0.0870	0.0745
Contact	**0.3077**	0.1410	0.1172
Cookies and Tracking	**0.6000**	0.5238	0.4348
Demographic	**0.3750**	0.2553	0.1064
Financial	**0.3529**	0.1818	0.1008
Generic Personal	0.0227	**0.2222**	0.1872
Health	0.3636	0.2059	**0.4571**
IP Address and Device IDs	0.0000	0.1379	**0.1923**
Location	**0.2222**	0.0789	0.0762
Other	0.1185	**0.1357**	0.1053
Personal Identifier	0.0000	**0.0208**	0.0195
Survey Data	0.0000	0.0000	**0.0093**
Unspecified	**0.3681**	0.2910	0.2508
User Online Activities	0.2566	**0.3881**	0.2619
User Profile	0.0000	**0.0525**	0.0480

5.1.3.3 Few-Shot Prompting

We experimented with Mixtral by providing 0, 1, 2, 3-shot examples in the prompt. Table 5.8 shows F1 scores for all attributes across 0, 1, 2, and 3 shots. *Cookies and Tracking Elements* peaks at 0.6000 with 1 shot but the F1 score decreases to 0.3846 with 3 shots, suggesting that additional knowledge in the prompts may introduce noise for the model. *Health* performs best with 0 shots (0.5714), dropping to 0.3265 with 3 shots. *Financial* achieves 0.4167 with 0 shots but falls to 0.0000 with 2 or 3 shots, indicating sensitivity to prompt count. *Generic personal information* remains weak across all settings (0.1439 at 0 shots, 0.0000 at 2 and 3 shots), and sparse categories like *personal identifier*, *survey data*, and *user profile* consistently score 0.0000. This means adding more shots in the prompt adds noise for the model. A one-shot prompt is the best configuration in our scenario for the model.

Table 5.8: F1 scores for all attributes across few-shot prompt counts (0, 1, 2, 3 shots) using Mixtral. Missing values indicate unavailable data. Highest values in each row are bolded.

Attribute	**0-shot**	**1-shot**	**2-shot**	**3-shot**
Computer Information	0.1071	**0.1818**	**0.1818**	–
Contact	0.1711	**0.3077**	0.2857	–
Cookies and Tracking	0.5867	**0.6000**	0.5098	0.3846
Demographic	0.2703	**0.3750**	0.1429	0.0714
Financial	**0.4167**	0.3529	0.0000	0.0000
Generic Personal	**0.1439**	0.0227	0.0000	0.0000
Health	**0.5714**	0.3636	0.3636	0.3265
IP Address and Device IDs	**0.2500**	–	0.1250	–
Location	0.0727	**0.2222**	0.1538	–
Other	0.1393	0.1185	**0.1571**	–
Personal Identifier	**0.0150**	0.0000	0.0000	–
Survey Data	**0.0000**	**0.0000**	**0.0000**	–
Unspecified	0.3092	**0.3681**	0.2896	–
User Online Activities	**0.2844**	0.2566	0.2833	–
User Profile	**0.0000**	**0.0000**	**0.0000**	–

5.1.3.4 Presence of contextual definitions

We tested mixtral by adding or removing contextual definitions in the prompt. For example, we provided the information type definition in the prompt to help model understand the promptl. Table 5.9 presents F1 scores for all attributes with and without definitions. *Cookies and Tracking Elements* improves with definitions (0.6667 vs. 0.6000 without), and *financial* reaches 0.5455 with definitions compared to 0.4167 without. *Demographic* also benefits (0.4167 vs. 0.3750). However, *Generic Personal Information* performs worse with definitions (0.0417 vs. 0.1439), and *Health* drops from 0.5714 (without) to 0.3750 (with). Sparse categories like *Personal Identifier*, *Survey Data*, and *User Profile* show F1 scores of 0.0000 in most settings, indicating limited effectiveness of definitions for these type of informatoin types.

For detailed analysis of definition effects, see Figure B.6 in Appendix B.

Table 5.9: F1 scores for all attributes with and without definitions. Highest values in each row are bolded.

Attribute	With Definitions	Without Definitions
Computer Information	**0.2593**	0.1818
Contact	0.2069	**0.3077**
Cookies and Tracking	**0.6667**	0.6000
Demographic	**0.4167**	0.3750
Financial	**0.5455**	0.4167
Generic Personal	0.0417	**0.1439**
Health	0.3750	**0.5714**
IP Address and Device IDs	0.1538	**0.2500**
Location	**0.3000**	0.2222
Other	0.1241	**0.1571**
Personal Identifier	0.0000	**0.0150**
Survey Data	**0.0000**	**0.0000**
Unspecified	0.2000	**0.3681**
User Online Activities	**0.3497**	0.2844
User Profile	**0.0000**	**0.0000**

5.1.4 Performance Evaluation of Grok for Privacy Policy Classification

In this section, we evaluates the performance of Grok3-beta model model using OPP_115 dataset. In this analysis, we have examined the impact of few-shot prompting at temperature 0 for Grok3-beta. For Grok, we restricted our self to just one ablation study as it is a paid service. So, we kept our focus on one type of experiment only.

5.1.4.1 Prompt engineering

As each model expects a different foramt or strucutre of the prompt, so we did some prompt engineering experiments for Grok to check at which prompt, the model is at its peak performance. After several experiments with prompt design, we choose the following prompt for the rest of our experiments.

Grok Prompt Example

System Instruction:

```
Task: You are a helpful assistant, expert in analyzing this
privacy policy for third-party cookies and tracking elements
disclosures. Respond ONLY with "YES" or "NO".
Analysis criteria:
1. cookies and tracking elements are defined as "Cookies,
web beacons, pixel tags, or similar tracking technologies
used to collect information about your activity." 2. Only
mark as YES if the policy clearly states that cookies or
tracking elements are shared with third parties 3. If the
policy is unclear or doesn't explicitly mention sharing, mark
as NO. 4. Be very strict in your analysis - require clear
evidence 5. If the policy says "May be", treat it as YES
```

```
User Input:
Policy: [policy text]
```

5.1.4.2 Effect of few shot prompts

The F1 score analysis across few-shot learning configurations with the Grok model reveals varying performance trends. Health achieves the highest F1 score of 0.8889 at 1-shot, indicating strong classification performance, while Financial also performs well with a peak of 0.6667 at both 1-shot and 3-shot. Cookies and Tracking Elements show a solid score of 0.7407 at 0-shot, and Contact improves to 0.3091 at 1-shot. Categories like Personal Identifier (0.0938 at 1-shot) and Survey Data (0.4000 at 0-shot) exhibit lower scores, suggesting classification challenges. The limited data for 2-shot and 3-shot settings indicates that 0-shot and 1-shot configurations are most effective, with performance varying by attribute. These findings underscore the importance of optimizing few-shot strategies for specific privacy-related classifications.

The following table summarizes the F1 scores for each attribute across the different few-shot learning configurations.

As shown in Prompt 5.1.2.1, the instruction template enforces a strict YES/NO response format.

For detailed analysis of few-shot prompting effects, see Figure B.9 in Appendix B.

Table 5.10: F1 scores for each attribute in 0, 1, and 2-shot settings during the few-shot experiment. Highest values in each row are bolded.

Attribute	**0-shot**	**1-shot**	**2-shot**
Cookies and Tracking Elements	0.7407	**0.7692**	0.6914
Computer Information	**0.2791**	**0.2791**	0.2632
Contact	0.3091	0.3269	0.2913
Financial	**0.6667**	0.5882	0.7097
Demographic	0.1429	0.1739	**0.1778**
Generic Personal Information	0.4825	**0.5040**	0.4800
Health	0.7059	**0.8889**	0.7778
IP Address and Device IDs	**0.6667**	0.5926	0.6000
Location	**0.5714**	0.5333	0.5000
Personal Identifier	0.0632	**0.0938**	0.0517
Survey Data	**0.4000**	**0.4000**	0.2857
User Profile	**0.1728**	0.1395	0.1224
User Online Activities	0.5045	0.4737	**0.5283**

5.1.5 Performance Evaluation of Gemini for Privacy Policy Classification

In this section, we evaluated the performance of Gemini model in identify whether the privacy policy states that it shares user's personal information type with third parties or not. We performed the same ablation study with limited criteria due to the budget limit of Gemini credits. We assessed the model's performance using two different criteria. Firstly, we tested the model's performance on different temperature levels and a few different shot examples. The results for each are mentioned below

5.1.5.1 Prompt engineering

For Gemini, we ran a few experiments to verify the prompt that works best for Gemini. Since it was a paid service. The testing remained limited for prompt engineering as well. The final prompt that we went with for the rest of the experiments is mentioned in Figure 5.1.5.1

Privacy Policy Cookie Analysis Prompt

System Instruction: You are a helpful assistant, expert in analyzing this privacy policy for third-party cookies and tracking elements disclosures. Respond ONLY with 'YES' or 'NO'."

Instructions:

1. Cookies and tracking elements are defined as 'Cookies, web beacons, pixel tags, or similar tracking technologies used to collect information about your activity.'
2. Only mark as YES if the policy clearly states that cookies or tracking elements are shared with third parties.
3. If the policy is unclear or doesn't explicitly mention sharing, mark as NO.
4. Be very strict in your analysis | require clear evidence.
5. If the policy says 'May be', treat it as YES.

User Input:

Read the privacy policy and answer ONLY "YES" or "NO". Does the following policy mention that it shares Cookies and tracking information with third parties?
Policy: [text]

5.1.5.2 Effect of Temperature Variation

This experiment evaluates the impact of temperature settings (0, 0.1, 0.2) on F1 scores when instructions and definitions are provided. Temperature controls the randomness of the model's output, with lower values producing more deterministic results.

The F1 scores vary across categories and temperature settings. For *Cookies and Tracking Elements*, the F1 score decreases from 0.6383 (temp 0) to 0.5254 (temp 0.2), indicating

that increased randomness harms performance, likely due to more false positives reducing precision. Similarly, *Computer Information* and *Demographic* show declines at higher temperatures (e.g., Demographic drops from 0.4545 to 0.2353), suggesting deterministic outputs are preferable for these categories. Conversely, *Health* improves from 0.8421 to 0.9000 at temperatures 0.1 and 0.2, and *IP Address and Device IDs* increases from 0.3500 to 0.5161, indicating that slight randomness aids in capturing sparse positive instances. *Generic Personal Information* peaks at 0.5822 (temp 0.1), while *Location* and *User Online Activities* show mixed results, with higher F1 scores at temp 0.2. These trends suggest that optimal temperature depends on the category, with low temperatures favoring precision-driven tasks and slight randomness benefiting recall-driven ones.
For detailed visual analysis of temperature effects, see Figure B.7 in Appendix B.

Table 5.11: F1 scores for all attributes across temperature settings (0, 0.1, 0.2). Highest values in each row are bolded.

Attribute	**Temp 0**	**Temp 0.1**	**Temp 0.2**
Cookies and Tracking Elements	**0.6383**	0.5391	0.5254
Computer Information	**0.3404**	0.2791	0.2857
Contact	0.2500	0.2500	**0.2698**
Demographic	0.4545	**0.5000**	0.2353
IP Address and Device IDs	0.3500	0.4571	**0.5161**
Health	0.8421	**0.9000**	**0.9000**
Generic Personal Information	0.5316	**0.5822**	0.5794
Location	0.4348	0.4167	**0.5455**
Personal Identifier	0.0909	0.1017	**0.1132**
Survey Data	0.2222	**0.4000**	0.2857
User Profile	0.1918	**0.2286**	0.2121
User Online Activities	0.5133	0.4865	**0.5243**
Financial	0.6316	**0.6471**	0.5882

5.1.5.3 Effect of Few-Shot Prompting

This experiment compares F1 scores under zero-shot, one-shot, and two-shot prompting strategies, where examples are provided to guide the model.
Few-shot prompting yields varied results. *Demographic* and *Financial* show consistent improvement, with F1 scores rising from 0.5000 to 0.6316 and 0.6471 to 0.6667, respectively, as shots increase, indicating that examples enhance model understanding for sparse categories. *IP Address and Device IDs* also improves from 0.5161 to 0.5833, benefiting from additional context. However, *Cookies and Tracking Elements* declines from 0.6383 to 0.5405, and *Generic Personal Information* drops from 0.5822 to 0.5104 with two shots, suggesting that excessive examples may introduce noise or ambiguity for complex categories. *Health* remains stable (0.9000 to 0.8889), reflecting robust performance regardless of examples. *Survey Data* and *User Profile* show declining F1 scores, indicating limited benefit from examples. Overall, few-shot prompting is effective for categories with few positive instances but can degrade performance for broader or noisier categories when too many examples are provided.
For comprehensive analysis of few-shot prompting effects, see Figure B.8 in Appendix B.

Table 5.12: F1 scores for each attribute in 0-shot, 1-shot, and 2-shot prompting with Gemini. Best score per row is bolded.

Attribute	0-shot	1-shot	2-shot
Computer Information	**0.3404**	0.3000	0.2667
Contact	0.2500	0.2632	**0.2830**
Cookies and Tracking Elements	**0.6383**	0.6200	0.5405
Demographic	0.5000	0.5882	**0.6316**
Financial	0.6471	0.6250	**0.6667**
Generic Personal Information	**0.5822**	0.5729	0.5104
Health	**0.9000**	**0.9000**	0.8889
IP Address and Device IDs	0.5161	0.5385	**0.5833**
Location	**0.5000**	0.4211	0.4348
Personal Identifier	0.1132	**0.1739**	0.1429
Survey Data	**0.4000**	0.2857	0.2222
User Online Activities	0.5243	**0.5273**	0.4870
User Profile	**0.2286**	0.1972	0.2000

5.1.6 Performance Evaluation of llama for Privacy Policy Classification

The effectiveness of large language models (LLMs) in information extraction tasks heavily depends on prompt design, particularly the use of in-context examples (Brown et al., 2020). This chapter presents an ablation study to evaluate how the number of examples in few-shot prompts impacts the extraction of three privacy-relevant data categories from policy texts:

5.1.6.1 Prompt engineering

When we did prompt engineering for Llama, we started with complex prompts including instructions and a lot of contextual knowledge, as we were doing with DeepSeek, Grok, and Gemini. But for Llama 3.1, the simplest clear prompts worked much better as compared to the prompts with multiple instructions domain knowledege. So, after testing multiple prompts, we finalized the following prompt mentioned in the figure 5.1.6.1 for Llama 3.1 8B experiments.

Cookies and Tracking Data Classification Prompt

```
You are a helpful privacy policy classifier. Your task is
to classify whether a privacy policy states that it shares
cookies or tracking data with third parties.

Rules:

  • Only answer "YES" if the policy mentions sharing or
    providing cookies and tracking data to third parties.

  • Only answer "NO" if the policy does not mention that
    cookies and tracking elements are shared with third
    parties.

  • If the policy states "May be", take it as NO.
```

Examples:
Policy: "We use Google Analytics to monitor performance."
Does this policy disclose sharing cookies or tracking data with third parties?
Answer: NO

Policy: "We share cookie and tracking information with our advertising partners to deliver personalized ads."
Does this policy disclose sharing cookies or tracking data with third parties?
Answer: YES

Now classify the following policy:
Policy: [text]
Does this policy disclose sharing cookies or tracking data with third parties?
Answer:

5.1.6.2 Effect of Temperature

We evaluate LLaMA's performance under three temperature settings: 0.0, 0.1, and 0.2. Table 5.13 shows the F1 scores across 13 attributes. The model performs best at temperature 0.0, with higher precision and F1 scores overall. As temperature increases, performance drops slightly across most categories, particularly in more ambiguous ones such as Personal Identifier and Contact Information.
For detailed visual analysis of temperature effects, see Figure B.10 in Appendix B.

Table 5.13: LLaMA F1 Scores Across Temperature Settings

Attribute	Temp 0.0	Temp 0.1	Temp 0.2
Financial	**0.2750**	0.2292	0.1980
Computer Information	**0.0302**	0.0299	0.0249
Contact	**0.2609**	0.2353	0.1958
Cookies and Tracking	**0.5143**	0.4533	0.3902
Demographic	**0.0811**	0.0784	0.0727
Generic Personal Info	0.2350	**0.2475**	0.2429
Health	**0.5714**	0.5000	0.3810
IP Address and Device IDs	**0.3077**	0.2909	0.1667
Location	0.1404	**0.1563**	0.1124
Personal Identifier	0.0284	**0.0458**	0.0253
Survey Data	**0.0426**	0.0000	0.0000
User Profile	0.0628	**0.0896**	**0.0896**
User Online Activities	**0.2160**	0.2121	0.1988

5.1.6.3 Effect of Few-Shot Prompting

To assess the value of in-context examples, we prompt the LLaMA model with 0-shot, 1-shot, and 2-shot examples. Results in Table 5.14 show that 1-shot prompting generally improves F1 scores across most categories. The 2-shot setting leads to further improvements in some cases (e.g., Financial, User Online Activities), but stagnates or slightly drops in others (e.g., Contact, Generic Personal Information), suggesting diminishing returns.
For comprehensive analysis of few-shot prompting effects, see Figure B.11 in Appendix B.

Table 5.14: LLaMA F1 Scores With Few-Shot Prompting

Attribute	0-Shot	1-Shot	2-Shot
Financial	0.2750	0.3607	**0.4255**
Computer Information	0.0302	0.0698	**0.0722**
Contact	0.2609	**0.4528**	0.3492
Cookies and Tracking	**0.5143**	0.4722	0.4722
Demographic	0.0811	0.1613	**0.1702**
Generic Personal Info	0.2475	**0.2857**	0.2688
Health	**0.5714**	0.0000	0.0000
IP Address and Device IDs	**0.3077**	0.1856	0.2903
Location	0.1563	**0.3810**	**0.3810**
Personal Identifier	0.0458	**0.0816**	0.0227
User Online Activities	0.2160	0.3776	**0.4381**

5.1.6.4 Effect of Instructions

Lastly, we examine the effect of explicit instructions. Table 5.15 compares model outputs with and without instruction guidance. Across nearly all attributes, the presence of task-specific instructions improves F1 scores. The most notable improvements appear in high-stakes categories such as Financial, Cookies and Tracking, and User Online Activities.
For detailed analysis of instruction effects, see Figure B.12 in Appendix B.

Table 5.15: LLaMA F1 Scores With and Without Instructions

Attribute	With Instructions	Without Instructions
Financial	**0.4255**	0.3607
Computer Information	**0.0722**	0.0504
Contact	**0.4528**	0.3596
Cookies and Tracking	**0.5143**	0.1634
Demographic	0.1702	**0.1852**
Generic Personal Info	0.1702	**0.2469**
Health	**0.5714**	0.1875
IP Address and Device IDs	**0.3077**	0.1887
Location	**0.3810**	0.2581
Personal Identifier	**0.0816**	0.0000
User Online Activities	**0.4381**	0.3015

5.1.7 Performance Evaluation of Mixtral-instruct for Privacy Policy Classification

As we fine tune Mixtral-8x7b-Instruct model, we ran a very basic inferance pipeline this model as well. The scope was only on the temperature and few shot prompting technique.

5.1.7.1 Prompt engineering

Similar to our older technique, we tried multiple prompts styles using Cookies ad tracking elements as the base. And then we decided which prompt gave the best results. Then we picked that prompt for the rest of out experimentation.
The prompt that we used for Mixtral-instruct is mention in the figure

Privacy Policy Cookie Analysis Prompt

System Instruction: You are a helpful assistant, expert in analyzing this privacy policy for third-party cookies and tracking elements disclosures. Respond ONLY with 'YES' or 'NO'."
Instructions:
1. "Cookies and tracking elements are defined as 'Cookies, web beacons, pixel tags, or similar tracking technologies used to collect information about your activity."
2. "Only mark as YES if the policy clearly states that cookies or tracking elements are shared with third parties"
3. "If the policy is unclear or doesn't explicitly mention sharing, mark as NO."
4. "Be very strict in your analysis - require clear evidence"
5. "If the policy says 'May be', treat it as YES"

User Input:

"Read the privacy policy and answer ONLY "YES" or "NO"."
"Does the following policy mentions that it shares Cookies and tracking information with third parties?"
Policy: [text]

5.1.7.2 Effect of Temperature

We evaluate Mixtral-instruct's performance under temperature setting 0.0. Table 5.16 shows the F1 scores across 13 attributes. The model demonstrates varying performance across different categories, with Cookies and Tracking Elements showing the highest F1 score of 0.5283, while Health Information Classification shows the lowest performance with an F1 score of 0.0000.

Table 5.16: Mixtral-instruct F1 Scores at Temperature 0.0

Attribute	Temp 0.0
Financial	0.4167
Computer Information	0.3333
Contact	0.1569
Cookies and Tracking	0.5283
Demographic	0.2353
Generic Personal Info	0.3537
Health	0.0000
IP Address and Device IDs	0.1765
Location	0.1333
Personal Identifier	0.0235
Survey Data	0.1818
User Profile	0.0795
User Online Activities	0.4082

5.1.7.3 Effect of Few-Shot Prompting

To assess the value of in-context examples, we prompt the Mixtral model with 0-shot, 1-shot, and 2-shot examples. Results in Table 5.17 show that few-shot prompting generally improves F1 scores across most categories. The 1-shot setting leads to improvements in several categories such as financial (0.4167 to 0.4444), Contact (0.1569 to 0.2712), and Health (0.0000 to 0.3704). However, some categories show performance degradation, indicating the sensitivity of the model to prompt design.

Table 5.17: Mixtral F1 Scores With Few-Shot Prompting

Attribute	0-Shot	1-Shot
Financial	0.4167	**0.4444**
Computer Information	**0.3333**	0.2500
Contact	0.1569	**0.2712**
Cookies and Tracking	**0.5283**	0.5185
Demographic	0.2353	**0.3750**
Generic Personal Info	–	**0.1667**
Health	0.0000	**0.3704**
IP Address and Device IDs	**0.1765**	0.1290
Location	**0.5000**	0.1739
Personal Identifier	**0.0000**	–
Survey Data	**0.2222**	0.0000
User Profile	0.0795	**0.1250**
User Online Activities	**0.4082**	0.3279

5.2 Model Performance Comparison

This section presents a comprehensive comparison of all models evaluated in the OPP_115 experimentation, focusing on overall performance patterns and computational efficiency.

5.2.1 Overall Performance Heatmap

The heatmap in Figure 5.1 shows the performance of all models across different personal information types. We used the best F1 score values for all information types from all ablation studies. This heatmap demonstrates the capacity of each model in predicting the correct values for all personal information types.

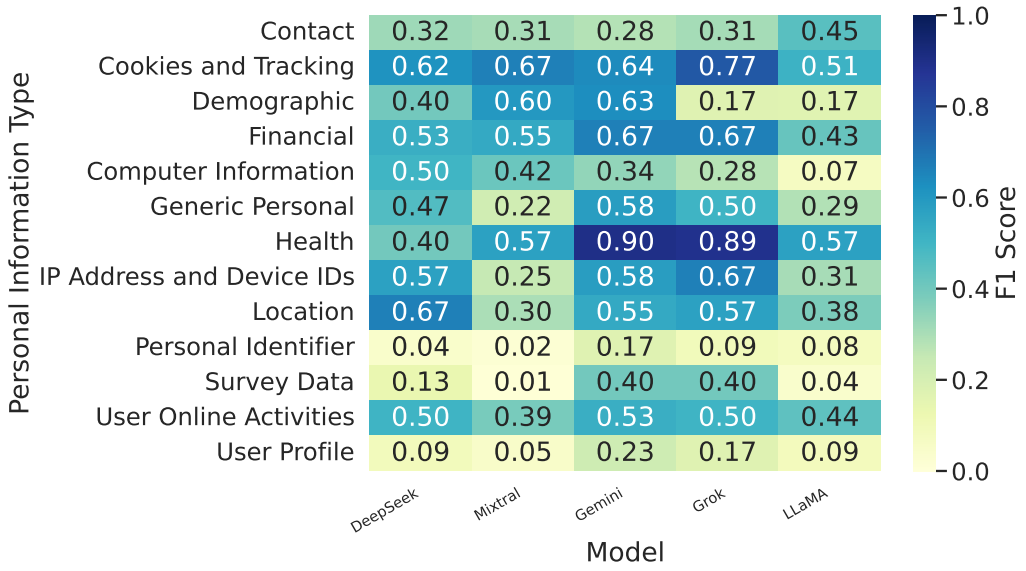

Figure 5.1: Heatmap of best F1 scores across all models and attributes.

Based on the heatmap, the best results are achieved by Gemini and Grok with 0.90 and 0.89 F1 scores, respectively, for health information classification. The worst performance was by LLaMA and Mixtral for survey data, with 0.04 and 0.02 F1 scores, respectively. It's worth noting that Gemini and Grok had limited ablation studies due to limited budget constraints, but they outperform even with the limited ablation studies.

5.2.2 Computational Performance Analysis

Figure 5.2 shows the average inference time per prompt for each model across 20 prompts. This analysis provides insights into the computational efficiency of different models when processing privacy policy classification tasks.

Gemini token length is exceptionally high as compared to others, as it also take into account thoughtsTokenCount along with the promptTokenCount. For Grok, as it is not open source, we calculated the token count from Grok API's.

Grok and Gemini have stable inference times. However, Mixtral has the most outliers, with the average of 4 seconds per prompt for the 20-prompt study for inference time vs. token length analysis.

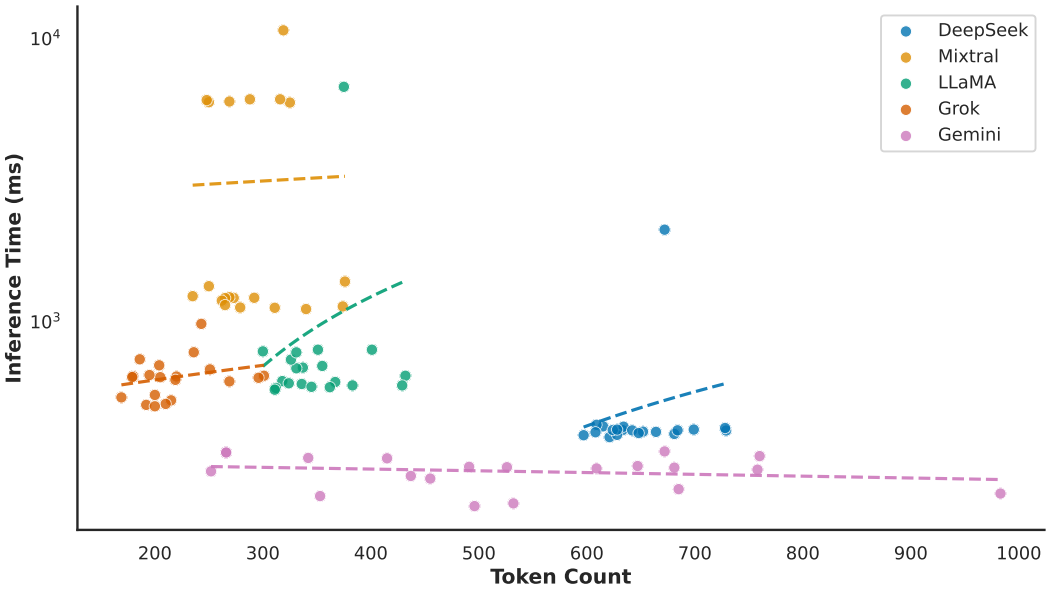

Figure 5.2: Average inference time per prompt for each model (20 prompts total)

The inference time analysis reveals significant variations in computational efficiency across models. For detailed ablation study results including temperature effects, few-shot prompting analysis, and cross-model comparisons, see Appendix B.

5.2.3 Temperature Effect Across Models

Figure 5.3 shows the results achieved by taking the average over all information types for the relevant temperature settings. Gemini shows the best performance over all temperatures. In general, almost all models perform better when temperature is 0 except for Gemini (which is better at 0.1 or 0.2).

5.2.4 Few-Shot Effect Across Models

Based on the results in Figure 5.4, Gemini performs best for all few-shot configurations, and the maximum average is achieved when 0-shot has been used. Then Grok is the winner with 1 shot. The worst performer is Mixtral at two shots, and overall, Mixtral is the worst.

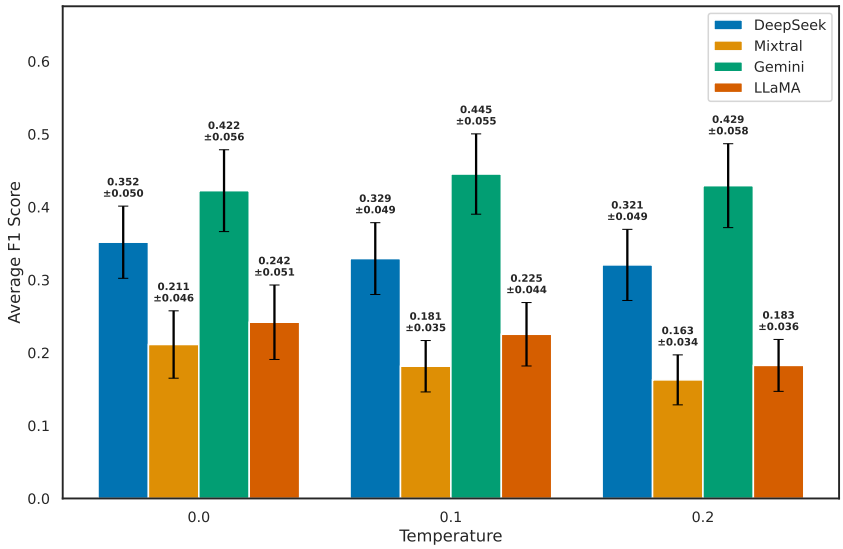

Figure 5.3: The impact of 3 different temperatures on 4 different models

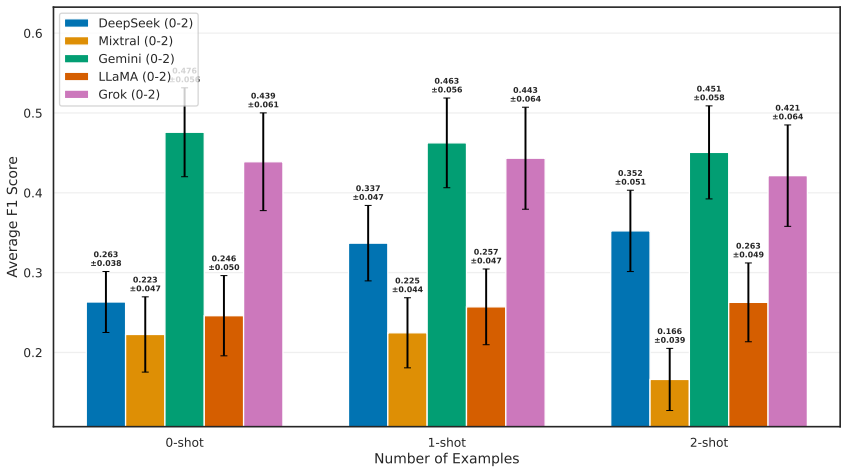

Figure 5.4: The impact of few-shot prompts on 5 different models

6

Fine-tuning Large Language Models for Privacy Policy Classification

Fine-tuning LLMs enables adaptation to specialized tasks, such as privacy policy classification, by leveraging domain-specific data. In this chapter, we systematically fine-tune and evaluate three LLMs using OPP_115 [12] dataset. The models included in our fine tuning study are Mixtral 8x7B, Llama 3.1-8B, and DeepSeek-R1-Distill-Qwen-32B. We used parameter-efficient techniques and prompt engineering to fine tune our models. We compared the performance of base (non-fine-tuned) and fine-tuned models and analyzed the impact of different fine-tuning strategies, such as fine-tuning using causal language modeling [70] and fine-tuning using sequence classification [71].

6.1 Fine-tuning Approaches

To evaluate different learning paradigms for privacy policy analysis, we explored two modeling approaches. The first approach was fine tuning for sequence classification, and the other appraoch was fine tuning for causal language modeling. For both of these approached we used the Hugging Face Transformers library: *AutoModelForSequenceClassification* for discriminative classification tasks, and *AutoModelForCausalLM* for generative language modeling tasks.

Each of these model classes wraps a pretrained transformer backbone and appends task-specific heads to adapt the model for the target task: Following is the explanation of the hugging face classes that we used for the fine tuning.

- **AutoModelForSequenceClassification:** We used this class for fine tuning our model for sequence classification tasks. This class adds a classification head usually a simple linear layer on top of the final hidden state of the model. It focuses on the output from a special token (like `<s>` or `[CLS]`) and learns to predict one or more labels from it [72]. When we use this with a model like Mixtral, it keeps the original transformer structure but adds this extra classification layer on top.

  Mixtral is based on a special architecture called a *mixture-of-experts* (MoE) trans-

former. Unlike regular transformers that activate all parts of the model for every input, Mixtral uses only a few selected "experts" (small subnetworks) for each token. Think of it like a team of specialists where, for each input, only 2 out of 8 experts are chosen to work—this makes the model faster and more efficient while still performing well [73].

- **AutoModelForCausalLM:** We used this class for fine tuning our model for causal modeling for language modeling tasks. This class configures the model for autoregressive generation by adding a language modeling head (usually a linear layer tied to the embedding layer) that outputs token probabilities at each time step. It enables the model to generate text one token at a time, conditioned on previous tokens [72].

6.1.1 Comparison of Fine-Tuning Strategies: Sequence Classification vs. Instruction-Based Causal Language Modeling

Table 6.1 presents a comparison between two widely used strategies for fine-tuning large language models: (1) sequence classification [74, 75] and (2) instruction-based causal language modeling (CLM)[76]. These approaches differ significantly in how inputs and outputs are structured, how they scale to lengthy or complex inputs, and what kind of downstream processing they require.

Sequence classification trains a model to assign discrete class labels (e.g., YES/NO) to a given input, making it straightforward and efficient for tasks like binary prediction or multi-label tagging. In contrast, instruction-based CLM involves prompting a model with task-specific instructions and allowing it to generate free-form text outputs. This method is more flexible and better suited to handling lengthy documents, though it often requires additional parsing or normalization of the generated outputs.

Table 6.1: Comparison of Fine-Tuning Strategies: Sequence Classification vs. Instruction-Based Causal Language Modeling

Aspect	**Sequence Classification**	**Causal LM (Instruction-based)**
Context Window	Depends on base model (e.g., 512 to 32k tokens)	Typically 600–2048+ tokens
Output Format	Class label	Generative
Long Document Handling	Limited by base model's context length	Excellent
Post-processing Required	YES (raw logits)	No (plain text)

The terms used in the above table are explained as follows

Context Window In the Table 6.1, context windows refer to the maximum number of tokens a model can process in a single forward pass. For example, BERT has a context window of 512 tokens [71], while newer models like Mixtral support windows from 2k up to 32k tokens [77, 78]. A larger context window allows the model to handle longer documents without truncation.

Output Format Sequence classification models return outputs such as logits numerical scores corresponding to class labels, rather than explicit answers like "YES" or "NO". These logits are post-processed (e.g., via softmax and argmax) to determine the predicted class. In contrast, causal language models generate open-ended token sequences, producing responses as natural language text based on the prompt [72].

6.2 Technical Details

This section outlines the key technical components of our approach to fine-tuning large-language models. We have mentioned the details of the preprocessing pipeline, addressed various challenges related to class imbalance, and explained how we applied parameter-efficient fine-tuning using QLoRA.

6.2.1 Preprocessing and Handling Class Imbalance

Before fine-tuning, we performed several preprocessing steps to improve data quality and address class imbalance:

- **Cleaning:** Removed special characters such as escaped quotes (e.g., \").
- **Class Imbalance:** As noted in Section 3.2.2.1, our dataset is highly imbalanced. The number of "NO" samples significantly outweighs "YES" samples across all information types. In order to divide our data into train, validation, and test splits, we employed stratified splitting [6] to ensure a proportional distribution of both classes across the training, validation, and test sets. The split distribution can be seen in the table. 3.6

  Stratification is essential in unbalanced classification tasks to avoid scenarios where a split may contain mainly one class, which would mislead model training or evaluation. By maintaining consistent class ratios in each split, the model is exposed to a representative distribution of the problem during training and validation.

  During fine-tuning, balanced class distributions often lead to more stable gradient updates and reduce bias toward the majority class, especially in binary tasks with severe skew.

- **Downsampling majority class:** For our experiments in this thesis, we opted for fine-tuning using the downsampled majority class. We kept all the positive samples, as they are already in the minority. We downsampled the majority class (NO label) and picked 30 percent of this class's samples in our experiments. [79]. We choose this value after testing the fine-tuning process with 25 percent, 30 percent, and 35 percent of "NO" samples. We used resample from the sklearn library to achieve this downsampling. The best results are obtained with 30 percent

- **Weighted sampling:** Even after downsampling the majority class, it was still 4 or 5 times more in number as compared to the minority class. For this reason, we applied a sampling strategy that gives more weight to underrepresented classes. We first calculated how frequently each class appeared in the training set and then derived weights

inversely proportional to those frequencies. This means classes with fewer examples received higher weights. These weights were then used to assign a sampling probability to each instance in the training set. During training, we used a sampling method that draws examples based on these probabilities, with replacement. This approach ensured that each mini-batch contained a more balanced representation of classes, helping the model learn more effectively from all categories rather than overfitting to the dominant class.

6.2.2 Parameter-Efficient Fine-Tuning with Instruction-Based Modeling

To enable efficient adaptation of large-scale language models on limited hardware, we employed parameter-efficient fine-tuning techniques combined with instruction-based training.

- **Low-Rank Adaptation (LoRA)** [80] is a technique that inserts small, trainable low-rank matrices (adapters) into transformer layers while freezing the original weights. This drastically reduces the number of trainable parameters and memory consumption, making it feasible to fine-tune models like Mixtral and Llama 2 on commodity hardware without significant loss in performance.
- **Quantized Low-Rank Adaptation (QLoRA)** [81] builds on LoRA by applying 4-bit quantization to the base model during training. This reduces GPU memory usage further while maintaining task performance, allowing fine-tuning of models with tens of billions of parameters on a pair of A100-80GB GPUs.
- **Adapter Saving and Merging:** During training, only the adapter weights were saved, and they were merged into the base model at inference time using PEFT's `merge_and_unload` function [82, 83]. Initially, we attempted to merge the adapters while the model was loaded in quantized precision (e.g., 8-bit), but observed no performance gains. Upon further investigation and reference to community discussions, we found that the LoRA merging process requires the base model to be in full (FP32) precision. Merging in quantized mode can lead to loss of adapter contributions due to rounding or clipping errors in low-bit representations, thus degrading the accuracy of the final model. When we merged the Lora adapters, we found overall improvements in our results.

To guide this fine-tuning process, we used the Supervised Fine-Tuning (SFT) Trainer with instruction-based data. The specific format of the data vary with the model we are fine-tuning. In our scenario, the best results were obtained when we used the chat template that is a part of model's tokenizer class. This means each example was presented as a natural-language chat format (e.g., "Does this privacy policy segment mention financial data being shared?") followed by the model's response ("YES" or "NO"). Instruction-based formatting aligns well with causal language models (CLMs), which are trained to autoregressively generate tokens given a prompt. The following pseudocode represents how we formated our instruction-based data for training the model. We first converted the prompts in the chat format in this pseudocode: 1. All of these chat templates are then

passed to the tokenizer's apply chat template function, which converts the chat into tokens that the model can understand. Each model has their own tokenizer and their own chat template. For example, for Mixtral, once we apply the chat template, we get the following string mentioned in this figure 6.1

Algorithm 1 Build Generic Chat Structure

function TO_CHAT_TEMPLATE(*example*)
    value ← UPPERCASE(TRIM(*example.output*))
    text ← CLEAN_WHITESPACE(*example.prompt*)

    chat ← [
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": INSTRUCTION + *text*},
        {"role": "assistant", "content": *value*}
    ]

    $$label \leftarrow \begin{cases} 1 & \text{if "YES"} \in value \\ 0 & \text{otherwise} \end{cases}$$

    return {text: *chat*, label: *label*}
end function

Algorithm 2 Apply Model-Specific Formatting

function APPLY_CHAT_TEMP(*example*)
    rendered ← *tokenizer*.apply_chat_template(*example.text*, *tokenize* = FALSE)
    return {text: *rendered*}
end function

Algorithm 3 Dataset Preprocessing Pipeline

Step 1: Convert to generic chat format
dataset ← *dataset*.map(to_chat_template)

Step 2: Apply tokenizer-specific formatting
dataset ← *dataset*.map(apply_chat_temp)

Data Collation and Loss Focus We used Hugging Face's Data Collator for completion language modeling during training to compute the loss only over the target response tokens (i.e., "YES"/"NO"), excluding the prompt from loss calculation. This ensures the model learns to generate accurate labels while ignoring the prompt itself for prediction. In our case, this collation strategy improved the model's ability to follow instructions and mitigated overfitting on the input format. Before using this strategy, the results were not satisfactory, but after we opted for data collation, our predictions got better.

```
<s> [INST] You are a legal expert specializing in privacy
policies.  Your task is to determine if a policy states that
cookies or tracking data are shared with third parties for
purposes like advertising or analytics.  Analyze the text
carefully and provide a definitive 'YES' or 'NO' answer.

Read the privacy policy and answer ONLY 'YES' or 'NO'.
Does the following policy mentions that it shares cookies
and tracking elements with third parties?
Data Retention Upon your request, we will remove your
personal information from the website but we will retain and
use your information as necessary to comply with our legal
obligations, resolve disputes, and enforce our Fool Rules
and take other actions otherwise permitted by law.  [/INST]
NO</s>
```

Figure 6.1: Example of formatted chat template for privacy policy analysis task

6.2.3 From Sequence Classification to Language Modeling: Rationale and Results

At the beginning of this research, we approached the privacy policy classification task as a standard binary classification problem. For testing purposes, we picked an example of cookies_and_tracking_elements to test the binary classification fine tuning strategy. The goal of this test experiment using cookies and tracking elements was to determine, for each policy, whether it mentions that the cookies' data is shared with third parties.
To test the classification feature of transformers, we fine-tuned the Mixtral 8x7B model using a sequence classification head, treating the task as a straightforward supervised classification problem. We assigned the labels as 0 and 1 for both binary classes.

Initial Results with Sequence Classification As mentioned above, we conducted a sample experiment using the "Cookies_and_tracking_elements" label to test the viability of sequence classification. After we fine-tuned the model using our training data, we observed an increase in F1 score. The Table 6.2 shows the results we achieved after fine-tuning with the training data.

Table 6.2: Sequence Classification Results for Cookies_and_tracking_elements

Model	TP	FP	FN	TN	Accuracy	Precision	Recall	F1
Mixtral8x7B	26	19	7	707	0.9657	0.5778	0.7879	0.6667
Fine-tuned Mixtral8X7B	26	14	7	712	0.9723	0.6500	0.7879	0.7123

Re-evaluating the Research Objective While these results were promising, we realized that this approach did not fully align with the core objective of our research. The primary aim was not just to classify texts, but to evaluate the capability of large language models (LLMs) to analyze privacy policy text and generate responses using their own language modeling abilities. In other words, the focus was on assessing the model's instruction-following and generative skills, rather than its performance as a classifier.

Transition to Language Modeling Based on this insight, we shifted our methodology to leverage the `AutoModelForCausalLM` architecture. This allowed the model to process instruction-based prompts and generate free-form answers, more closely simulating real-world applications where LLMs are expected to interpret and respond to complex queries in natural language. This approach also enabled the use of prompt engineering and instruction tuning, which are central to modern LLM applications.

6.3 Experiments and Results

We conducted fine-tuning experiments on three different language models: DeepSeek-R1, Mixtral, and LLaMA. We started with DeepSeek-R1 to learn the fine-tuning process and optimize hyperparameters, achieving a 6% improvement in F1 score. Each model helped us improve our fine-tuning techniques, with the experience from earlier models making the later experiments more effective. The following sections present the detailed results and analysis for each model.

6.3.1 DeepSeek-R1-Distill-Qwen-32B: Foundation Building and Systematic Optimization

In this section, we have mentioned the details of our foundational experiments with DeepSeek-R1-Distill-Qwen-32B, where we established our core fine-tuning methodology through systematic hyperparameter exploration.

Configuration and Methodology Development We started our fine-tuning with DeepSeek-R1-Distill-Qwen-32B, a 32B parameter model. We ran systematic experiments by testing different hyperparameters to check at which configuration we get the best performance of the model.

Initial Setup:

- Started with standard fine-tuning configurations
- Used QLoRA (Low-Rank Adaptation) for parameter efficiency
- Focused on the `Cookies_and_tracking_elements` analysis task as our primary benchmark

Progressive Hyperparameter Optimization Our optimization journey followed a systematic progression, with each experiment building upon insights from the previous one.

6.3.1.1 Learning Rate Exploration

We began by exploring different learning rates, as this is often the most critical hyperparameter for fine-tuning success. As shown in Table 6.3, by changing different learning rates, we saw changes in F1 score. The base DeepSeek-R1-Distill-Qwen-32B has an F1 of 61 percent for cookies and tracking elements, and we saw a 6 percent increase. In general, the Learning

rate 2×10^{-5} achieved the best F1 score, providing an optimal balance between the speed of convergence and the stability of the model.

Table 6.3: Learning Rate Impact on DeepSeek Performance

Learning Rate	**F1 Score**	**Key Observation**
1×10^{-5}	0.6462	Conservative, good precision
2×10^{-5}	**0.6667**	Optimal balance
5×10^{-5}	0.6118	Too aggressive, precision drops

6.3.1.2 Epoch Progression Analysis

After settling on an appropriate learning rate, we explored how training duration affects performance by testing different numbers of epochs, specifically 1, 2, and 3. Surprisingly, the best results came after just 1 epoch of fine-tuning.

In our experiment (Table 6.4), our data set was relatively small, only around 700 examples, and we were not training from scratch. We were fine-tuning DeepSeek, which was already pre-trained.

This suggests that for DeepSeek and small datasets, the low number of epochs performs better.

Table 6.4: Epoch Progression Analysis for DeepSeek

Epochs	**F1 Score**	**Precision**	**Recall**	**Key Insight**
1	0.6750	0.5745	**0.8182**	Peak performance
2	0.6562	0.6774	0.6364	Slight decline
3	0.5957	0.4590	0.8485	Clear overfitting

Critical Discovery: Performance peaked at just 1 epoch, with additional fine-tuning leading to overfitting.

6.3.1.3 Gradient Clipping Optimization

After tuning the learning rate and number of epochs, we focused on stabilizing training dynamics by adjusting the gradient clipping threshold. Gradient clipping is a technique used to prevent exploding gradients during backpropagation, particularly important when fine-tuning large models. It works by capping the maximum norm of the gradients, which helps ensure smoother and more stable updates.

We experimented with several values for the `max_grad_norm` parameter and observed their effect on model performance. The results are summarized in Table 6.5.

Table 6.5: Gradient Clipping Impact on Training Stability

Max Grad Norm	**F1 Score**	**Impact**
0.3	0.6667	Optimal stability
1.0	0.6667	Same performance
2.0	0.6329	Slight degradation

A value of 0.3 gave us the most consistent training behavior and highest F1 score. Increasing

the threshold to 1.0 did not hurt performance but also didn't yield further improvement. However, at 2.0, we observed a slight degradation in F1 score, suggesting that larger gradients may have introduced instability or overfitting.

6.3.1.4 Memory Management Strategy

Due to the limited GPU memory available during training, we had to carefully manage our batch size and gradient accumulation settings. Larger batch sizes improve training stability and convergence but require significantly more memory. To work around this, we used gradient accumulation, which simulates larger effective batch sizes without actually increasing the per-step memory footprint.
Table 6.6 shows the different configurations we tested.

Table 6.6: Memory Management Strategy for DeepSeek

Batch Size	**Gradient Accumulation**	**Effective Batch Size**	**Memory Usage**
16	8	128	CUDA Out of Memory
8	8	64	Manageable
4	8	32	Optimal
2	2	4	Conservative

We found that setting the batch size to 4 and the gradient accumulation steps to 8 gave us an effective batch size of 32. This configuration stayed within our memory limits and also maintained stable training performance. Larger batch sizes (like 128) consistently led to out-of-memory errors, while very small setups (like batch size 2) were too conservative and inefficient for fine-tuning. This trial-and-error tuning helped us strike a good balance between performance and hardware constraints.

6.3.1.5 Final Optimized Configuration

Through systematic experimentation, we established our optimal DeepSeek configuration as shown in Table 6.7:
Table 6.8 summarizes the performance improvements achieved through systematic optimization for the Cookies_and_tracking_elements classification task:
While DeepSeek provided excellent learning opportunities and solid performance gains, the computational intensity became a significant bottleneck. Each experiment required several hours of GPU time and made our exploration difficult. Because of this, we decided to switch to Mixtral for the remaining experiments. The key motivation was its faster turnaround time, which allowed us to run more experiments, iterate quickly, and fine-tune hyperparameters more efficiently. Faster feedback from the training loop made it easier to explore and optimize the model.

Table 6.7: Final Optimized Configuration for DeepSeek-R1

Parameter	**Value**
QLoRA Configuration	
quantization	4-bit
r	4
lora_alpha	16
target_modules	[”q_proj”, ”v_proj”]
lora_dropout	0.1
bias	none
task_type	CAUSAL_LM
Training Arguments	
learning_rate	2×10^{-5}
num_train_epochs	1
per_device_train_batch_size	4
gradient_accumulation_steps	8
max_grad_norm	0.3
warmup_ratio	0.03
lr_scheduler_type	cosine

Table 6.8: DeepSeek Performance Achievement for Cookies_and_tracking_elements

Metric	**Value**
Baseline F1 Score	0.6176
Optimized F1 Score	0.6750
Absolute Improvement	6%
Relative Improvement	9.3%

6.3.2 Mixtral 8x7B

This section reports our fine-tuning experiments with the Mixtral 8x7B model. We started from the same parameters that we used for DeepSeek. But since the model took fewer hours as compared to deepseek, we tried multiple scenarios while fine-tuning Mixtral. Throughout the experiments, Our focus was instruction following and the learning rate tuning while keeping all other hyperparameters constant (finalized during DeepSeek fine-tuning) during the fine-tuning process.

6.3.2.1 Learning Rate Exploration

We began with a baseline experiment using the base Mixtral 8x7B model.
During our experiments, to improve performance, we conducted a series of controlled ablation studies targeting the `Cookies_and_tracking_elements` label. The learning rates explored in these experiments ranged from 1×10^{-5} to 1×10^{-4}, specifically: 1×10^{-5}, 2×10^{-5}, 5×10^{-5}, and 1×10^{-4}. This sweep was designed to evaluate how both conservative and aggressive learning rates influence model performance when fine-tuning on a binary classification task. However, across these configurations, we observed only marginal differences in F1 scores, and none of them surpassed the performance achieved by the instruction-tuned version of the Mixtral model.

Table 6.9: F1 scores for different learning rates on `Cookies_and_tracking_elements` (Mixtral 8x7B)

Learning Rate (LR)	**F1 Score**
2×10^{-5}	0.1818
1×10^{-4}	0.5091
5×10^{-5}	0.5634
2×10^{-5}	0.5538
1×10^{-5}	0.4706
5×10^{-5}	0.6154
5×10^{-5}	0.5974
5×10^{-5}	0.5435

6.3.2.2 Limitations of Mixtral 8x7B

All results in this section are based solely on experiments conducted with Mixtral 8x7B. The base Mixtral 8x7B has F1 score of 66%. However, even though they perform none of the fine-tuned model results exceed the Mixtral 8 x 7B base model's F1 score for cookies and tracking elements. This made us think about reevaluating the steps that we are doing. Through further investigation, we found consistent evidence both in published research and community practice that starting from a base (non-instruction-tuned) model is often less effective for instruction-style tasks. The QLoRA study shows that instruction-aligned models, when further fine-tuned, achieve near-state-of-the-art results using less compute and time [81]. Similarly, the developer community also reported that "base models suck at properly answering human instructions" and recommended beginning with instruction-tuned variants. Based on these results (summarized in Table 6.9), we shifted our strategy. Instead of continuing to fine-tune the base model from scratch, we opted to fine-tune an instruction-aligned Mixtral checkpoint using QLoRA and structured, chat-style prompts. This change led to a noticeable improvement in performance, suggesting that starting from a model already aligned for instruction-following tasks is more effective than training from a raw foundation model.

6.3.3 Mixtral 8x7B-Instruct

From the above results, we switched to the Instruct-tuned Mixtral 8x7B model to verify the claims explained in Section 6.3.2.2. We tried a few learning rates and observed differences in the fine-tuned model results.

6.3.3.1 Learning rate Exploration

We tried multiple learning rates with Mixtral 8x7B-instruct model. We acheived good results with out third learning rate which is 8e-6. The following table 6.10 shows the learning rates we tried and their respective F1 score for cookies and tracking elements. The complete experiments for mixtral are mentioned in the table below 6.12

Table 6.10: Learning Rate Impact on F1 Score for Cookies and Tracking Elements Detection

Learning Rate	**F1 Score**
2×10^{-6}	0.4490
2×10^{-5}	0.5376
8×10^{-6}	**0.7013**

6.3.3.2 Epoch Progression Analysis

While fine-tuning the Mixtral-8x7B-instruct model, we tested multiple learning rates. These learning rates range from 6, 8, 12, and 20. We observed when the epoch was set at 12, we got the best results. The results are for F1 score for all the epochs accidentally got deleted. The best score that we got was mentioned in the table. 6.10

6.3.3.3 Final Optimized Configuration

The full hyperparameter configuration used for the final Mixtral 8x7B -instruct fine-tuning experiment is detailed in Table 6.11.

Table 6.11: QLoRA (4-bit) Fine-Tuning Hyperparameters for Mixtral 8x7B

Parameter Category	**Parameter**	**Value**
QLoRA (4-bit Quantization)	quantization	4-bit
	r	4
	lora_alpha	8
	target_modules	[q_proj, k_proj, v_proj, o_proj]
	lora_dropout	0.1
	bias	none
	task_type	CAUSAL_LM
Training Arguments	learning_rate	8×10^{-6}
	num_train_epochs	12
	per_device_train_batch_size	2
	max_grad_norm	0.3
	warmup_ratio	0.03

Table reftab:f1-comparison shows the results that we achieved using Mixtral8x7B-instruct 8e-6, and also its comparison with the base model.

Table 6.12: F1 Score Comparison Before and After Fine-Tuning

Label	**Base Model (F1 Score)**	**Fine-tuned Model (F1 Score)**
Cookies and Tracking Elements	0.5283	0.7013
Generic Personal Information	0.3537	0.4066
IP Address and Device IDs	0.1765	0.3256
Financial	0.2963	0.5556
Computer Information	0.3333	0.3226
User Online Activities	0.3958	0.3529
User Profile	0.1250	0.0000

Based on the results in the Table 6.12, it is evident that instruction-tuned models are better for instruction-based fine-tuning. However, during our experimentation, fine-tuning these led to mixed results as well. Attributes such as cookies and tracking elements and

financial showed clear gains, whereas others like user profile and user online activities saw little to no improvement. This discrepancy likely arises from label distribution imbalance, low prevalence of positive examples, or inherent ambiguity in some categories. For example, attributes like User Profile or Computer Information may be more context-dependent and harder to detect from short policy segments, making them challenging to learn from limited examples,even when fine-tuning. In contrast, more explicitly stated practices like cookies and tracking elements are easier for the model to capture and generalize.

6.3.4 Llama 3.1–8B

This section outlines the fine-tuning configuration and experimentation process used with the Llama 3.1–8B Instruct model. We did not test with Llama 3.1-8B. We directly started our experimentation with the instruction-tuned version of this model.

6.3.4.1 Learning rate exploration

We started testing llama with the same learning rate as mixtral, but we didnt see any improvements in Llama. So, we did an exploration study for llama as well. In the following table 6.13, we have mentioned the results we got when we tried Llama with multiple learning rates.

Table 6.13: LLaMA 3.1–8B Instruct Results with QLoRA and Prompt Variations

LR	**Epochs**	**Prompt**	**Acc.**	**Prec.**	**Recall**	**F1**	**FP**	**QLoRA**
Base Model (No Fine-tuning)			0.9552	0.4889	0.6667	0.5641	23	False
8×10^{-6}	2	No few-shot	0.8524	0.2199	0.9394	0.3563	110	True
2×10^{-6}	2	2 NO, 1 YES	0.9592	0.5263	0.6061	0.5634	18	True
2×10^{-6}	2	2 NO, 2 YES	0.9196	0.3158	0.7273	0.4404	52	True
2×10^{-5}	2	No emphasis	0.8011	0.1793	1.0000	0.3041	151	True
2×10^{-5}	2	NO-emph.	0.9420	0.4225	0.9091	0.5769	41	False

The LLaMA 3.1-8B-instruct results show that there is a critical learning rate sensitivity issue when fine-tuning for this classification jobs. When the learning rates were higher (2×10^{-5} and 8×10^{-6}), the model became too aggressive in predicting positive instances. The recall improved to a high extent (0.9394–1.0000) but low precision (0.1793–0.2199), and it made too many false positives (110–151). The best setup employed a lower learning rate (210^{-6}) and "2 NO, 1 YES" few-shot prompting. It had the best results with an F1 score of 0.5634, a precision of 0.5263, and just 18 false positives.

6.3.4.2 Epoch Progression Analysis

For our experiments, we only tried 2 learning rate which were 2 and 3. Although because of the time and GPU constraints, we couldn't run multiple experiments to reach a better F1 score. But with our limited research, Epoch 2 resulted in a better F1 score as shown in the table 6.14. So, we used this for the rest of our experiments.

Table 6.14: Impact of Training Epochs on F1 Score (Learning Rate: 2×10^{-6}, No Few-shot)

Epochs	F1 Score
2	0.5385
3	0.5283

Llama 3.1 Limitation: For Llama 3.1–8B Instruct, we adopted a similar approach as we did with mixtral 8x7B instruct model. We started fine tuning the model using QLoRA, which loads the model in 4 bit quantization. But even after trying multiple learning rates, the performance of the model wasn't surpassing the F1 score that of llama 3.1-8B Instruct, even though we tried multiple learning rates.
We tried to use the prompt engineering strategy and few-shot learning technique while fine-tuning. All of these techniques had an intense effect on false positives. After running multipe experiments, we came accross an article [84], in which they tested out the performance of QLora on Llama3. Their studies showed that the performance of Llama, when fine-tuned in a quantized format, degrades the overall performance. So, with that, we removed the QLoRA configuration and fine-tuned the model in 16-bit quantization.
The results for all the experiments are as follow in 6.13
This section reports fine-tuning results of the LLaMA 3.1–8B Instruct model using QLoRA under different prompt and learning rate configurations, highlighting the impact of prompt engineering versus hyperparameter tuning.

Key Results As shown in Table 6.13, the most substantial gains in F1 score were not achieved through hyperparameter tuning alone, but through careful prompt engineering—particularly those that emphasized the NO class and included hard negatives. For instance, the highest F1 score of 0.5769 was achieved with a prompt that explicitly emphasized conservative NO responses at a learning rate of 2×10^{-5}.
In contrast, the same learning rate with no prompt emphasis yielded a much lower F1 score (0.3041) and a significantly higher false positive count. This suggests that the framing and balance of examples in the prompt can affect the model's precision-recall tradeoff, often more than adjusting learning rates or batch sizes.
The final configuration we came up with is mentioned in the table. 6.15

Table 6.15: Hyperparameters used for LoRA fine-tuning of LLaMA 3.1-8B

Parameter Category	Parameter	Value
LoRA Config	r	4
	lora_alpha	8
	target_modules	[q_proj, k_proj, v_proj, o_proj]
	lora_dropout	0.1
	bias	none
	task_type	CAUSAL_LM
Training Arguments	learning_rate	2×10^{-6}
	num_train_epochs	2
	per_device_train_batch_size	2
	max_grad_norm	0.3
	warmup_ratio	0.03

Limitations of QLoRA Fine-Tuning Although QLoRA makes fine-tuning large models feasible on limited hardware by compressing weights to 4-bit, this aggressive quantization introduces trade-offs. Huang et al. [85] found that ultra-low bit precision can degrade performance in tasks requiring subtle semantic distinctions or precise reasoning—both critical in privacy policy classification. These limitations are consistent with our findings: despite aggressive prompt optimization, fine-tuned models did not outperform the base model significantly for certain information types, likely due to quantization-induced loss of expressiveness.

The fact that the base model (non-QLoRA, full precision) achieved a comparable F1 score of 0.5641 without fine-tuning suggests that QLoRA's efficiency comes at a cost, especially in domains requiring nuanced understanding.

6.4 Key Insights

Through our systematic experimentation across three distinct LLM architectures (DeepSeek-R1, Mixtral, and LLaMA), we derived several critical insights that inform best practices for fine-tuning large language models on privacy policy classification tasks. These findings emerged from our progressive methodology development and comparative analysis of different fine-tuning strategies.

- **Foundation Model Selection is Critical:** Instruction-tuned models consistently outperformed their base counterparts across all architectures. For instance, Mixtral-Instruct achieved an F1 score of 0.7013 compared to the base Mixtral's best performance of 0.6154, representing a 32% improvement. This validates the importance of starting with models already aligned for instruction-following tasks rather than attempting to train base models from scratch.

- **Prompt Engineering vs. Fine-tuning Trade-offs:** For LLaMA 3.1-8B, careful prompt engineering achieved comparable or superior results to hyperparameter tuning. The highest F1 score (0.5769) was obtained through prompt optimization during fine-tuning rather than learning rate adjustments, suggesting that well-designed prompts can also be more practical than extensive fine-tuning for certain applications.

- **Few-shot Learning and Class Balance:** Incorporating balanced few-shot examples can be beneficial in case when model is over predicting one class.

- **QLoRA Efficiency Trade-offs:** While QLoRA enables efficient fine-tuning on consumer hardware, our results with LLaMA 3.1-8B suggest that 4-bit quantization may limit performance in tasks requiring nuanced semantic understanding. The base model (full precision) achieved comparable performance (F1=0.5641) to the best fine-tuned QLoRA variant (F1=0.5769), indicating potential expressiveness loss due to aggressive quantization.

- **Architecture-Specific Optimization:** Each model architecture required tailored approaches. DeepSeek benefited from systematic hyperparameter optimization, Mix-

tral from instruction-tuning alignment, and LLaMA from prompt engineering, highlighting the importance of architecture-aware fine-tuning strategies.

6.5 Fine-tunning Efficiency Analysis

Figure 6.2 shows the average time taken per epoch for fine-tuning three different instruction-tuned models: Mixtral-8x7B-Instruct, DeepSeek-R1-Distill-Qwen-32B, and Llama-3.1-8B-Instruct. Among them, DeepSeek was the slowest, taking around 23 minutes per epoch, while Llama 8B, because of fewer parameters, was the fastest at just 9 minutes.

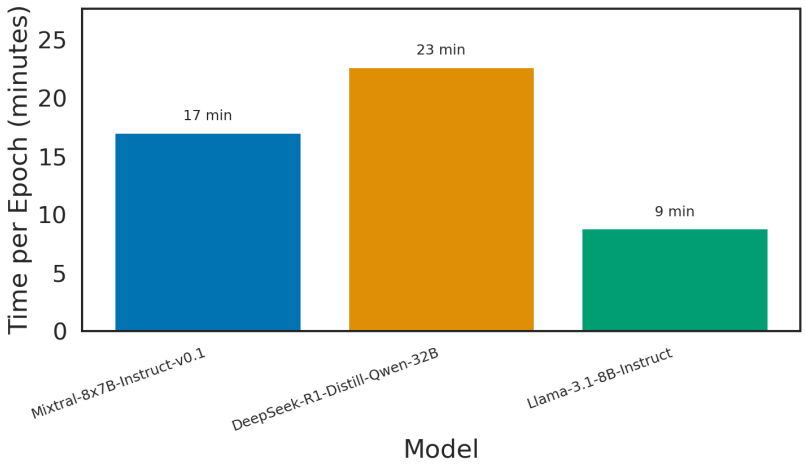

Figure 6.2: Average training time per epoch for Mixtral, DeepSeek-R1-Distill-Qwen-32B, and LLaMA.

The results demonstrate significant variations in computational time requirements across architectures. DeepSeek-R1-Distill-Qwen-32B, with its 32B parameters, required longer training times per epoch as compared to other less parameter models.
The above training time that we have reported for each epoch does not reflect the total fine tuning time for a training run. In addition to the actual training steps, validation after each epoch, checkpoint saving, and other system overheads (like memory syncing and logging) contribute significantly to the total duration. As a result, even though logs showed 23 minutes per epoch, the full run with 5 epochs often took 4–6 hours.

6.6 Conclusion

Our systematic fine-tuning study across three diverse LLM architectures revealed several important findings for privacy policy classification that extend beyond the specific domain to inform broader LLM adaptation strategies.
Foundation Model Criticality: The choice of foundation model emerged as the most significant factor affecting performance. Instruction-tuned variants consistently outperformed base models, with Mixtral-Instruct achieving 32% higher F1 scores than its base counterpart. This finding underscores the importance of model selection over extensive hyperparameter

optimization when working with limited computational resources.

Prompt Engineering as Alternative Strategy: Perhaps most significantly, our LLaMA experiments revealed that careful prompt engineering can match or exceed the performance gains achieved through traditional fine-tuning. This finding suggests that for resource-constrained applications or rapid prototyping scenarios, investing in prompt design may yield better returns than extensive hyperparameter tuning.

Practical Implications: For practitioners working on similar text classification tasks in specialized domains, our results suggest a tiered approach: (1) begin with instruction-tuned models, (2) invest in prompt engineering before extensive fine-tuning, and (3) when fine-tuning is necessary, start with conservative hyperparameters and systematic single-parameter optimization.

These findings contribute to the broader understanding of LLM adaptation strategies and provide concrete, evidence-based guidance for researchers and practitioners working on domain-specific text classification tasks. The methodological framework developed through this study offers a replicable approach for systematic LLM optimization under computational constraints.

7

Conclusion and future work

7.1 Conclusion

In this thesis investigated the effectiveness of large language models (LLMs) in the identification of third-party data sharing practices within privacy policies. We evaluated five modern LLMs DeepSeek-R1-Distill-Qwen-32B, Mixtral, LLaMA, Grok, and Gemini, across 15 distinct categories of personal information using OPP_115 and MAPP datasets. Our experiments included structured prompting, ablation studies for various components of prompts, and parameter-efficient fine-tuning (LoRA/QLoRA).

The results demonstrate that, when we use well-guided prompts, we can acheive high F1 scores. The other main point to note here is that LLM performance also varies by different information types. On well-defined and distinctive categories such as *cookies and tracking elements* and *financial information,* LLMs perform better. But for vague or general information types, the performance decreases for most of the models. Structured outputs, low-temperature inference, and instruction phrasing were among the most influential factors affecting performance. While some models, such as Grok nd Gemini, consistently performed well among the other open source models

In many cases, prompt engineering outperformed full fine-tuning in terms of efficiency and reliability. Overall, combining different prompt strategies produced measurable gains for the information types.

For specific models and labels, fine-tuning using QLoRA techniques yielded additional improvements, but the marginal gains must be weighed against the required compute, data balancing strategies, and tuning sensitivity. Our experiments confirmed that a hybrid strategy, combining targeted prompt engineering with lightweight fine-tuning, is the most efficient approach for real-world LLM-based policy analysis.

7.2 Limitations and Future Work

This study has a few limitations:

- **Class Imbalance**: The OPP-115 and MAPP datasets both include a lot of class imbalance, notably in sensitive areas like *Health*, *Survey Data* etc. This makes it

hard for LLMs to learn where the right boundaries are for qualities that aren't very common.

- **Limited Multilingual Evaluation**: MAPP works with both English and German, but our evaluation was only about policies in English. More research is needed to find out how well LLM works in situations when there are more languages involved in the research.
- **Dependence on Prompt Sensitivity**: The results of many models varied greatly based on how the prompt was designed. Small modifications in wording or examples have effects on F1 scores. This suggest to invest time in engineering the best prompt for each model separately.
- **Evaluation Scope**: We looked at just YES/NO analysis for the sharing of data with Third parties for each segment and each attribute. It doesn't currently deal with end-to-end summarization, which could also help users in understanding the privacy policies and making informed decisions.
- **Platform Constraints**: Proprietary models like Grok and Gemini had problems since they offer paid services and couldn't be fine-tuned. The amount of available GPU also limited some tests for open source models.

This is the base that future work can build on in the following ways: This study has a few problems, even though it got some good results:

1. Multilingual Expansion: Use enlarged MAPP or newly scraped corpora other than the English language to evaluate more models for different languages.
2. **Justification/Summarization capabilities**: Analyse how well LLMs can give not just YES/NO answers but also *justifications* or summarization for their classifications to make them more trustworthy and useful.
3. Real-time systems: Use tailored LLMs to make browser plugins or compliance solutions that automatically highlight third-party disclosures for end users.
4. Cross-Document Analysis: For fine-tuning, more experiments can be done with more learning rates and variation in hyper parameters parameters to acheive a substantial gain in F1 score.

By looking into these areas, future research can get closer to making useful tools that help regulators, researchers, and end users find their way through the confusing world of privacy policies.

Bibliography

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679 (2016). Official Journal of the European Union, L 119/1.

[2] Obar, J. A. and Oeldorf-Hirsch, A. The biggest lie on the Internet: ignoring the privacy policies and terms of service policies of social networking services. *Information, Communication & Society*, 23(1):128–147 (2020). URL https://doi.org/10.1080/1369118X.2018.1486870.

[3] Obar, J. Unpacking "the Biggest Lie on the Internet" Part Two: An Assessment of the Complexity of Terms of Service and Privacy Policies for 70 Digital Services. *SSRN Electronic Journal* (2024).

[4] Senarath, A., Arachchilage, N., and Slay, J. Designing Privacy for You : A User Centric Approach For Privacy (2017).

[5] Korunovska, J., Kamleitner, B., and Spiekermann-Hoff, S. The challenges and impact of privacy policy comprehension. In {Association for Information Systems}, editor, *Twenty-Eighth European Conference on Information Systems (ECIS2020)*, pages 1–17 (2020).

[6] Wagner, I. Privacy Policies Across the Ages: Content and Readability of Privacy Policies 1996–2021 (2022). URL https://arxiv.org/abs/2201.08739.

[7] Reidenberg, J., Bhatia, J., Breaux, T., and Norton, T. Automated Comparisons of Ambiguity in Privacy Policies and the Impact of Regulation. *SSRN Electronic Journal* (2016).

[8] California State Legislature. California Consumer Privacy Act of 2018 (2018). URL https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375. Assembly Bill No. 375.

[9] McDonald, A. M. and Cranor, L. F. The cost of reading privacy policies. *I/S: A Journal of Law and Policy for the Information Society*, 4(3):543–568 (2008). URL https://api.semanticscholar.org/CorpusID:197633124.

[10] Wang, Y., Xu, C., Wang, C. a., Wang, H., and Zhao, G. Demystifying Privacy Policy of Third-Party Libraries in Mobile Apps. In *Proceedings of the 45th International Conference on Software Engineering*, pages 1679–1691. IEEE (2023).

[11] Arora, A., Silcock, E., Heldring, L., and Dell, M. A tale of two regulatory regimes: Creation and analysis of a bilingual privacy policy corpus. In *Proceedings of LREC*, pages 5452–5462 (2022).

[12] Wilson, S., Schaub, F., Dara, A., and et al. The Creation and Analysis of a Website Privacy Policy Corpus. In *Proceedings of ACL*, pages 1330–1340 (2016).

[13] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgesius, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[14] Reynolds, L. and McDonell, K. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. arXiv preprint arXiv:2102.07350 (2021).

[15] Zimmeck, S., Wang, P., Zou, L., and et al. Automated analysis of privacy requirements for mobile apps. *Proceedings on Privacy Enhancing Technologies*, 2017(2):122–142 (2017).

[16] Rodriguez, A., Silcock, E., Heldring, L., and Dell, M. Large Language Models for Privacy Policy Analysis: A Comparative Study. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*. Association for Computational Linguistics (2024). URL https://arxiv.org/abs/2406.15576.

[17] Tene, O. and Polonetsky, J. Big Data for All: Privacy and User Control in the Age of Analytics. *Northwestern Journal of Technology and Intellectual Property*, 11 (2012).

[18] Voigt, P. and Von dem Bussche, A. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing (2017).

[19] Goldman, E. and Barton, P. *An Introduction to Privacy Law*. West Academic Publishing, St. Paul, MN, 2nd edition (2020).

[20] Reidenberg, J. R. E-Commerce and Trans-Atlantic Privacy. *Houston Law Review*, 38(3):717–749 (2001). Symposium: Law of Electronic Commerce.

[21] Contissa, G., Lippi, M., Sartor, G., Torroni, P., and Bartolini, C. CLAUDETTE: An Automated Detector of Potentially Unfair Clauses in Online Terms of Service. *Artificial Intelligence and Law*, 26(2):103–123 (2018).

[22] Costante, E., Sun, Y., Petković, M., and Etalle, S. A machine learning solution to assess privacy policy completeness: (short paper). *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pages 91–96 (2012).

[23] Adhikari, A., Das, S., and Dewri, R. Natural Language Processing of Privacy Policies: A Survey (2025).

[24] Keymanesh, M., Elsner, M., and Parthasarathy, S. Toward Domain-Guided Controllable Summarization of Privacy Policies. In *Proceedings of the 2020 Natural Legal Language Processing (NLLP) Workshop*, pages 18–24. ACM, New York, NY, USA (2020). URL http://ceur-ws.org/Vol-2645/paper3.pdf.

[25] Del Alamo, J., Guaman, D., García, B., and Díez Medialdea, A. A systematic mapping study on automated analysis of privacy policies. *Computing*, 104 (2022).

[26] Zimmeck, S. and Bellovin, S. M. Privee: An architecture for automatically analyzing web privacy policies. *Proceedings of the 23rd USENIX Security Symposium*, pages 639–654 (2014).

[27] Wilson, S. et al. The creation and analysis of a website privacy policy corpus. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1335–1344 (2016).

[28] Harkous, H. et al. Polisis: Automated analysis and presentation of privacy policies using deep learning. *Proceedings of the 29th USENIX Security Symposium*, pages 531–548 (2020).

[29] Alshamsan, A. and Chaudhry, S. Machine Learning Algorithms for Privacy Policy Classification: A Comparative Study. pages 214–219 (2022).

[30] Devlin, J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[31] Brown, T. B. et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901 (2020).

[32] Ravichander, A. et al. Question answering for privacy policies: Combining computational and legal perspectives. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3550–3559 (2019).

[33] Rogers, A., Kovaleva, O., and Rumshisky, A. A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 61:65–95 (2020).

[34] Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., and Huang, X. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897 (2020).

[35] Kenton, J. D. M.-W. C. and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2019).

[36] Conneau, A. and Lample, G. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems*, 32 (2019).

[37] Hu, J., Ruder, S., Siddhant, A., Neubig, G., Firat, O., and Johnson, M. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *International Conference on Machine Learning*, pages 4411–4421 (2020).

[38] Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in NLP. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650 (2019).

[39] Bender, E. M. et al. On the dangers of stochastic parrots: Can language models be too big? *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623 (2021).

[40] Blodgett, S. L., Barocas, S., Daumé III, H., and Wallach, H. Language (technology) is power: A critical survey of "bias" in NLP. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476 (2020).

[41] Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215 (2019).

[42] Lipton, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57 (2018).

[43] Harkous, H., Rahman, M., Karlas, B., Fawaz, K., Aberer, K., and Shafiq, Z. Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 531–548. USENIX Association (2018). URL https://www.usenix.org/conference/usenixsecurity18/presentation/harkous.

[44] Zhang, S., Yi, X., Li, S., Xing, H., and Li, H. PrivCAPTCHA: Interactive CAPTCHA to Facilitate Effective Comprehension of APP Privacy Policy. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–20. ACM (2025). URL https://dl.acm.org/doi/10.1145/3706598.3713928.

[45] CLEAR: Towards Contextual LLM-Empowered Privacy Policy Analysis and Risk Generation for Large Language Model Applications. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*. ACM (2025). URL https://dl.acm.org/doi/10.1145/3708359.3712156.

[46] Anonymous. You Don't Need a University Degree to Comprehend Data Protection This Way: LLM-Powered Interactive Privacy Policy Assessment (2024). URL https://www.researchgate.net/publication/389616490. ResearchGate preprint.

[47] AI, D. DeepSeek-R1-Distill-Qwen-32B. https://huggingface.co/deepseek-ai/DeepSeek-V2-GPTQ (2024). Accessed: 2025-07-09.

[48] AI, D. DeepSeek-R1-Distill-Qwen-32B. https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B (2024). Available at https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B.

[49] DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu,

K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning (2025). URL https://arxiv.org/abs/2501.12948.

[50] Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Bou Hanna, E., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Le Scao, T., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and El Sayed, W. Mixtral of Experts: Sparse Mixture-of-Experts Language Model (2024). URL https://arxiv.org/abs/2401.04088. ArXiv preprint arXiv:2401.04088.

[51] Team, M. A. Mixtral of Experts (2023). URL https://mistral.ai/news/mixtral-of-experts. Model announcement blog post.

[52] Dubey, A., Jauhri, A., et al. The Llama 3 Herd of Models. *arXiv* (2024). URL https://arxiv.org/abs/2407.21783.

[53] xAI. Grok 3 Beta Release. https://x.ai/blog/grok-1.5-release (2024). Accessed: 2025-07-09.

[54] DeepMind, G. Gemini 2.5 and Gemini Flash: Updates. https://blog.google/technology/ai/google-gemini-updates-may-2024/ (2024). Accessed: 2025-07-09.

[55] Jiang, A. Q., Sablayrolles, A., et al. Mixtral of Experts. *arXiv* (2023). URL https://arxiv.org/abs/2401.04088.

[56] AI, M. Meta Llama 3.1 8B. https://huggingface.co/meta-llama/Meta-Llama-3-8B (2024). Accessed: 2025-07-09.

[57] JSON Lines. JSON Lines Format. https://jsonlines.org (2023).

[58] vLLM Team. vLLM: A high-throughput and memory-efficient inference engine for LLMs. https://vllm.ai (2024).

[59] Colvin, S. and contributors. Pydantic. https://github.com/pydantic/pydantic (2024). Available at https://docs.pydantic.dev/.

[60] Lhoest, Q., Villanova, L., Jernite, Y., et al. Datasets: A community library for NLP datasets. https://github.com/huggingface/datasets (2024). Available at https://huggingface.co/docs/datasets.

[61] Paszke, A., Gross, S., Massa, F., et al. PyTorch: An open source machine learning framework. https://github.com/pytorch/pytorch (2024). Available at https://pytorch.org/.

[62] pandas development team, T. pandas: Python Data Analysis Library. https://github.com/pandas-dev/pandas (2024). Available at https://pandas.pydata.org/.

[63] Python Software Foundation. re — Regular expression operations (2024). Available at https://docs.python.org/3/library/re.html.

[64] xAI Team. xAI API Client for Grok and other models. https://github.com/xai-org/xai-api-client (2025). Available at https://x.ai.

[65] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. Scikit-learn: Machine Learning in Python. https://scikit-learn.org/ (2024). Available at https://github.com/scikit-learn/scikit-learn.

[66] Fawcett, T. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874 (2006).

[67] AI, M. Llama 2: Open Foundation and Chat Models. https://huggingface.co/meta-llama/Llama-2-70b-hf (2023). Accessed: 2025-07-09.

[68] Microsoft. Phi-3: Small Language Models with Big Impact. https://huggingface.co/microsoft/phi-2 (2024). Accessed: 2025-07-09.

[69] Rodriguez, D., Yang, I., Alamo, J. M. D., and Sadeh, N. Large Language Models: A New Approach for Privacy Policy Analysis at Scale (2024). URL https://arxiv.org/abs/2405.20900.

[70] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving Language Understanding by Generative Pre-Training. *OpenAI Blog* (2018). Available at https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[71] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186 (2019).

[72] Wolf, T., Debut, L., Sanh, V., and et al. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. ACL (2020). Introducing HuggingFace Transformers library.

[73] Shazeer, N., Mirhoseini, A., Maziarz, K., and et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *Proceedings of ICLR* (2017).

[74] Fields, J., Chovanec, K., and Madiraju, P. A Survey of Text Classification With Transformers: How Wide? How Large? How Long? How Accurate? How Expensive? How Safe? *IEEE Access*, PP:1–1 (2024).

[75] Sun, X., Li, X., Li, J., Wu, F., Guo, S., Zhang, T., and Wang, G. Text Classification via Large Language Models (2023).

[76] Wu, X.-K., Chen, M., Li, W., Wang, R., Lu, L., Liu, J., Hwang, K., Hao, Y., Pan, Y., Meng, Q., Huang, K., Hu, L., Guizani, M., Chao, N., Fortino, G., Lin, F., Tian, Y., and Niyato, D. LLM Fine-Tuning: Concepts, Opportunities, and Challenges. *Big Data and Cognitive Computing*, 9:87 (2025).

[77] Brown, T. B., Mann, B., Ryder, N., and et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems*, 33:1877–1901 (2020).

[78] Touvron, H., Lavril, T., Izacard, G., and et al. LLaMA: Open and Efficient Foundation Language Models. In *Proceedings of the International Conference on Machine Learning (ICML)* (2023).

[79] Jung, V. and Plas, L. Understanding the effects of language-specific class imbalance in multilingual fine-tuning. In Graham, Y. and Purver, M., editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2368–2376. Association for Computational Linguistics, St. Julian's, Malta (2024). URL https://aclanthology.org/2024.findings-eacl.157/.

[80] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., and Li, Y. LoRA: Low-Rank Adaptation of Large Language Models. In *Proceedings of the 2021 International Conference on Learning Representations* (2021).

[81] Dettmers, T., Pagnoni, A., Holtzman, A., and et al. QLoRA: Efficient Fine-Tuning of Quantized LLMs. In *Proceedings of NeurIPS* (2023).

[82] Wang, L., Chen, S., Jiang, L., Pan, S., Cai, R., Yang, S., and Yang, F. Parameter-efficient fine-tuning in large language models: a survey of methodologies. *Artificial Intelligence Review*, 58 (2025).

[83] Hugging Face. PEFT: Parameter-Efficient Fine-Tuning (2024). URL https://huggingface.co/docs/peft/en/index. Accessed: 2024.

[84] Huang, W., Zheng, X., Ma, X., Qin, H., Lv, C., Chen, H., Luo, J., Qi, X., Liu, X., and Magno, M. An empirical study of LLaMA3 quantization: from LLMs to MLLMs. *Visual Intelligence* (2024). URL https://doi.org/10.1007/s44267-024-00070-x.

[85] Huang, W., Zheng, X., Ma, X., Qin, H., Lv, C., Chen, H., Luo, J., Qi, X., Liu, X., and Magno, M. An empirical study of LLaMA3 quantization: from LLMs to MLLMs. *Visual Intelligence*, 2(36) (2024). URL https://link.springer.com/article/10.1007/s44267-024-00070-x.

thesis

A
Appendix

This appendix documents the prompt templates used for analyzing privacy policies. Each prompt is designed to elicit a binary answer (YES/NO) based on whether a specific category of personal information is shared with third parties.

A.0.1 template prompts tested with MAPP

```
Input: You are a helpful assistant trained to analyze privacy
policies. Always respond with YES or NO as instructed. The
following content between the double quotation marks is a privacy
policy. "Affiliates Press Contact Support Terms Privacy Site
Notice"
Definition of Third-Party: For this task, "third-party" refers to
any external entity (e.g., advertisers, analytics providers,
social media platforms, or other organizations) that is not the
primary operator of the website. Third-party data collection or
sharing occurs when the privacy policy explicitly states that a
data type is collected by or shared with such external entities.
Task:
Determine whether the privacy policy explicitly affirms that any
of the following personal data types are collected by or shared
with third parties. Only return " YES" if the policy clearly
states or directly implies that the specific data type is
collected by or shared with a third-party. If the data type is
not mentioned or the policy is unclear, return " NO." Do not
assume data collection or sharing based on generic terms like
"personal information" or references to other policies (e.g., app
privacy policy).
Data Types:
"Financial": "Financial information, such as credit/debit card
data, other payment information, credit scores, etc.",
```

"Health_genetic_or_biometric_data": "Information about a person's health, genome, or biometric markers.", "Contact_information": "Contact information, such as name, email address, phone number, street address, etc.", "Location": "Geo-location information (e.g., user's current location) regardless of granularity, i.e., could be exact location, ZIP code, city level.", "Demographic_data": "Demographic information, e.g., gender, sexual orientation, race, ethnicity, age, occupation, education, etc.", "Personal_identifier": "Identifiers that uniquely identify a person, e.g., SSN, ID card number, driver's license number, etc.", "User_online_activities": "The user's online activities on the firstparty websites/apps or other (thirdparty) websitesapps, e.g., user profiles, pages visited, time spent on pages, general user behavior online, etc.", "Social_media_data": "User profile and data from a social media websiteapp or other third-party service to which the user gave the First_Party access, e.g., by connecting with Facebook, Twitter, or other services. Exchanged data may include user profile, photos, comments, friends, etc.", "IP_address_and_device_IDs": "Permanent (e.g., device IDs, MAC address) or temporary (e.g., IP address) identifiers needed to establish a connection for the current browsing session.", "Cookies_and_tracking_elements": "Identifiers locally stored on the user's device by the companyorganization or third parties, including cookies, beacons, or similar that are commonly used to identify users uniquely but are not essential to establish a connection with the user's device or to provide a service.", "Computer_information": "The type of operating system OS or web browser that the user uses, or similar computer or device information.", "Generic_personal_information": "No specific type of information is mentioned, but the policy talks about ṕersonal informationór ṕersonally identifiable informationín general.", "Political_religious_or_philosophical_belief": "Any data that describes political, religious, or philosophical beliefs of individuals.", "Other": "A specific type of information not covered by other values for this attribute." "Unspecified": "The type of information is not explicitly stated or unclear (e.g., refers to 'information' very generically).",
Output Format Instruction: Please format your answer as follows: Data: Answer where Data is the data type above, and Answer must be only YES or NO. Strictly follow the output format. Do not add anything extra in the response. "

Main Prompt

Mixtral

```
<s>[INST]
You are a privacy policy analysis assistant. Respond ONLY with
"YES" or "NO" to indicate whether the
privacy policy mentions third-party {Personal_Information_Type}
sharing.
Example 1:
Privacy Policy: "We may use third-party cookies to track your
behavior across websites."
Answer: {{ {Personal_Information_Type}: YES }}
Example 2:
Privacy Policy: "We only use internal analytics tools."
Answer: {{ {Personal_Information_Type}: NO }}
Now analyze the following:
Privacy Policy: {text}
[/INST]
```

Grok

```
Task: You are a helpful assistant, expert in analyzing this
privacy policy for third-party
{Personal_Information_Type} disclosures. Respond ONLY with "YES"
or "NO".
Analysis criteria:
1. {Personal_Information_Type} includes cookies, web beacons,
pixel tags, etc.
2. Only mark as YES if the policy clearly states that data is
shared with third parties
3. If the policy is unclear or doesn't explicitly mention
sharing, mark as NO
4. If the policy says "May be", treat it as YES
5. Be very strict in your analysis { require clear evidence
Example 1:
Privacy policy: "...may use technology such as cookies... when
they advertise... which may send these advertisers
information..."
Answer: {{ {Personal_Information_Type}: YES }}
Example 2:
```

```
Privacy policy: "Cookies help us make our Service and your visit
better..."
Answer: {{ {Personal_Information_Type}: NO }}
```

LLaMA

```
Task: You are a helpful assistant, expert in analyzing this
privacy policy for third-party
{Personal_Information_Type} disclosures. Respond ONLY with "YES"
or "NO".
Analysis criteria:
1. {Personal_Information_Type} includes cookies, web beacons,
pixel tags, or similar technologies
2. Only mark as YES if the policy clearly states that data is
shared with third parties
3. If the policy is unclear or doesn't explicitly mention
sharing, mark as NO
4. If the policy says "May be", treat it as YES
5. Be very strict in your analysis { require clear evidence
Now analyze the following privacy:
"{text}"
```

Gemini

```
Task: You are a helpful assistant, expert in analyzing this
privacy policy for third-party
{Personal_Information_Type} information disclosures. Respond ONLY
with "YES" or "NO".
Analysis criteria:
1. Only mark as YES if the policy clearly states that
{Personal_Information_Type} is shared with third parties
2. If the policy is unclear or doesn't explicitly mention
sharing, mark as NO
3. Be very strict in your analysis { require clear evidence
4. If the policy says "May be", treat it as YES
Example 1:
Privacy policy: "...may use technology such as cookies... which
may send these advertisers information"
Answer: {{ {Personal_Information_Type}: YES }}
```

```
Example 2:
Privacy policy: "We use cookies to improve our service
experience."
Answer: {{ {Personal_Information_Type}: NO }}
```

B

OPP_115 Experimentation - Supplementary Figures

This appendix contains detailed ablation study figures for each model evaluated in the OPP_115 experimentation. These figures provide comprehensive visual analysis of how different prompt engineering techniques affect model performance across various personal information types.

B.1 DeepSeek-R1-Distill-Qwen-32B Ablation Studies

B.1.1 Temperature Effect Analysis

The graph shows that the model performs the best at temperature 0 for most of the personal information types. However, in some cases like Financial and others, the model performed better at a temperature of 0.2. In these cases, model benefited from its creativity.

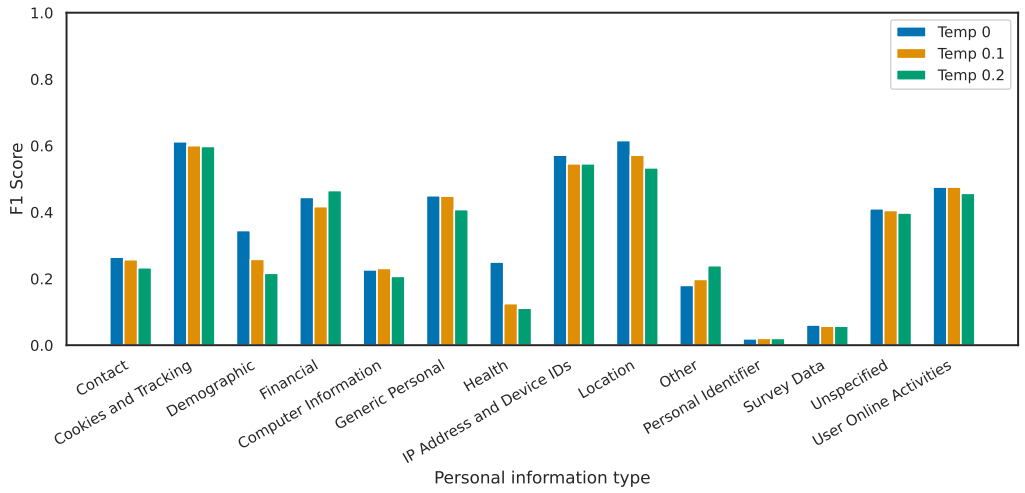

Figure B.1: DeepSeek: F1 score for personal information type under different temperature settings.

B.1.2 Few-Shot Prompting Analysis

The graph shows that DeepSeek's performance becomes better and better as it gets more few-shot instances for most categories of personal information. This improvement happens because the examples show the annotation task in action, which helps the model better comprehend the exact patterns and rules for finding different sorts of information in privacy policy text. Few-shot prompting makes it possible to learn in context, which is especially useful for this type of classification assignment.

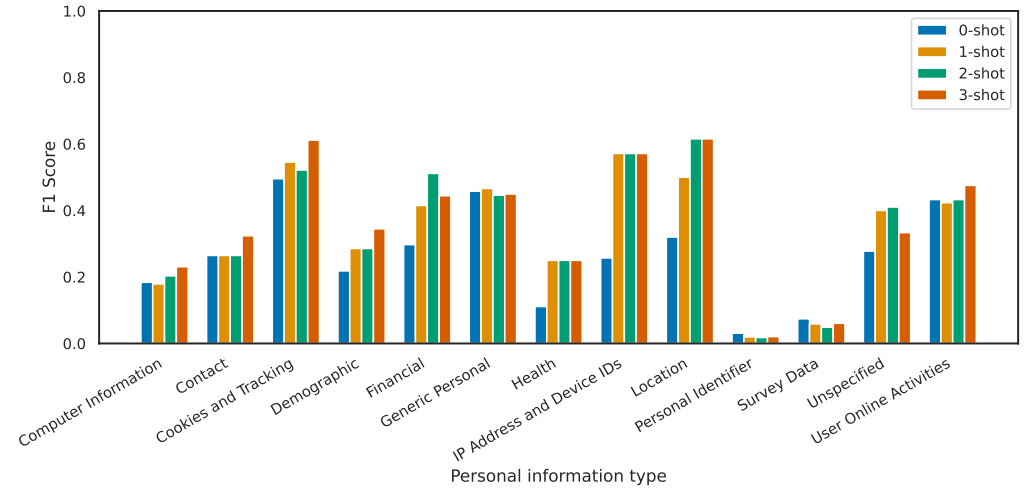

Figure B.2: DeepSeek: F1 score impact with 0-shot to 3-shot prompting.

B.1.3 Third-Party Definition Effect

This plot explains how DeepSeek reacts when provided with the contextual information in the prompt. In the given graph, we tried to provide the model with the third-party definition. And in another test, we removed the third-party definition from the prompt. The effect of removing it on F1 score is higher as compared to the effect of adding it, which are rare cases. Adding does increase the effect, but for just 2 or 3 information types. but the gain we got after we removed it overpowers its existence in the prompt. One explanation of these results could be that adding more context, in our case adding third-party definitions in the prompt, may restrict the model from using its own reasoning and might cause conflict with its own pre training data.

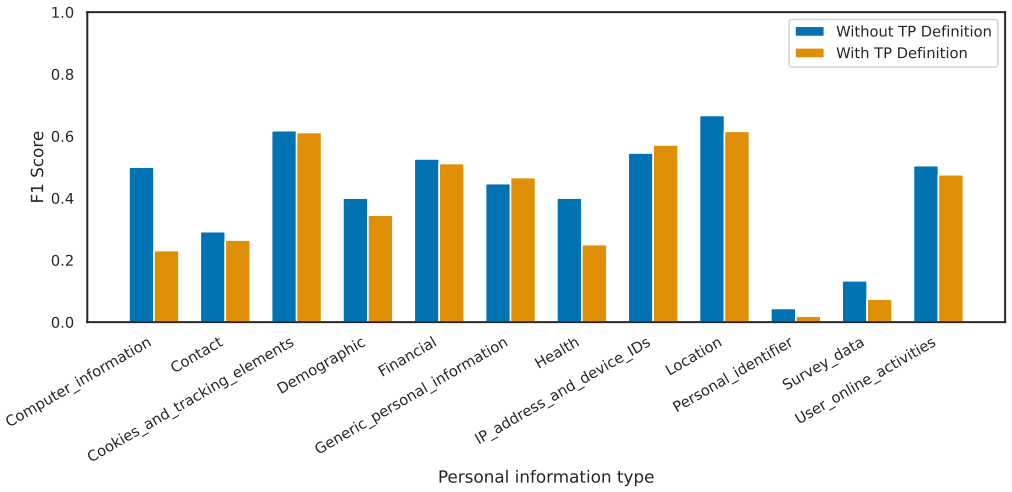

Figure B.3: DeepSeek: Effect of third-party definition on F1 score per attribute.

B.2 Mixtral 8x7B Ablation Studies

B.2.1 Temperature Effect Analysis

The graph shows that for most forms of personal information, Mixtral works best at temperature 0. At temperature 0.1, the outcomes are mixed, while at temperature 0.2, the results are frequently the worst. Temperature 0 works best for distinguishing information kinds like contact, cookies and tracking, demographic, and financial. But for a few specific categories, including health and user online activities, slightly higher temperature values seem to help a little, which means that a little bit of model originality can sometimes help with some categorization jobs.

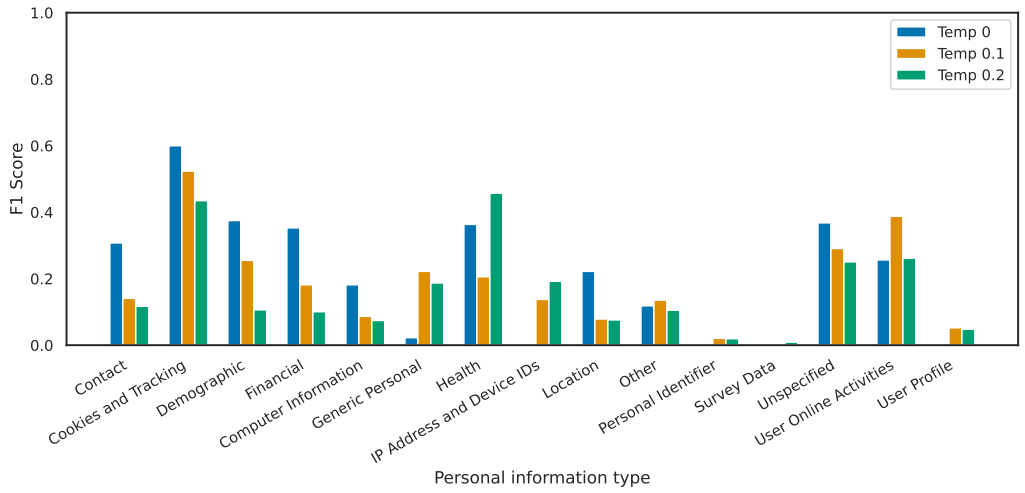

Figure B.4: Mixtral: F1 score across temperature settings.

B.2.2 Few-Shot Prompting Analysis

With few-shot examples, Mixtral doesn't always act the same way. 1-shot prompting usually works best (as seen in the Demographic, Financial, and Health categories), but this isn't always the case. For example, 2-shot instances work better in the Contact and Cookies/-Tracking categories. This means that Mixtral's performance is very sensitive to the number and quality of examples given, not just that it gets better with more examples like other models.

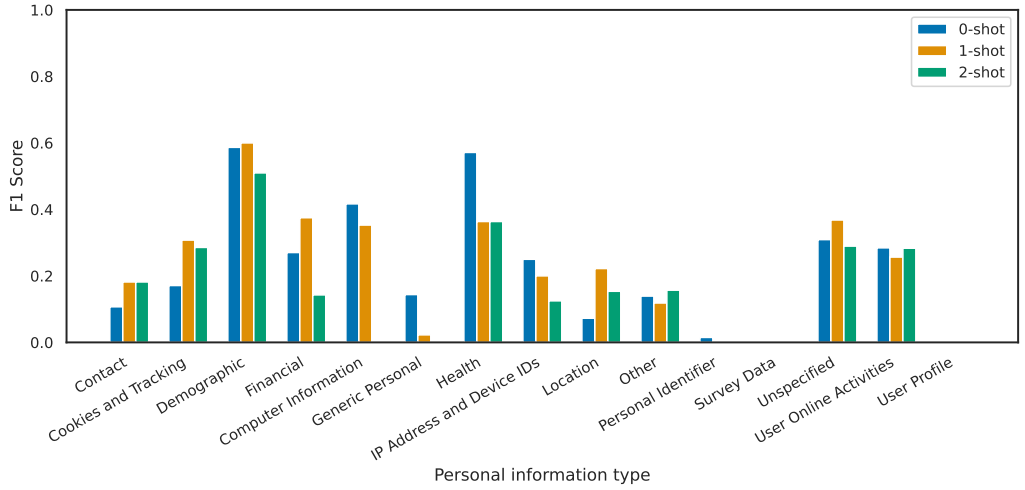

Figure B.5: Mixtral: Few-shot prompting impact on F1 score.

B.2.3 Definition Effect Analysis

The definitions in the prompt include definitions of information types. The results reveal that adding these definitions to Mixtral has both good and bad consequences. Providing definitions in categories like Contact, Cookies and Tracking, Demographic, and Location made the model work better, which suggests that having explicit contextual knowledge helps it interpret certain types of information better. But for categories like Health, Generic Personal, and User Online Activities, taking away definitions (orange bars) made the model work better. This suggests that the extra definitional context may have messed up the model's pre-trained grasp of these ideas. This means that Mixtral's response to definitional prompts depends a lot on the category. Some sorts of information do better with clear instructions, while others do better when they rely on the model's own understanding.

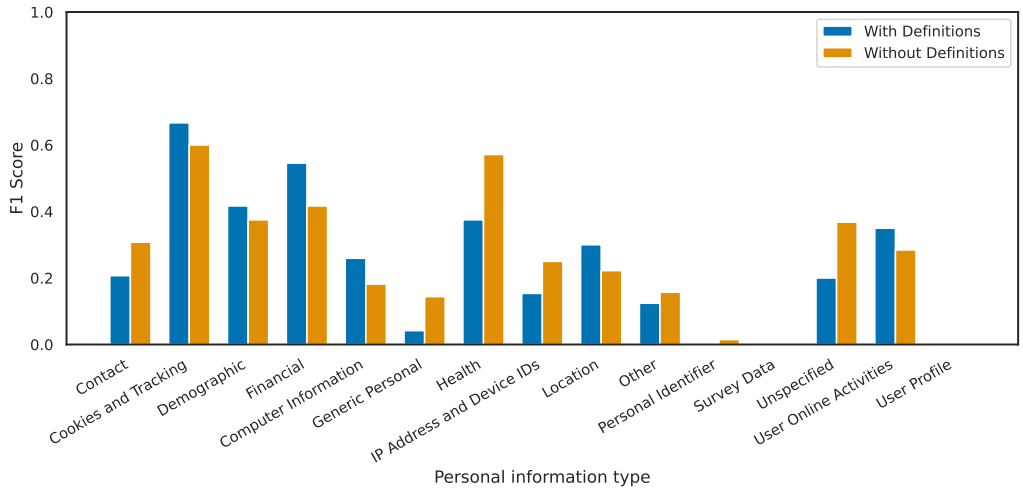

Figure B.6: Mixtral: Effect of definitions on classification performance.

B.3 Gemini Ablation Studies

B.3.1 Temperature Effect Analysis

Gemini shows varied performance across temperature settings, with temperature 0.1 frequently outperforming the deterministic setting (temperature 0) in categories like demographic, location, and generic personal information. However, temperature 0 still performs best or competitively in several categories, including cookies and tracking, computer information, and user online activities. This suggests that Gemini benefits from minimal creativity (slight randomness) for certain information types. The optimal temperature appears to be information type dependent rather than consistently favoring higher/lower temperatures.

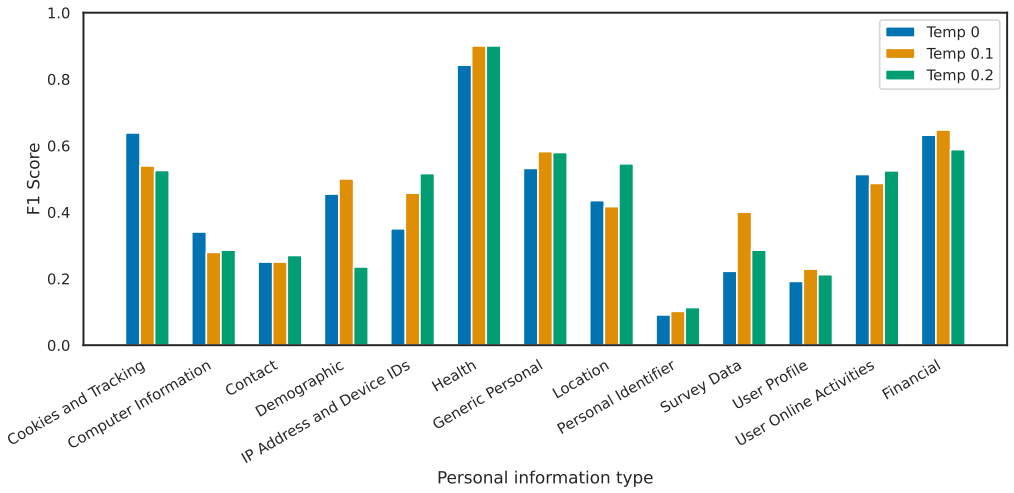

Figure B.7: Gemini: Temperature ablation effect on F1 score.

B.3.2 Few-Shot Prompting Analysis

Gemini's performance with few-shot samples is also unpredictable, just like with temperature settings. It seems that the model is very sensitive to the instances that were given. In other cases, like with IP addresses and Device IDs, it works much better with few-shot samples. But in some circumstances, like Cookies and Tracking, fewer examples work better, thus the examples actually hurt performance. Gemini seems to care more about the quality of examples than the number of them. Good examples help the model comprehend the task better, while bad examples might confuse it and mess up what it already knows. This finding shows that when employing few-shot prompting with Gemini, it's more necessary to choose good examples than to just give more examples.

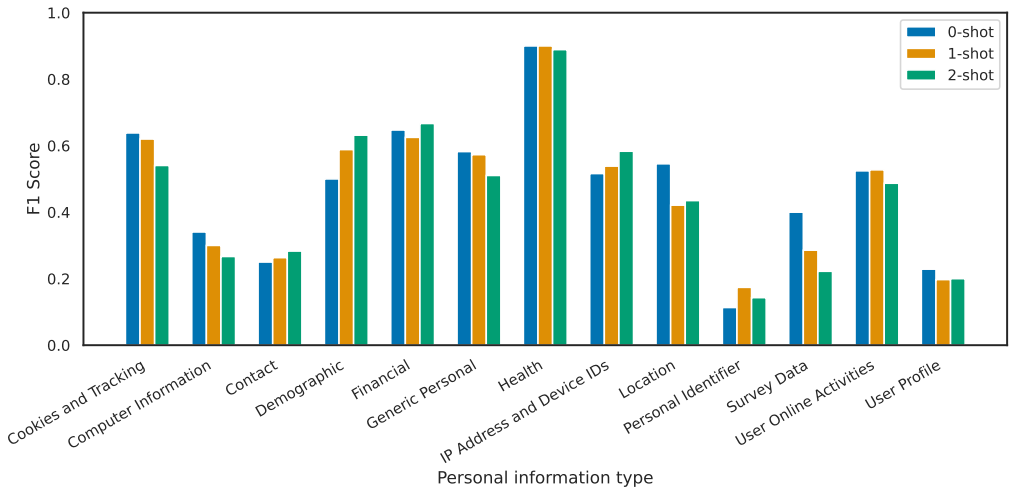

Figure B.8: Gemini: Performance change from 0-shot to 2-shot prompts.

B.4 Grok Ablation Studies

B.4.1 Few-Shot Prompting Analysis

The graph shows that Grok's performance changes depending on the few-shot configuration, and there is no clear trend showing that more examples lead to better performance. Some categories, including Cookies and Tracking, Health, and IP Address and Device IDs, do best with 1-shot prompts. Other categories, like Financial and User Online Activities, do better with 2-shot prompts. Some categories, including Computer Information and Contact, don't change much between setups. This suggests that some forms of information don't depend as much on the number of examples given. This inconsistent pattern shows that Grok's few-shot learning works best for some tasks and not others, rather than following a predictable scaling relationship. This shows how important it is to test things out in real life to figure out how many examples are best for different types of information classification tasks.

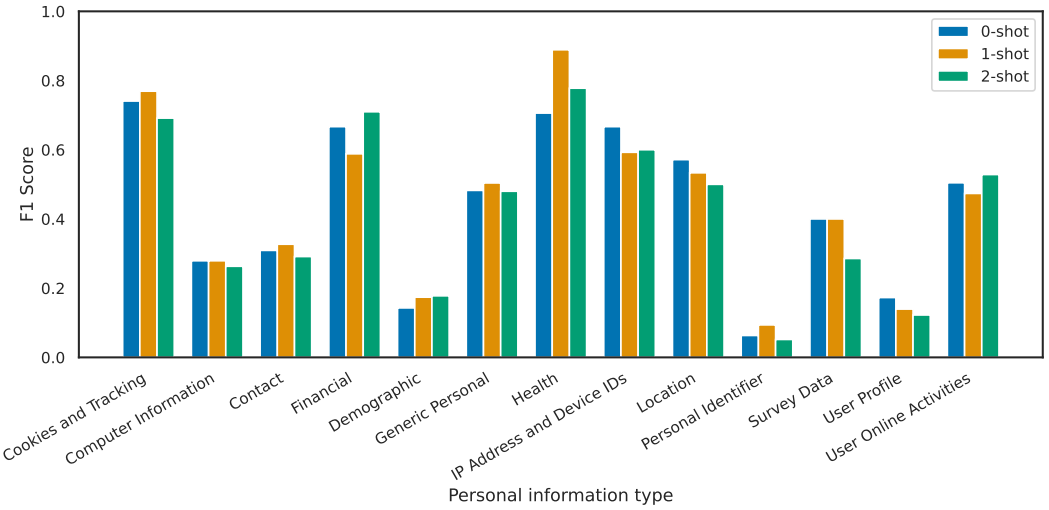

Figure B.9: Grok: Effect of few-shot examples on F1 score.

B.5 LLaMA Ablation Studies

B.5.1 Temperature Effect Analysis

For most types of personal information, such as contact, cookies and tracking elements, health, and IP addresses and device IDs, LLaMA works best at temperature 0 (deterministic output). This pattern shows that the model works best when it makes consistent, targeted decisions instead of adding randomness to the inference process. The categories "Generic Personal Information" and "User Online Activities" don't change much when the temperature changes, which suggests that these sorts of information may not be as affected by how creative the model is. The results show that for tasks that include classifying privacy policies, LLaMA works better with deterministic processing, which is in line with the need for accuracy in identifying information types.

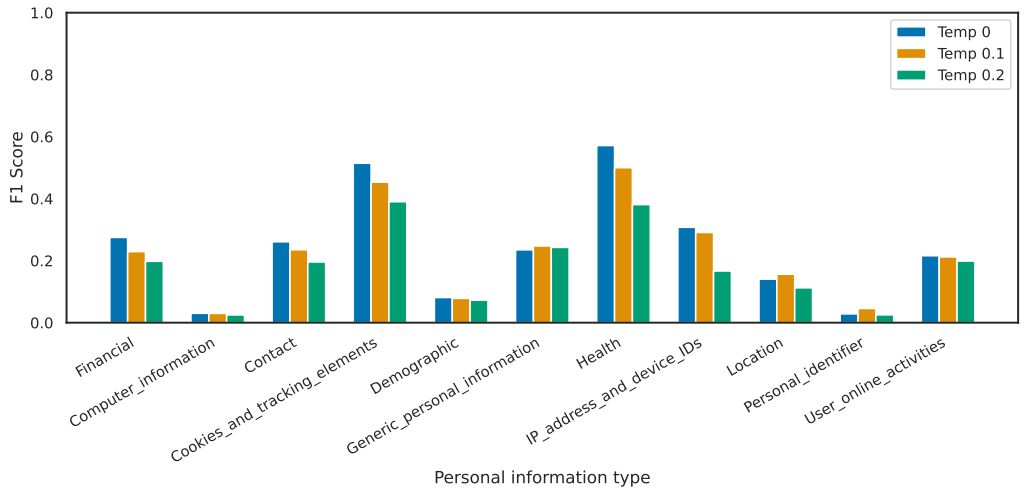

Figure B.10: LLaMA: F1 scores by attribute across temperature settings.

B.5.2 Few-Shot Prompting Analysis

LLaMA shows a range of responses to few-shot instances across different categories of information, and most of them are positive. With more examples, categories like Contact, Cookies and Tracking Elements, and User Online Activities are getting better. 1-shot and 2-shot do about the same. Some categories, like Health, do far better with 0-shot instances than with few-shot examples, while others, like Generic Personal Information, do about the same across both configurations. This trend shows that LLaMA's few-shot learning works better with some types of information than others. For example, some categories may benefit from examples, while others may be hurt by demonstrations that are not relevant or are hard to understand.

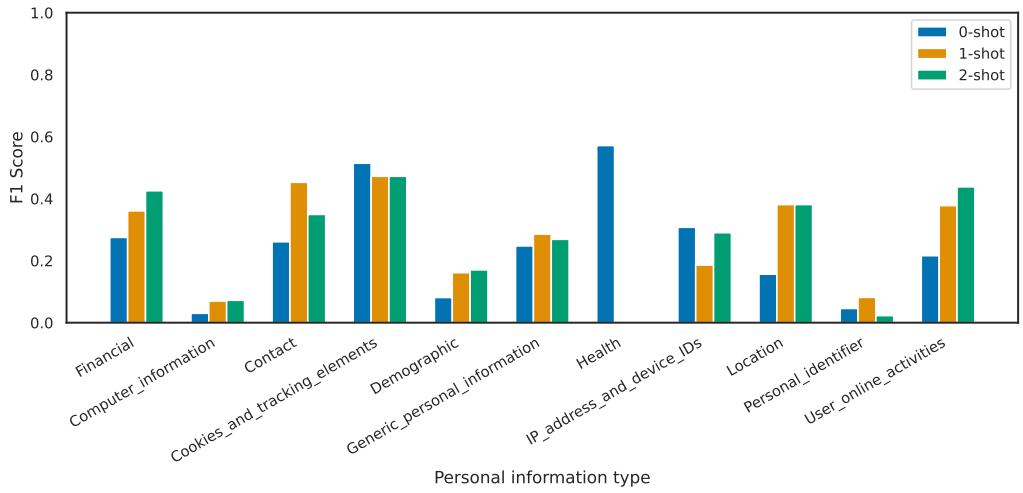

Figure B.11: LLaMA: Few-shot prompting ablation on attribute-wise performance.

B.5.3 Instruction Effect Analysis

The graph illustrates that LLaMA does better when it is given clear instructions for most types of personal information. Contact, Cookies and Tracking Elements, Health, and user online activities are only a few examples of categories that show big gains when given instructions. This shows that LLaMA needs systematic help to comprehend the classification task. But some groups, like Generic Personal Information and Personal Identifier, don't show much of a difference between the two situations. This suggests that the model's pre-trained information may be enough for these easier categorization tasks. Overall, the results show that giving LLaMA explicit instructions makes it better at annotating privacy policies, especially for information categories that are more complicated or unclear.

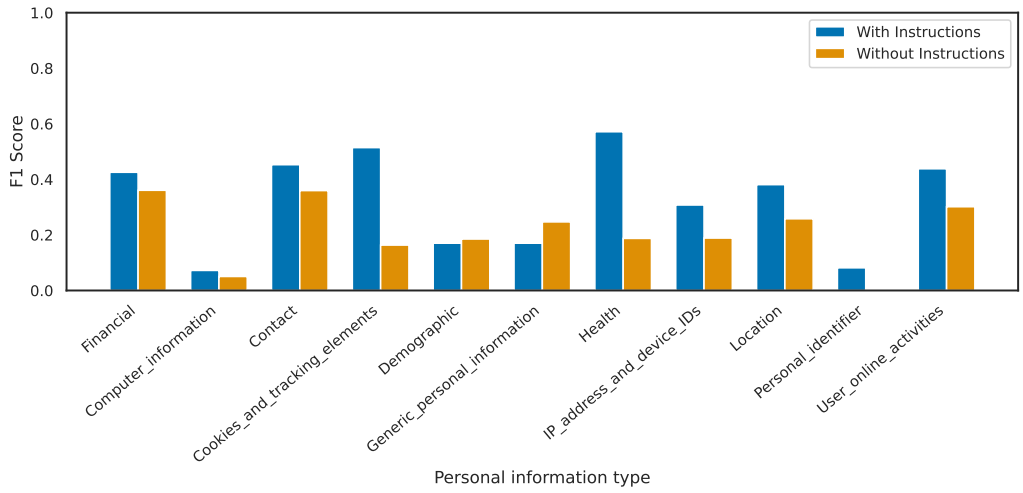

Figure B.12: LLaMA: Effect of instructions on F1 score across personal information types.

University of Basel

Faculty of Science

Declaration on Scientific Integrity

(including a Declaration on Plagiarism and Fraud)
Translation from German original

Title of Thesis:

Name Assessor: Prof. Isabel Wagner

Name Student: Maryem Fatima

Matriculation No.: 21-067-079

I attest with my signature that I have written this work independently and without outside help. I also attest that the information concerning the sources used in this work is true and complete in every respect. All sources that have been quoted or paraphrased have been marked accordingly.

Additionally, I affirm that any text passages written with the help of AI-supported technology are marked as such, including a reference to the AI-supported program used. This paper may be checked for plagiarism and use of AI-supported technology using the appropriate software. I understand that unethical conduct may lead to a grade of 1 or "fail" or expulsion from the study program.

Place, Date: 17 july, 2025 Student: Mafem

Will this work, or parts of it, be published?

No

Yes. With my signature I confirm that I agree to a publication of the work (print/digital) in the library, on the research database of the University of Basel and/or on the document server of the department. Likewise, I agree to the bibliographic reference in the catalog SLSP (Swiss Library Service Platform). (cross out as applicable)

Publication as of: _____

Place, Date: 17th july, 2025 Student: Mafem

Place, Date: _____ Assessor: _____

Please enclose a completed and signed copy of this declaration in your Bachelor's or Master's thesis.