Automatic Privacy Compliance Checks for Mobile Apps Using Natural Language Processing

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney

Bhanuka Malith Silva PINCHAHEWAGE
Networked Systems and Security (NSS) Research Lab
2026

Declaration of Authorship

I, Bhanuka PINCHAHEWAGE, declare that this thesis titled *'Automatic Privacy Compliance Checks for Mobile Apps Using Natural Language Processing'* and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Abstract

The rapid growth of the mobile app ecosystem has intensified concerns around how user data is collected, shared, and represented through privacy disclosures. While regulatory frameworks such as General Data Protection Regulation (GDPR) establish strong foundations for data protection and disclosure transparency, privacy compliance in app marketplaces relies heavily on developer self-reporting and user awareness. As a result, privacy information, whether in detailed policy documents or in summarised forms, often fail to accurately portray data practices. This thesis explores how recent advances in natural language processing can be leveraged to enable automated, scalable privacy compliance checks in the Google Play Store. We identify key factors limiting the transparency and usability of privacy policies and propose enhanced parsing and structuring methodologies that improve comprehension and support regulatory oversight.

Existing encoder-based models for privacy policy related classification often yield accurate predictions but lack interpretability, while decoder-based large language models (LLMs) can provide explanations yet suffer from hallucinations and lack verifiability. To address this gap, first chapter introduces an *entailment-driven LLM framework* that couples generative reasoning and re-evaluation strategies with embedding-based verification to ensure factual and logically consistent outputs. The proposed approach enhances both the reliability and explainability of privacy policy paragraph classification compared to traditional methods.

Next, we investigate the privacy compliance landscape in Google Play Store via *PrivPRISM, a novel language-modelling framework* that leverages both encoder and decoder architectures for fine-grained extraction and evaluation of privacy statements. By cross-analysing privacy policy text, Google Play's data summary labels, and evidence from app installation files, PrivPRISM performs multi-layered consistency checks to detect mismatches in declared versus implemented data practices. Empirical studies show that 53% of analysed apps exhibit discrepancies in their disclosures re-emphasising necessity of evidence-driven compliance auditing at scale.

Despite ongoing efforts to improve transparency, privacy policies remain complex legal documents that are difficult to interpret in practice. Existing automated analysis approaches often treat policies as flat text, overlooking their internal structure and hierarchical organisation, which is critical for accurately interpreting data disclosures. In the final chapter, we introduce *PrivSTRUCT, a systematic encoder–decoder framework* that leverages developer-defined structural cues to disentangle complex privacy disclosures by linking data items to their stated purposes. Our findings reveal a persistent transparency gap in which broadly defined purpose disclosures obscure sensitive first- and third-party data practices in mobile apps.

Acknowledgements

First and foremost, I thank my primary supervisor, Associate Professor Suranga Seneviratne from the University of Sydney, for his expert guidance and encouragement. I am equally grateful to Dr. Anirban Mahanti (University of Sydney) and Professor Aruna Seneviratne (University of New South Wales) for their continuous guidance and scholarly advice throughout my candidature. My thanks also go to my co-authors and peers at the Networked Systems and Security (NSS) Research Lab for their support.

Funding Acknowledgements

This research was supported by the Australian Research Council (ARC) Discovery Project (DP220102520), titled *"Betrayed by Apps: Automated, Scalable Detection of Mobile App Malpractices."*

I would further acknowledge the support provided by the University of Sydney Tuition Fee Scholarship and Faculty of Engineering Research Stipend Scholarship during my candidature period.

List of Publications

Publications relevant to this thesis;

1. **Silva, B.**, Denipitiyage, D., Seneviratne, S., Mahanti, A., and Seneviratne, A. (2024). Entailment-Driven Privacy Policy Classification with LLMs. in: IEEE Conference on Building a Secure and Empowered Cyberspace (BuildSec).

2. **Silva, B.**, Denipitiyage, D., Mahanti, A., Seneviratne, A., and Seneviratne, S. (2026). PrivPRISM: Automatically Detecting Discrepancies Between Google Play Data Safety Declarations and Developer Privacy Policies. *Under review in*: 26th Privacy Enhancing Technologies Symposium

3. **Silva, B.**, Mahanti, A., Seneviratne, A., and Seneviratne, S. (2026). PrivSTRUCT: Untangling Data Purpose Compliance of Privacy Policies in Google Play Store. *Under review in*: 21st ACM ASIA Conference on Computer and Communications Security (ASIACCS)

Publications during candidature but not relevant to this thesis;

4. Denipitiyage, D., **Silva, B.**, Gunathilaka, K., Seneviratne, S., Mahanti, A., Seneviratne, A, and Chawla, S. (2025). Detecting and Characterising Mobile App Metamorphosis in Google Play Store. In: IEEE Transactions on Mobile Computing (TMC)

5. Denipitiyage, D., **Silva, B.**, Seneviratne, S., Seneviratne, A, and Chawla, S. (2025). A Vision-Language Approach with Cross Attention for Detecting Content Rating Malpractices in Android Applications. In: 24th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (Trustcom)

6. Ginige, Y., **Silva, B.**, Dahanayaka, T., and Seneviratne, S, (2025). TrafficLLM: An LLM Approach for Open-Set Encrypted Traffic Classification. In: Elsevier Computer Networks)

7. Karunanayake, N., **Silva, B.**, Ginige, Y., Seneviratne, S., and Chawla, S. (2025). Quantifying and Exploiting Adversarial Vulnerability: Gradient-Based Input Pre-Filtering for Enhanced Performance in Black-Box Attacks. In: ACM Transactions on Privacy and Security (TOPS).

Authorship Attribution Statement

We present the authorship attribution for the published works from the previous section (List of Publications) included in the thesis chapters.

- Chapter 3 of this thesis has been published as [1]. I designed the framework, training/testing pipelines, benchmarked $> 80\%$ existing models, analysed the results and drafted the manuscript.
- Chapter 4 of this thesis has been submitted as [2]. I designed the PrivPRISM framework, data pre-processing steps, training/testing/at-scale inferencing pipelines, benchmarked PrivPRISM, analysed the results and drafted the manuscript.
- Chapter 5 of this thesis has been submitted as [3]. I designed the PrivSTRUCT framework, data pre-processing steps, training/testing/at-scale inferencing pipelines, benchmarked PrivSTRUCT, analysed the results and drafted the manuscript.

I certify that the authorship attribution statements provided above are correct, and I have obtained permission from the other authors to include the published materials. As the lead author, and by convention in my research field, I have made the primary contribution to the publications. Additionally, I am the corresponding author for the publications included in this thesis.

Bhanuka PINCHAHEWAGE

As the supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Dr. Suranga SENEVIRATNE

Use of Generative AI

During the preparation of this thesis, *Meta-Llama* models (locally hosted via University of Sydney computing resources) under the *Community Licence Agreement* and *GPT* models (accessed via API) in compliance with the *OpenAI Sharing and Publication Policy* were employed as integral components of the research design. Full details of their implementation are provided in the methodology section of each technical chapter.

I confirm that in limited instances where text was modified by generative AI (OpenAI ChatGPT) for grammatical checks / minor refinements, the content was reviewed for possible errors, inaccuracies, and bias. I take full responsibility for the submitted thesis and confirms the work is my own and has used generative AI in accordance with University guidelines and policies.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

APK	Android Package Kit
APP	Australian Privacy Principles
CCPA	California Consumer Privacy Act
DPO	Direct Preference Optimisation
DS	Data Safety
GDPR	General Data Protection Regulation
GenAI	Generative Artificial Intelligence
IQR	Inter Quartile Range
LIME	Local Interpretable Model-Agnostic Explanations
LLM	Large Language Model
LoRA	Low Rank Adaptation
NER	Named Entity Recognition
NLP	Natural Language Processing
NLU	Natural Language Understanding
PEFT	Parameter Efficient Fine Tuning
PP	Privacy Policy
RAG	Retrieval Augmented Generation

Chapter 1

Introduction

Mobile applications (apps) have become an integral part of everyday life, supporting activities ranging from entertainment and education to productivity, health and well-being, navigation, and financial services. According to annual statistics published by the Global System for Mobile Communications Association (GSMA), the number of unique mobile subscribers increased from 5.2 billion in 2020 to 5.8 billion in 2025, and is projected to reach 6.2 billion, approximately 76% of the global population by 2030 [1, 2]. Over the same period, smartphone connections increased by nearly 15 percentage points, reaching 80% in 2025, and are expected to account for 91% of all mobile connections by 2030. Already a multi-trillion-dollar industry (USD 6.5 trillion, corresponding to approximately 5.8% of global GDP), the mobile ecosystem continues to expand, with an increasing number of services being designed and delivered primarily through mobile applications.

This ubiquity is evident across diverse sectors. In Australia, for example, the most popular financial application, the *CommBank App* has grown its user base from 6.1 million in 2020 to over 9 million users by 2025, representing nearly half of the bank's total customers [3, 4]. Similarly, *Uber*, an entirely mobile-first service, reached 7.4 million Australian users in 2025, surpassing traditional taxi services by nearly 3 million users, despite being established locally only in 2012 [5]. The quick-service restaurant industry has likewise embraced mobile applications, driven by order accuracy, contactless payments, and loyalty programmes; recent statistics indicate that 73% of Australian diners use mobile apps for quick-service ordering and pickup [6]. With an average user interacting with approximately nine apps per day, or thirty apps per month [7], concerns surrounding data privacy have become unavoidable. Modern smartphones operate as continuous, multi-modal sensing platforms, capturing fine-grained signals leveraging the device's rich suite of sensors and background processes related to user behaviour, mobility, communication, and situational context. Mobile apps are deeply embedded within this data-rich environment and routinely collect, process, and transmit sensitive personal information including behavioural, financial, and location data, not only for core functionality, but also for analytics, profiling, and advertising purposes.

Privacy policies (PPs) have long served as the primary legal instrument through which service providers disclose their *data practices*, including what information is collected or shared, how it is processed, and for which purposes it is used. Despite their central role, privacy policies are widely regarded as lengthy, complex, and difficult for end-users to understand [8, 9, 10]. A recent study reports that 94% of Australians do not read all privacy policies that apply to them [11], a pattern that is consistently observed at a global scale [12, 13]. Moreover, the writing style and presentation of these documents frequently render them inaccessible [14], resulting in consent that is often uninformed or superficial [15], where users agree to terms they do not understand.

Recent regulatory initiatives, most notably the European Union's General Data Protection Regulation (GDPR), have sought to address these challenges by mandating privacy disclosures and encouraging greater transparency. While such efforts have yielded measurable improvements such as a 4.9% increase in the availability of privacy policies [16], they have also produced unintended consequences. Longitudinal analyses indicate that privacy policies have grown substantially longer, increasing by approximately 25% globally [17]. Median word counts now range between 1,500 - 2,500 words [18, 19], making them prohibitively time-consuming to read [9, 18]. As a result, users often disengage from privacy disclosures altogether [20], reinforcing a **"transparency paradox"** in which *individuals are expected to make informed decisions about their personal data using information that is largely inaccessible*.

While data practices enable essential app functionality, personalisation, analytics, and targeted advertising, *ensuring that the data practices comply with the privacy policy disclosures, regional laws and market operator regulations*, i.e., **"privacy compliance"**, remains a significant challenge. Compliance violations are reported regularly [21, 22, 23] across major platforms, including the Google Play Store [24], Apple's App Store [25, 26], and third-party marketplaces [27]. A particularly severe privacy violation occurs, for example, when a policy states that location data is not collected, yet technical analysis reveals otherwise. Despite the formulation of new legislation and guidelines, there is currently no straightforward and automated technological solution to verify adherence to these rules. The efforts of consumer protection agencies, privacy advocates, and market operators are currently limited by scalability issues in the face of an ever-growing app market.

To mitigate these transparency limitations, particularly to improve end-user understanding of developer intended data practices, platform operators have introduced new mechanisms. For instance, Google and Apple now require developers to submit summarised labels known as "data safety" or "app privacy" declarations. An example data safety declaration highlighting the collection of approximate location for analytics, advertising and other purposes is depicted in Figure 1.1. These labels are intended to provide simplified, high-level descriptions

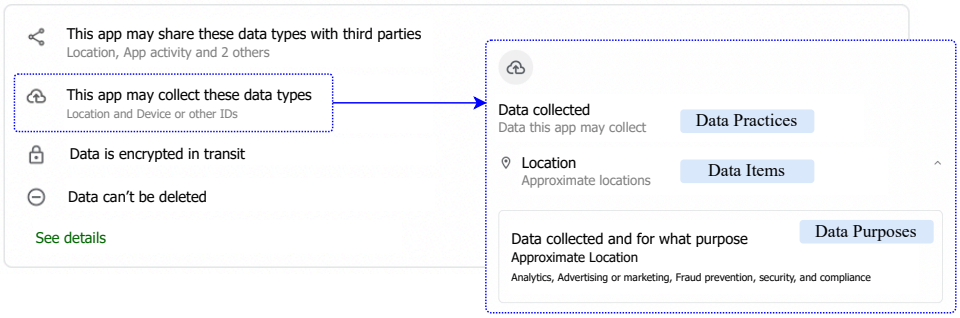

Figure 1.1: A representative Android Data Safety declaration. Highlighted blue colour labels are fine-granular details disclosed to end-users, linking the general data practice (first party collection) to specific data items (location) with attributes ('approximate' location) and their intended purposes (analytics, etc.).

of data handling practices to complement the full privacy policy. However, recent investigations have shown that discrepancies between these simplified labels and the full legal policies are widespread, often revealing outright contradictions [28, 29, 30, 31, 32, 33, 34]. Such discrepancies undermine user trust and weaken the effectiveness of regulatory oversight, further motivating the need for scalable, automated methods to analyse and verify privacy disclosures in mobile app ecosystems.

Determining whether an app complies with privacy regulations requires inferring concrete data practices from natural language policy statements and assessing their consistency with observed or declared behaviour. The broader natural language processing (NLP) landscape has recently undergone a paradigm shift. New generations of Large Language Models (LLMs), such as GPT and LLaMA exhibit strong capabilities in natural language understanding (NLU) and inference, appearing promising for interpreting complex regulatory documents. Yet, these models are not a silver bullet; they suffer from well-known pitfalls, including hallucinations, reasoning errors, and brittle behaviour when processing the precise styles of legal prose. As a result, naïvely applying vanilla LLMs to privacy disclosures is insufficient for reliable compliance assessment or user-facing explainability.

Privacy Policy Paragraph:	"Through cookies, pixels, beacons, log files and other technologies, we may collect information about how you use the Site and the Services. For example, we may determine through an IP address that a particular computer or device is located in New York City, and we may use this information to deliver advertisements promoting New York City-based businesses."
GPT-4 Response:	"Third Party Sharing: Although not explicitly stated if third parties are involved, the mention of "advertisements promoting New York City-based businesses" suggests that there might be sharing with or collection by third parties (advertisers in this case) to deliver targeted advertisements."

Figure 1.2: An example of incorrect identification by GPT-4.

Figure 1.2 depicts a clear example of this limitation using a sample from the legal expert annotated OPP-115 dataset (a formal introduction to this dataset will be provided in Section

2.5.1.1). When prompted with a privacy policy paragraph and a list of detailed data practices with the task to identify the most suitable data practice class and depict why that class was selected, GPT-4 fails to capture the nuance of the text despite the policy explicitly describing a first-party collection for targeted advertising (noting that no evidence is present that IP addresses are disclosed to third parties), the model incorrectly classifies it as third-party sharing. Furthermore, we highlight that this error occurred within a small context window, whereas providing an entire privacy policy text to generative AI models often exacerbates these issues, triggering "context forgetting" or the "lost-in-the-middle" phenomenon, where attention is not distributed evenly, leading to further incorrect interpretations.

Taken together, these observations illustrate a broader challenge in analysing privacy policies using automated methods. Even when policies explicitly describe specific data practices, accurately interpreting such disclosures at scale remains difficult, particularly as policy length and complexity increase. This gap motivates the need for more robust approaches to privacy policy analysis that can operate reliably in real-world settings. We next situate this challenge within the regulatory landscape governing mobile app privacy disclosures.

1.1 Mobile App Privacy Policies and the Regulatory Landscape

To understand why privacy compliance remains difficult to automate even in the presence of increasingly powerful NLP models, it is necessary to examine how regulatory interventions and platform governance have historically shaped privacy disclosures in mobile app ecosystems. Figure 1.3 presents a two-decade timeline illustrating how regulatory pressures, platform policies, and developer responses have co-evolved. Rather than progressing linearly toward greater transparency, this history reveals a recurring pattern: regulatory intervention is followed by adaptive developer behaviour that often preserves formal compliance while undermining practical clarity.

Minimal compliance phase: The period prior to 2012 represents the formative years of large-scale mobile app ecosystems where privacy regulation was fragmented, weakly enforced, and largely disconnected from mobile-specific data practices. although the infrastructure for the app ecosystems was established with the launch of first iPhone (2007), first Android smartphone (HTC Dream - 2008), and their respective marketplaces (App Store and Android Market - 2008), privacy disclosures remained voluntary. Discussions regarding privacy policies rarely extended to mobile applications, and academic research was similarly sparse, focusing primarily on the emerging risks associated with GPS sensors embedded to mobile smart-phones and respective location services [35, 36].

Reactive compliance phase: The years following 2012 mark a turning point driven by the

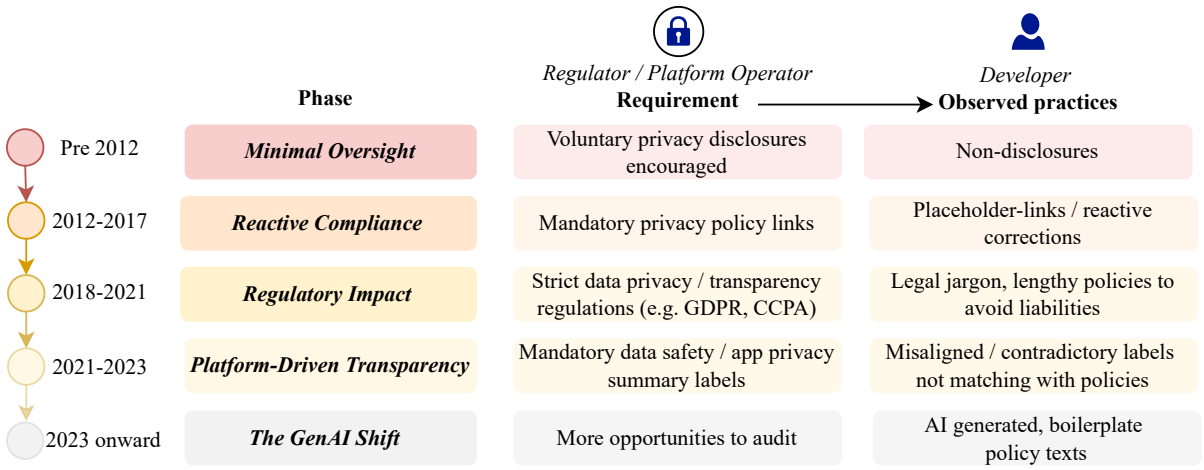

Figure 1.3: A timeline constituting notable regulatory efforts for the past two decades, contrasted with key developer habits negatively impacting privacy compliance. Warmer colours represent comparatively less regulatory efforts, affecting more negatively towards end-user privacy.

rapid expansion of smartphones, app-based services, and data-driven business models. The rebranding of Android Market as the Google Play Store in 2012 coincided with substantial growth in app distribution, monetisation, and the consolidation of centralised platform governance. As mobile applications became primary interfaces for digital services, regulatory attention to their privacy disclosures increased.

In the same year, California Attorney General reached an agreement with leading operators on a set of principles requiring mobile apps to include a publicly accessible privacy policy where applicable law requires (e.g., in California) [37]. Enforcement actions began to emerge, often triggered by public complaints or high-profile incidents, but they remained largely reactive rather than systematic. Developers typically addressed privacy concerns only after scrutiny, reinforcing a compliance culture centred on ex-post correction.

Empirical evidence from this period further illustrates the limited maturity of privacy practices. Many developers relied on placeholder or generic URLs, and Sunyaev et al. reported that only 30.5% of mobile health applications disclosed a privacy policy, with approximately two-thirds of those policies not being specifically tailored to the app itself [38]. Consequently, privacy policies functioned primarily as legal formalities rather than substantive transparency mechanisms, establishing early norms that prioritised legal sufficiency over user comprehension—norms that continue to shape developer behaviour in mobile app ecosystems today.

Regulatory impact phase: The period between 2018 and 2021 represents a structural shift driven by comprehensive regulatory frameworks such as the EU GDPR, alongside strengthened privacy laws in other jurisdictions including the United States and Australia. GDPR introduced

explicit requirements for transparency, purpose limitation, and accountability, while substantially increasing the legal consequences of non-compliance. However, these stronger obligations also incentivised risk-averse disclosure strategies. Privacy policies expanded rapidly, incorporating exhaustive lists of data categories and purposes to hedge against legal uncertainty. While availability of policies and formal transparency increased, practical comprehensibility often declined, creating a widening gap between regulatory intent and user understanding [16, 17].

Platform-driven transparency phase: Recognising the limitations of traditional privacy policies, platform operators such as Google and Apple introduced standardised, user-facing summaries of data practices: iOS App privacy disclosures in 2021 and Android Data Safety labels in 2022. This period marks a shift from primarily regulatory enforcement to platform-mediated transparency mechanisms embedded directly in app markets. These labels were designed to offer concise, comparable insights at the point of installation. However, they rely on self-declaration by developers and are not intended to replace full privacy policies. This dual-disclosure requirement introduces a new compliance challenge: developers are now expected to maintain consistency between two different representations of the same practices. In practice, this led to widespread misalignments and contradictions [28, 29]. This platform-driven model improved visibility but also exposed structural weaknesses in existing disclosure practices which we further discuss in literature review Sub Section 2.1.

The GenAI shift: The most recent phase is defined by the emergence of generative AI (GenAI), which introduces a fundamentally different dynamic. Unlike earlier regulatory or platform interventions, GenAI affects both the production and evaluation of privacy disclosures. On the developer side, it dramatically lowers the cost of generating, rewriting, and localising privacy policies thereby increasing the prevalence of plausible but inaccurate or overly generic or non-personalised disclosures [39, 40]. On the regulator and auditor side, GenAI enables large-scale, automated analysis of privacy texts, offering new possibilities for identifying inconsistencies, omissions, and structural ambiguities.

This bidirectional impact intensifies the long-standing *tug-of-war* between disclosure and enforcement. As disclosure generation becomes easier and faster, ensuring alignment between stated and actual practices becomes increasingly critical. Addressing this challenge requires analytical frameworks that move beyond surface-level text processing and instead support structured, reasoning-driven analysis of privacy text motivating the approach explored in the remainder of this thesis.

1.2 Accessibility of Mobile App Privacy Policies

Beyond the semantic content of privacy policies, their accessibility and presentation within app marketplaces play a critical role in shaping both user understanding and regulatory compliance. In principle, platform operators require developers to provide a publicly accessible, non–geo-fenced privacy policy URL as part of app metadata, ensuring that disclosures are available to users prior to installation. In practice, however, the availability, format, language, and app-specificity of these disclosures vary substantially - even among highly popular applications.

In this thesis, we focus exclusively on the Google Play Store, motivated by its dominant global market share and its central role in shaping mobile privacy disclosure practices. Android applications account for the majority of mobile installations worldwide, and Google Play enforces a structured disclosure regime that combines mandatory privacy policy links with developer-declared Data Safety labels (cf. Figure 1.1). This combination makes Google Play a particularly relevant environment for examining privacy transparency at scale.

We examine highly downloaded apps, where developers are generally more resourced and face stronger reputational, regulatory, and platform-level incentives to comply with disclosure requirements. Our research findings are centred around a large-scale analysis of approximately 15,000 mobile apps, spanning both mobile games and generic apps (see Sub Sections 4.3.5 and 5.3.2 for full dataset composition). For each app, we collected key forms of public-facing metadata, including download counts, developer-provided Data Safety declarations, and privacy policy URLs. Privacy policies were then retrieved using developer-declared links and downloaded in their rendered HTML form, including linked resources such as JavaScript, reflecting how policies are actually presented to end-users.

Despite explicit platform requirements, we observe that accessibility issues persist even among this upper tier of apps. A non-trivial fraction of applications either fail to provide a usable privacy policy link or rely on formats that undermine accessibility. Common issues include missing or unreachable URLs, policies hosted as static documents (e.g., PDF or plain text files), and disclosures embedded within third-party document hosting services. Such practices increase friction for users attempting to access privacy information and complicate automated analysis. Language accessibility presents an additional barrier. Even when app listings are retrieved via English-speaking regions (i.e. geo location set to Australia when crawling), a noticeable subset of privacy policies are written exclusively in non-English languages. In these cases, end-users are implicitly required to rely on translation services to understand legally binding disclosures. This raises concerns about meaningful consent, particularly when policies govern applications with millions or tens of millions of downloads.

Another striking characteristic of the ecosystem is the widespread reuse of privacy policies

across multiple apps. A substantial portion of popular applications share identical policy URLs, often spanning dozens of titles produced by the same developer. While policy reuse may reduce maintenance overhead, it weakens app-specific accountability and complicates comparisons with platform-level summaries such as Data Safety labels. When a single generic policy governs diverse applications with differing permissions and functionalities, users are left to infer how disclosures apply to a specific app instance.

Apart from the above mentioned key observations, We will further examine literature on privacy policy accessibility in Section 2.2 and further details on our findings in Chapters 4 and 5. Taken together, these observations illustrate that privacy policy accessibility is not merely a question of whether a document exists, but how it is surfaced, scoped, and contextualised within app metadata. Even among highly downloaded apps where incentives for compliance are strongest, end-users encounter missing links, inaccessible formats, language barriers, reused policies and generic disclosures that dilute transparency.

1.3 Limitations of Existing NLP Approaches for Privacy Policy Analysis

Analysing privacy policies via NLP approaches has been a long-standing focus of the research community. Early work focused primarily on classical NLP tools [41, 42] later surpassed by embedding-based transformer models such as BERT variants trained on privacy datasets to perform tasks like paragraph classification, summarisation, contradiction detection, and extraction of data items [43, 44]. Although these approaches offer computational efficiency and interpretable vector spaces, they suffer from notable drawbacks as highlighted below. We further indicate the most suitable technical chapter where we discuss these drawbacks in detail, often with related experiments.

- **Lack of explainability:** Embedding-based methods typically operate as black box models and post-hoc explainability techniques, such as LIME [45] offer only coarse, local explanations that provide limited insight into model reasoning and lack the flexibility required to support fine-grained verification of legal claims or compliance assessments. *[Chapter 03]*
- **Semantic search unreliability:** Embedding-based retrieval may produce noisy results, particularly when policy text is highly heterogeneous or context-dependent. Semantically similar sentences may convey different obligations or scopes, while surface-dissimilar text may describe equivalent practices. *[Chapter 04]*
- **Limited handling of structure:** Classical models trained for tasks such as named entity recognition (NER) or sentence-level classification treat documents as flat sequences,

ignoring the developer-intended flow, sectioning, and nesting that determine how disclosures relate. *[Chapter 05]*

More recent methods have explored autoregressive LLMs for policy summarisation, question answering, contradiction detection, and multi-label classification [46, 47, 48]. LLMs, however, introduce another set of challenges:

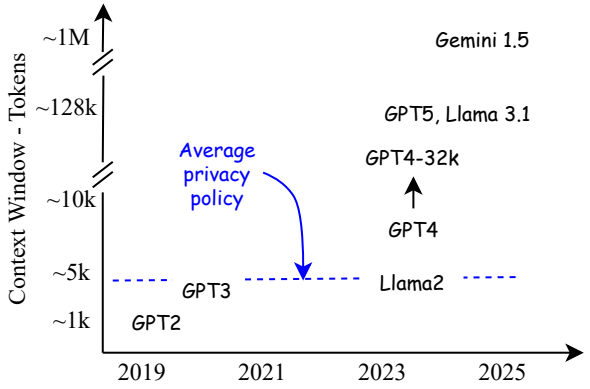

Figure 1.4: Growth timeline of LLM context window size: Early GPT and Llama models did not constitute a sufficiently large context window to accept full privacy policy text plus any prompts, given that the average tokenised length of our dataset is approximately 4100 tokens. In comparison, Llama3.1 has a 128k context window size comfortably to fit even the tail end of long privacy policies.

- **Hallucination risks and lack of verifiability:** State-of-the-art generative models may fabricate data practices not explicitly stated in the policy text, rely on implicit assumptions, or conflate unrelated disclosures. More importantly, their outputs do not inherently support verifiability, making it difficult to establish whether a generated claim is grounded in the source document; an unacceptable risk in regulatory or compliance-driven settings. *[Chapter 03]*

- **Challenges in adaptation:** Regulatory assessments / platform integrity checks require deployment of frameworks at scale, across diverse range of privacy policy inputs. Unlike embedding models that are easier to fine-tune, any downstream adaptation requires prompt engineering (e.g. optimising chain of thoughts prompts) or domain/task adaptation (e.g. supervised fine tuning) often under hardware constraints. *[Chapter 04]*

- **Managing the context window:** Modern LLMs now provide context windows capable of accommodating full-length privacy policies (as illustrated in Figure 1.4). However, beyond computational limitations, there remains no straightforward framework for quantifying output quality or verifying correctness—particularly when alignment techniques such as preference optimisation are employed to shape model behaviour for deployment-specific requirements. *[Chapter 05]*

These limitations collectively illustrate the need for hybrid, systematic approaches combining the semantic precision of encoder models, NLU and text-generative (autoregressive) capabilities of decoder models, and reasoning mechanisms that can verify or refine extracted information. We will further discuss in detail about existing work on leveraging NLP for privacy policies in Chapter 2 literature review.

1.4 Research Gaps in Automated Privacy Compliance at Scale

As we tackle the aforementioned key challenges of recent NLP landscape, we discover some notable gaps when it comes to at-scale deployment of the automated NLP frameworks that are targeted towards privacy compliance checks. While some of these gaps may be related with the key challenges of NLP we identified from previous section, we specifically highlight the following in the perspective of deployment of tools for platform integrity. We also highlight a technical chapter that can be mostly referred to, in terms of addressing these gaps.

- **Need for scalable, explainable, trustworthy Solutions:** Regulators, platform operators, and end-users need automated systems that not only classify or extract information from privacy policies but also provide human-interpretable explanations. This requires incorporating mechanisms such as entailment verification, multi-stage reasoning, and explicit separation of candidate outputs from final validated results. [Chapter 03]

- **Cross-disclosure misalignment:** Mobile app developers often provide two concurrent disclosures: a full privacy policy and a summarised Data Safety label. Empirical studies highlight widespread inconsistencies between these documents, where labels underdeclare practices disclosed in policies, or vice versa. Such misalignment can stem from outdated policies, generic policy templates shared across multiple apps, ambiguous language, or deliberate minimisation of perceived risks to users. Without automated tools capable of precise comparison, marketplace oversight remains largely reactive. [Chapter 04]

- **Information extraction limitations:** Embedding-only methods struggle with reliable extraction in large-scale deployments, whereas LLM-only methods, despite strong NLU of textual information, lack robustness and verifiability. These shortcomings undermine the reliability of downstream tasks such as detecting missing practices and resolving contradictions. [Chapter 04 and 05]

- **Structural ambiguity in policy texts:** The separation of data items, purposes and sharing practices across different sections in a privacy policy document leads to ambiguity

in how users (and algorithms) interpret them. Without explicit linking phrases or cross-references, relationships must be inferred implicitly, which may not reflect developer intent. Automated systems that ignore these structural cues risk over-generalising or under-specifying mappings, leading to flawed compliance assessments. Existing solutions rarely leverage the full structural hierarchy of policy documents. Section headings, hierarchical nesting, scoped definitions, and logical segmentation are often treated as incidental metadata rather than central signals that determine how information should be interpreted. [Chapter 05]

1.5 Overview of Thesis Contributions and the Scope

Across the integrated body of technical work presented in this thesis, we propose three complementary frameworks that address distinct layers of the *privacy compliance* drawbacks visible in mobile app ecosystems. Collectively, these contributions advance the state of automated privacy policy analysis by moving beyond surface-level policy-text evaluation toward scalable, explainable, systematic and verifiable compliance assessment.

An overview of the scope addressed by this thesis is illustrated in Figure 1.5. This diagram is centred around **(a)** - "privacy policies" that the mobile app developers are mandated to provide when offering app-based services. In general practice, some service providers operate both mobile and web-based online services (e.g., Meta providing the Facebook mobile app alongside a web platform), and may reuse a single privacy policy across multiple service contexts. However, in this thesis, we explicitly focus our analysis to disclosure consistency - **(b)** relevant to *mobile app services only*, and exclude compliance assessment of corresponding web-based services.

Beyond internal consistency, privacy policies must comply with regulatory requirements governing how personal data are collected, used, shared, and retained, as well as how these practices are disclosed to end users. We highlight this in the diagram notation - **(c)**. In parallel, app marketplace operators such as Google Play additionally impose platform specific disclosure requirements, including developer declared Data Safety labels, app privacy summaries, and related metadata designed to increase transparency at the point of installation, as depicted in the diagram notation - **(d)**.

Crucially, compliance cannot be assessed solely at the level of disclosures. The data practices ultimately rendered to end users during mobile app execution - **(f)** must align with both the privacy policy and platform-level disclosures. Moreover, privacy policies themselves must be transparent, comprehensible, and accurate to meaningfully support informed user decision-making as shown in **(e)**.

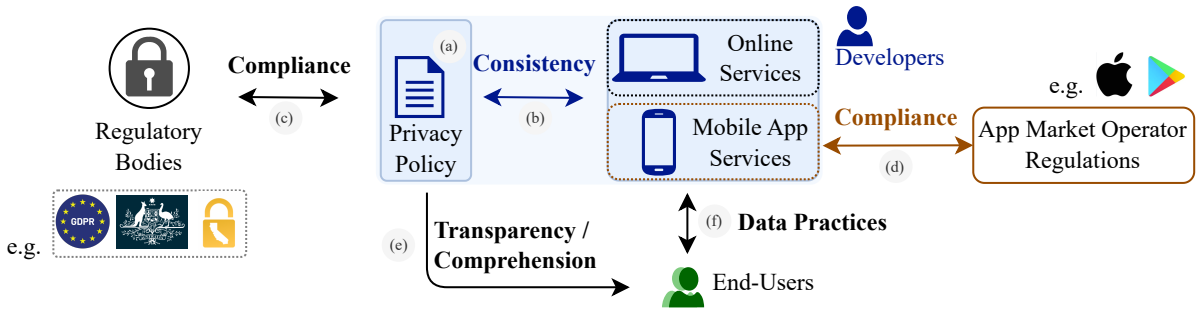

Figure 1.5: An overview highlighting key areas of this research. Regulatory Body examples (left to right): EU:GDPR, AU:APP, US:CCPA, and app market operator examples (left to right): iOS App Store, Android Play Store.

Chapter		Contributes to: Primary	Secondary
03	Entailment-Driven LLMs	**(a)**,	(e)
04	PrivPRISM	**(b)**,	(d),(f),(a),(c),(e)
05	PrivSTRUCT	**(e)**,	(b),(c),(a)

Table 1.1: Overview of our key contributions in relation to Figure 1.5

This multi-layered and interconnected compliance landscape motivates the need for automated methods capable of reasoning across disclosure documents, metadata, and regulatory expectations of mobile app. Within this scope, the technical contributions of this thesis are structured around three complementary frameworks. Table 1.1 summarises the primary and secondary contributions of each chapter and situates these contributions within the broader framework illustrated earlier.

1.5.1 Chapter 03: Entailment-Driven Privacy Policy Classification Using LLMs

This chapter introduces a novel LLM-based pipeline for privacy policy text analysis that improves prediction precision while explicitly supporting explainability. We propose a multi-stage language modelling framework that integrates (i) an explained classifier that generates both predictions and natural-language rationales, (ii) a blank-filling module that re-evaluates those rationales, and (iii) an entailment verifier that validates the final outputs against the source text. Extensive evaluation on the publicly available OPP-115 dataset demonstrates substantial performance gains over vanilla LLM baselines, alongside reasoning traces that better align with legal expert annotations.

The primary contribution of this chapter lies in advancing automated understanding of privacy policy content **(a)** through our proposed framework. By structuring inference as a verifiable reasoning process rather than a single-pass prediction task, our approach directly improves the transparency and interpretability of policy disclosures **(e)**.

1.5.2 Chapter 04: PrivPRISM: Automatically Detecting Discrepancies Between Google Play Data Safety Declarations and Developer Privacy Policies

This chapter explores the widespread misalignment between developer-provided privacy policies and Google Play Store Data Safety declarations. We propose PrivPRISM, a systematic language modelling framework combining encoder- and decoder-based models for fine-grained information extraction and verification of data practices. Using PrivPRISM, we conduct a large-scale empirical analysis on approximately 10,000 of popular mobile apps, revealing widespread under-declaration of practices in Data Safety labels, large-scale policy reuse across unrelated apps, and significant transparency gaps that undermine marketplace expectations.

The primary contribution of this chapter lies in assessing cross-disclosure consistency **(b)** at scale. In addition, we examine how developers adapt or exploit Google Play's disclosure requirements **(d)** and investigate the relationship between stated disclosures and observed behaviour through static code analysis evidence **(f)**. Beyond discrepancy detection, this chapter also contributes to improving privacy policy understanding **(a)**, examining how policy structure aligns with regulatory expectations **(c)**, and demonstrating how automated auditing can enhance transparency **(e)**.

1.5.3 Chapter 05: PrivSTRUCT: Untangling Data Purpose Compliance of Privacy Policies in Google Play Store

This chapter tackles the structural ambiguity inherent in privacy policies, where data items, purposes, and sharing practices are often distributed across loosely connected sections. We introduce PrivSTRUCT, a hierarchical analysis framework that reconstructs developer-intended disclosure structure through a systematic combination of vanilla and preference optimised decoder models, encoder-based semantic classifiers, and a structured mapping pipeline that infers dependencies between policy sections.

Benchmarking shows that PrivSTRUCT substantially outperforms state of the art frameworks such as PoliGraph [49] particularly in identifying distinct data items, extracting precise purposes, and accurately mapping purpose–item relationships. When deployed at scale, PrivSTRUCT uncovers systematic disclosure patterns including purpose dilution and over-generalised global purposes, which adversely affect policy clarity and interpretability.

Accordingly, this chapter primarily advances transparency and comprehension of privacy policies **(e)**. It further contributes to analysing cross-disclosure inconsistencies **(b)**, assessing developer intent in relation to data control and regulatory compliance **(c)**, and improving privacy policy parsing through structurally informed NLP methods **(a)**.

1.5.4 Organisation of the remainder of this thesis

In Chapter 2, we discuss the literature related to mobile app ecosystems, empirical studies and NLP based solutions in privacy policy analysis. Chapters 3, 4 and 5 elaborate on the technical contributions of this thesis as detailed above in Sub-sections 1.5.1, 1.5.2 and 1.5.3 in respective order. Finally, in a concluding remark, we highlight overall implications and possible future directions of this thesis in the Chapter 6.

Chapter 2

Literature Review

App stores serve as the primary gateways through which mobile device users discover, select, download, and sometimes purchase applications. While over 300 app stores exist worldwide [50], the market is overwhelmingly dominated by two platforms: the Google Play Store for Android and the Apple App Store for iOS. In 2019, Google reported over 2 billion active Android users worldwide [51], with approximately 3-3.5 million apps available on the Play Store, compared to 1-1.5 million on the Apple App Store [52]. Although recent regulatory changes and data safety mandate related purges have fluctuated these numbers with Android apps recorded at roughly 2.7 million in early 2024 [53], the duopoly remains unchallenged. Alternative stores managed by vendors like Samsung, Huawei, and Amazon collectively contribute to less than 5% of global downloads [52]. As of mid-2025, Android holds a global operating system market share of 43.1%, compared to 16.8% for iOS [54].

Considering the business model, Apple derives approximately 80% of its revenue from hardware sales, effectively incentivising a closed, tightly controlled ecosystem ("walled garden") that prioritises device security and user trust. In contrast, Google derives nearly 90% of its revenue from advertising, a model that benefits from the widespread reach and openness of the Android platform [52]. This separation in governance strategies has substantial implications for app safety. The relative openness of the Android ecosystem has historically correlated with higher security risks; a 2020 study reported that Android devices encountered nearly 15 times more malware infections than their iOS counterparts [55]. While Apple claims rigorous manual review of all App Store submissions to mitigate privacy invasions [56], the sheer scale of the Play Store and developers' expectation of quick app turnarounds make similar manual oversight challenging for Google.

Either way, the number of apps developed directly by Google and Apple represents a mere fraction of these ecosystems. The overwhelming majority of the millions of available apps are created by third-party developers, who exhibit highly heterogeneous data practices that often vary significantly from platform guidelines. Consequently, privacy policies serve as the critical medium for these developers to disclose their specific data practices to users. Given Android's global market dominance and the sheer scale of its third-party developer based app ecosystem,

the analysis presented in this thesis is primarily conducted within the context of the Google Play Store.

The remainder of this chapter reviews the literature related to mobile app data privacy disclosures. Section 2.1 examines the related work of privacy labels and Section 2.2 discusses the technical challenges and methodologies we observe in related studies with large-scale crawling of app markets. Section 2.3 synthesises key empirical studies on privacy policies, and Section 2.4 on accessibility of privacy policies based on mobile apps. Section 2.5 reviews automated methods for improving policy understanding, from classical machine learning to recent generative AI approaches. Finally, Section 2.6 provides the theoretical foundations of the Natural Language Processing (NLP) techniques that underpin these automated analyses.

2.1 Privacy Labels in App Markets and Challenges

Although the concept of standardised 'privacy nutrition labels' was proposed by Kelley et al. as early as in 2009 to mirror food nutrition facts that are easy for users to understand at a quick glance [57, 58], it was not until 2020, nearly a decade later, that Apple operationalised this framework through the introduction of iOS App Privacy Labels. These labels aimed to provide a user-friendlier disclosure concept in contrast to the traditional privacy policy concept. However, iOS still mandates that a comprehensive privacy policy is required in the app submission accompanying the App Privacy Labels. Following their deployment, researchers began to investigate the practical challenges of generating accurate labels. Through interviews with iOS developers, Li et al. [59] found that inaccuracies often stemmed from deep misconceptions regarding platform-specific definitions, such as the distinction between "data used for tracking users" and "data linked to users" as well as a lack of awareness regarding the data practices of embedded third-party libraries. To address these knowledge gaps, Gardner et al. [60] designed and evaluated "Privacy Label Wiz," a tool combining static source code analysis with an interactive wizard, demonstrating that developers could generate more compliant labels when guided through the opaque data collection behaviours of third-party libraries. In a large-scale analysis of over 1,687 iOS apps, Koch et al. [28] utilised network traffic monitoring to reveal that at least 276 apps explicitly violated their own privacy labels by transmitting data they claimed not to collect, suggesting that the platform's approval process does not effectively validate these declarations and non-compliance remains a significant issue.

With Google's introduction of the Data Safety section for Android in 2022, research expanded to compare the two ecosystems. Khandelwal et al. conducted a large-scale measurement of over 100k apps cross-listed on both platforms [31], uncovering a 60% inconsistency

rate where the same app reported different privacy practices on iOS versus Android. Furthermore, user-centric studies by [32] highlighted usability trade-offs; while Android's grouping of data practices based on first party collection or third party sharing was found to be intuitive, users were frequently misled by the interface design and surprised by policy "loopholes" that allowed developers to omit disclosing ephemeral (e.g. weather app only uses your location data in memory and does not store the data for longer than necessary to provide weather information) data processing.

More recently, scrutiny of Android's Data Safety section has revealed pervasive underreporting. Arkalakis et al. performed a longitudinal dynamic analysis of nearly 5,000 Android apps, finding that 81% misrepresented their data practices in the Data Safety sections, with majority of them (79.4%) do not asking the end-user to provide consent for the data they collect and share [33]. Complementing this, Khedkar et al. utilised static analysis to show that developers frequently struggle to accurately report data collection due to Google's abstract definitions and a lack of automated tools to identify data collected by system APIs [34].

2.2 Crawling App Markets

Crawling the app markets and obtaining a database of metadata resources (i.e. supplementary information developers provide in app listing pages such as app descriptions, privacy policy links, data safety / app privacy declarations, statistics such as number of downloads, etc.) is a main contribution of many prior research in order to properly analyse the compliance issues by cross-referencing them with the stated privacy policies. Due to the dynamic nature of mobile app updates and submissions, it is not possible to have a unique benchmarking dataset regarding the crawled apps as they become obsolete in a short span of time. Researchers employ different methodologies in crawling apps as there is no straightforward way in Play Store nor in App Store to access the full submission list. As a workaround, Zimmeck et al. in their work of Mobile App Privacy System (MAPS) [61], used to recursively crawl through Play store pages by following the links to similar apps. This method is still effective as Play Store web page provide a sufficient amount of similar app recommendations for one search result. Simultaneously looking for the counter part iOS app while crawling Android app store is intuitive but can have challenges. Some Android apps do not have a counterpart iOS app [62] and some developers may develop unique metadata for iOS. Observing the similarity of various apps is not a novel method. Rajasegaran et al. [63] introduced a neural embedding framework to identify similar apps using the features of style plus content embedding of the app icon and text embeddings generated from app description.

Scalability of app crawling can pose different challenges as the app markets contain millions

of apps. Crawling is usually done through web requests and if the host identifies recursive requests coming from a same network address, such points of origins can be blacklisted. Viennot et al. [64] utilised various hacks to go through the security measures imposed by Google to avoid app crawling such as using a pool of credentials, rate limitations and proxying via external service providers. Kumar et al. [65] mentioned that they had to use 10 linux measurement machines and 10 android devices in order to effectively generate an app database.

In summary, the systematic analysis of mobile app privacy compliance is constrained not only by the content of the policies but by the fundamental technical challenge of acquiring representative data, especially, developer provided privacy policy links and data label sections. High-volume crawling requires navigating dynamic market environments, overcoming discoverability limitations, and bypassing anti-scraping defences. Having established the methodologies used to acquire these datasets, the next section onwards focus on privacy policy documents, and reviews the empirical findings derived from them, accessibility challenges of them and automated methods for improving privacy policy understanding.

2.3 Empirical Studies on Privacy Policies

A privacy policy describes an application's data collection, sharing, retention, and security practices and, more importantly, is a mandatory requirement for an application's submission for public distribution. These disclosures frequently encompass personally identifiable information (PII), rendering end-user comprehension of data collection and sharing practices, as well as available data management mechanisms that are critical for transparency between service providers and users.

2.3.1 Privacy policies: long, complex and incomprehensive

A widely acknowledged characteristic of privacy policies is their excessive length, linguistic complexity, and general incomprehensibility to end users [8, 9, 10]. Owing to their legalistic nature, multiple studies employing established readability metrics argue that a college-level reading capacity is typically required to interpret these documents accurately [8, 66, 67, 68]. The challenge is further compounded by inherent ambiguities in policy language, which have been shown to impede understanding even among policy experts [69, 70].

In terms of scale, privacy policies exhibit median word counts ranging from approximately 1,500 [18] to over 2,500 words [19], rendering them prohibitively time-consuming for individual users to read in full [9, 18]. Prior work estimate that an average user would need to spend at least 181 hours annually to read all applicable privacy policies encountered in routine digital

interactions [9, 71]. Empirical evidence consistently demonstrates that such demands significantly exceed realistic user engagement levels. For instance, a recent study reports that 94% of Australians do not read all privacy policies that apply to them [11]. Comparable trends are observed internationally, with only 9–22% of users in the United States and 12–13% of users in the United Kingdom reporting that they always or frequently read applicable privacy policies [12, 13].

2.3.2 Regulatory impact

In response to longstanding concerns surrounding transparency and informed consenting, recent regulatory initiatives, most notably the GDPR, have sought to address deficiencies in privacy disclosures by mandating the provision of privacy policies and encouraging clearer communication of data practices. These interventions have led to measurable outcomes, including a reported 4.9% increase in the availability of privacy policies following regulatory enforcement [16].

However, longitudinal analyses suggest that these regulatory efforts have also generated unintended consequences. Rather than improving readability, privacy policies have expanded substantially in length, with global increases of approximately 25% observed over time [17]. This growth exacerbates existing usability and comprehension challenges, further distancing users from meaningful engagement with privacy disclosures.

As a result, regulatory mandates designed to empower users may inadvertently contribute to widespread disengagement from privacy policies [20]. This reinforces a persistent paradox in contemporary data protection regimes: individuals are formally expected to make informed decisions regarding the handling of their personal data, yet the information required to do so remains largely inaccessible in practice.

2.4 Accessibility of Privacy Policies

This section regarding the accessibility of a privacy policy reflects on any intermediate stage from an end-user discovering a privacy policy link up until to the point of privacy policy text being rendered for reading.

2.4.1 Availability and link validity

Mobile app marketplaces rely on developers to host their privacy policies on externally accessible web pages and require only that a link be provided on the app's download page. As a

result, the availability of a privacy policy constitutes a fundamental prerequisite for any empirical analysis of privacy practices. However, prior work has demonstrated that the mere existence of a link does not guarantee effective access to policy content.

Zimmeck et al. [61], in a 2019 large-scale study, question the effectiveness of Google's disclosure requirements, reporting that 49.1% of apps lacked a privacy policy link despite 88.6% of those apps engaging in at least one privacy-relevant practice. The authors further highlight that the presence of a link alone is insufficient, as links may be non-functional, redirect to error pages, or point to content written in languages inaccessible to the target user base.

Recent findings reinforce these concerns. A 2022 study [65] similarly reports widespread issues involving missing or broken privacy policy links, as well as expired domains, indicating that accessibility challenges persist despite increased regulatory and platform-level attention.

In response, both Google's Play Store and Apple's App Store have strengthened enforcement by making the submission of a valid privacy policy link a mandatory requirement. Developers may now be prevented from publishing new applications or rolling out updates without providing such a link [72, 73]. While these changes suggest improvements in nominal availability, their practical effectiveness remains an open question.

2.4.2 Challenges in technical accessibility

Beyond availability, the usability of privacy policies presents a separate and equally significant challenge. Habib et al. [74] argue that the existence of a privacy policy does not necessarily imply that users can effectively access or understand its contents. Based on user interviews, they report that poor formatting, dense presentation, or explanatory text frequently hinders users' ability to locate relevant information.

From a technical perspective, structural heterogeneity across privacy policies further complicates accessibility and analysis. Prior work highlights the difficulty of extracting policy text due to inconsistent document structures, including line breaks, nested lists, and segmented content divisions [75]. Polisis [14] additionally emphasises the dynamic nature of privacy policies in HTML form, which may include expandable sections or interactive elements that obscure content during automated processing.

Accessibility is further affected by geographic variability. As noted in [65], privacy policies may be served conditionally based on the user's location, meaning that the same developer can present different policy versions to users in different jurisdictions, such as Australia and the United States. This variability introduces additional challenges for comparative analysis and regulatory assessment.

2.4.3 Challenges in standardisation

Accessibility challenges also arise from ambiguities in policy scope and applicability. Cross-domain applications that span multiple functional categories may reference privacy policies that do not clearly align with the app's declared category, complicating efforts to interpret the policy's relevance [76]. Similarly, developers may provide a single privacy policy governing multiple applications, resulting in fewer unique policy documents than app entities and potentially distorting large-scale analyses.

Even when accessible, privacy policies may exhibit internal inconsistencies. Policylint [41] documents cases in which policies include blanket statements claiming non-collection of personal identifiers, while simultaneously disclosing the collection of specific subcategories of personal data. Such contradictions necessitate careful scrutiny in compliance-focused evaluations.

Efforts to standardise privacy policy representations have historically struggled to keep pace with the functional and scalability requirements of modern disclosures. Notably, the Platform for Privacy Preferences (P3P) proposed an XML-based, machine-readable policy format, but despite extensive review, it failed to achieve widespread adoption [77]. Consequently, reliance on natural-language privacy policies and implicit trust in developer transparency remains the dominant paradigm; one that limits the detection of questionable, undeclared, or opaque data practices commonly observed in mobile applications [78].

2.5 Automated Methods for Improving Privacy Policy Understanding

The previous section established that mobile app privacy policies suffer from persistent readability, accessibility, and compliance-related challenges. In response, a substantial body of research has explored automated methods to improve the interpretation, analysis, and usability of privacy policy text. This section organises these efforts into two complementary perspectives.

First, we review methodological advances in automated privacy policy analysis, progressing from traditional machine learning approaches to deep learning–based text encoding and, more recently, generative AI models. Second, we examine user-centric systems that operationalise these methods to support querying, detection of user choices, summarisation, and interactive assistance.

We summarise the related work we discuss in each subsection to the quick-overview depicted in Table 2.1. Note that we do not discuss in detail about the respective NLP concepts when discussing each prior work. We forward the readers to cross refer with Section 2.6 for explanations on NLP terminology.

Focus Area	Frameworks and Key Contributions
2.5.1 Methodological Advances	
Traditional ML & Feature Engineering	(2012) Ammar et al. [79]: Logistic regression for concept detection. (2012) Costante et al. [80]: Evaluation of k-NN, SVM, Decision Trees. (2014) Privee [81]: Naïve Bayes classifiers. (2016) Wilson et al. [82]: SVM baselines; OPP-115 corpus. (2018) Liu et al. [83]: TF-IDF combined with SVMs. (2019) PolicyLint [41]: Ontology-based contradiction detection. (2019) MAPS [61]: Large-scale compliance (TF-IDF + SVM).
Deep Learning & Context-Aware Encoding	(2018) Polisis [14]: CNNs for multi-label classification. (2020) Nejad et al. [84]: Fine-tuned BERT on OPP-115. (2020) Mustapha et al. [85]: XLNet for improved classification. (2021) PrivBERT [44]: PrivaSeer corpus and PrivBERT. (2023) Adhikari et al. [43]: longitudinal analysis of policies
Generative AI (LLMs)	(2023) PolicyGPT [47]: Zero-shot classification with GPT. (2024) Rodriguez et al. [46]: Prompt engineering and few-shot learning.
2.5.2 User-Centric Systems	
Choice Detection	(2016) Sathyendra et al. [86]: Classification of choice instances. (2017) Opt-Out Easy [87]: Browser extension for finding opt-outs. (2020) Kumar et al. [75]: Scalable choice detection.
Question Answering	(2017) Sathyendra et al. [88]: Open vs. Closed domain QA. (2018) PriBot [14]: QA interface built on Polisis. (2018-22) PrivacyCheck v1-3 [89, 90, 91]: Answer pre-defined Qs.
Summarisation & Assistants	(2020) Gopinath et al. [92]: Seq2Seq title generation. (2024) Privacify [93]: Mistral 7B for segment summaries. (2025) CLEAR [48]: Context-aware risk warnings via LLMs. (2025) PRISMe [94]: GPT-4o based browser assistant.

Table 2.1: Overview of automated methods for privacy policies.

2.5.1 Methodological advances in automated privacy policy analysis

The automation of privacy policy analysis fundamentally relies on the ability to translate legal policy text into more meaningful representations that can be better leveraged to user comprehension or downstream tasks. Over the past decade, this methodological landscape has evolved significantly, driven by broader advancements in NLP. This evolution can be categorised into three distinct eras: traditional machine-learning and feature based classification, deep learning and context-aware encoding, and the recent emergence of generative models capable of zero-shot reasoning.

2.5.1.1 Traditional machine learning and feature-based classification

Early automated approaches to privacy policy understanding framed the task primarily as a text classification problem, enabled by advances in feature engineering and classical machine learning. Ammar et al. proposed detecting the presence or absence of privacy-related concepts using salient linguistic features evaluated through logistic regression classifiers [79]. Costante et al. further explored alternative models, including k-NNs, SVMs, and decision trees, demonstrating the feasibility of supervised learning for privacy concept detection [80].

Zimmeck et al. in [81] evaluated multinomial Naïve Bayes classifiers to identify the presence of data practices such as collection, ad-tracking, and ad-disclosure. Subsequent work by Wilson et al. in [82] showed that SVM-based models significantly outperformed earlier approaches. This work marks a major milestone through the release of the OPP-115 corpus which is a dataset of 115 privacy policies annotated by legal experts with nearly 23k fine-grained annotations. OPP-115 has since been widely regarded as a gold-standard benchmark for privacy policy classification tasks. Building upon this foundation, Liu et al. [83] demonstrated that TF–IDF vectorisation combined with SVM classifiers further improved performance on OPP-115, reinforcing the effectiveness of feature-based models when paired with high-quality expert annotations.

Andow et al. proposed PolicyLint [41], a sentence-level NLP framework that focuses on detecting logical inconsistencies within privacy policies. The system constructs ontologies from over ten thousand mobile app policies using predefined seed terms, extracting tuples comprising actor, action, data object, and entity via dependency parsing. Logical rules are then applied to identify contradictions, with results indicating that 14.2% of policies contain conflicting statements; 510 of which were manually verified.

Zimmeck et al. further scaled automated analysis through MAPS, a system designed to evaluate privacy compliance across nearly one million Google Play Store apps [61]. MAPS uses a three-tier classification schema (data type, party, modality) as explained in [95] and

applies TF–IDF features with SVM classifiers to detect disclosed data practices. These are subsequently compared against behaviours observed in app source code, enabling the identification of discrepancies such as undisclosed device identifier transmission.

2.5.1.2 Deep learning and context-aware text encoding

As privacy policy analysis matured, researchers began moving beyond bag-of-words representations toward neural models capable of capturing semantic context and hierarchical structure. Harkous et al. introduced Polisis, a framework that extends flat classification into a multi-level ontology of data practices. Polisis employs a convolutional neural network (CNN) architecture with ReLU and max-pooling layers to assign probabilistic labels across both coarse- and fine-grained categories, enabling multi-label classification of policy segments [14].

The introduction of transformer-based architectures marked a decisive shift in privacy policy analysis. Nejad et al. demonstrated that fine-tuned BERT models substantially outperform CNN-based and traditional deep learning approaches on OPP-115, achieving approximately an 11 percentage point improvement in F1 score [84]. By releasing code and standardised data splits, this work established a reproducible benchmark highlighting the importance of contextualised representations for legal text.

Mustapha et al. further improved classification performance by replacing BERT with XLNet, achieving superior results without requiring domain-specific pre-training [85]. Srinath et al. introduced the PRIVASEER corpus of nearly one million English-language privacy policies and used it to domain-adapt RoBERTa into PrivBERT, which outperformed prior baselines including Polisis and generic RoBERTa models [44]. Subsequent analyses, such as [43], applied sentence-level XLNet classifiers to track longitudinal changes in privacy policy composition and semantics.

Other work has extended classification beyond policy text itself. Uddin et al. analysed cross-domain app functionality by applying supervised classification to tokenised and tagged app descriptions, ranking coexisting features across 20,000 apps to surface implicit personalisation practices [76].

2.5.1.3 Generative models for textual decoding

Recent advances in large language models (LLMs) have enabled zero-shot and few-shot approaches to privacy policy analysis. PolicyGPT [47] demonstrated that decoder-only models

can classify privacy policy paragraphs without task-specific training, achieving strong performance despite deviating from conventional OPP-115 evaluation pipelines. Rodriguez et al. propose using LLMs, specifically ChatGPT and Llama 2, as a cost-effective and scalable method for extracting data practices from privacy policies. The authors demonstrate that through optimised prompt engineering and few-shot learning, LLMs can achieve an F1 score exceeding 93% on benchmark datasets, significantly outperforming prior state-of-the-art automated analysis techniques [46]. Follow-up work has also shown that both proprietary and open-source LLMs exhibit competitive performance in extracting privacy norms under limited supervision [96].

2.5.2 User-centric systems for privacy policy interaction

While methodological advances focus on improving automated interpretation, a parallel line of work reframes privacy policy analysis from the end-user's perspective, emphasising interaction, comprehension, and decision support.

2.5.2.1 Detection of user choices and data control mechanisms

Detecting whether a privacy policy provides users with meaningful choices such as opt-in or opt-out options has been studied as a specialised classification problem. Sathyendra et al. modelled this task as binary classification, identifying "choice instances" using lexical features, modal verbs, and opt-out phrases [86]. Logistic regression outperformed more complex classifiers in this setting.

A subsequent two-stage architecture introduced by Sathyendra et al. first detects the existence of a choice and then categorises its type (e.g., opt-in, opt-out, deletion), supported by active learning to refine annotations [97]. These models were later operationalised in Opt-Out Easy, a browser extension that increases the visibility of privacy choices for users [87]. Their implementation also improved baseline bag-of-words models by incorporating stemmed unigrams/bigrams and the relative location of the text within the policy document. Kumar et al. further scaled this approach by incorporating HTML structure for segmentation while retaining lightweight classifiers to ensure real-time usability [75]. They specifically retained logistic regression compared to novel but more computationally expensive transformer models to ensure that Opt-Out Easy extension could execute efficiently in end-user devices.

2.5.2.2 Question answering and interactive query systems

A distinct line of research focuses on enabling users to directly query privacy policies using natural language questions, thereby avoiding the need to manually read lengthy and complex documents. Systems in this category aim to extract and surface relevant privacy-related information in response to user queries about data collection, sharing, and control practices.

Sathyendra et al. [88] proposed a flexible framework capable of answering user questions about specific data practices (e.g., "Does this app share my location?") by extracting relevant information directly from privacy policy text. Their work distinguishes between two complementary approaches. The first, closed-domain question answering, assumes that user questions map to a predefined set of categories, specifically those defined by the OPP-115 annotation schema. In this setting, k-means clustering over embedding representations is used to align user questions with policy labels. The second, open-domain question answering, supports free-form user queries and retrieves relevant policy segments using similarity metrics over word embeddings or BiLSTM-based deep neural models with attention mechanisms for answer selection.

Harkous et al. introduced PriBot [14], a question-answering interface built upon the Polisis framework. Their methodology involves scraping privacy policies and dividing them into semantically coherent segments using a graph-based segmentation algorithm operating over word embeddings. These segments are subsequently passed through a supervised classification pipeline to predict policy attributes. When evaluated against user queries, PriBot achieved an 89% relevance rate for the top three answers returned, demonstrating the effectiveness of structured segmentation combined with classification-based reasoning.

PrivacyCheck [89] represents another early effort in this space, designed to answer ten predefined, high-level questions grounded in data handling and user control principles, such as "How does the site handle your email address?". The system employed classification models trained using the Google Prediction API, which operates as a black-box service without public disclosure of the underlying model architectures. Subsequent iterations expanded the system's scope. PrivacyCheck v2 [90] introduced additional questions aligned with GDPR compliance requirements, while PrivacyCheck v3 [91] extended the framework to support longitudinal tracking of policy changes over time.

2.5.2.3 Summarisation, readability enhancement and LLM-based assistants

Another user-centric direction seeks to improve privacy policy comprehension through automated summarisation and readability enhancement. These approaches aim to reduce cognitive load by generating concise titles, abstracts, or structured overviews of policy content.

Gopinath et al. modelled privacy policy summarisation as a sequence-to-sequence learning task, focusing specifically on the automatic generation of section titles [92]. Their approach employed a transformer-based encoder–decoder architecture. Trained on a title–paragraph dataset derived from web privacy policies, the model outperformed baseline GRU architectures, producing valid section titles in approximately 50% of cases and achieving a high manual agreement score of 0.81.

More recent systems leverage LLMs to move beyond static summarisation toward contextualised and interactive privacy assistance. CLEAR (Contextual LLM-Empowered Privacy Policy Analysis and Risk Generation) [48] is a privacy assistant designed for users interacting with LLM-based services such as ChatGPT or Gemini plugins. Unlike generic policy summarisers, CLEAR incorporates the specific context of a user's interaction including the type of sensitive data they intend to provide, and evaluates it against the corresponding privacy policy to identify potential risks. The system uses LLMs to extract structured knowledge from academic literature on LLM privacy risks, summarises the relevant sections of the policy, and generates real-time risk warnings to inform users before data disclosure occurs.

PRISMe (Privacy Risk Information Scanner for Me) [98, 94] is a browser-based extension that applies LLMs to make lengthy privacy policies more accessible during web browsing. The tool directly prompts OpenAI's GPT-4o model to process and assess privacy policy text under an "LLM-as-a-judge" paradigm. It presents users with a concise dashboard summarising key data practices and compliance indicators. In addition, PRISMe integrates an LLM-powered conversational interface that allows users to ask follow-up questions and engage in in-depth discussions about policy content, supporting a more interactive and personalised understanding of data protection terms.

Privacify [93] adopts a similar user-centric philosophy while employing a structured summarisation strategy. The system segments privacy policies into manageable units, analyses each segment using the Mistral 7B Instruct language model to extract salient insights such as data collection practices and compliance signals, and then synthesises these outputs into human-readable summaries. This approach aims to balance scalability with interpretability when processing long and heterogeneous policy documents.

The next section discusses about foundations of natural language processing closely referenced with analysing the privacy policies.

2.6 Foundations of Natural Language Processing

Natural languages have co-evolved with humans as flexible and context-rich systems for communication, in contrast to formally constructed languages such as programming languages. Natural Language Processing (NLP) concerns itself with enabling computational systems to analyse, interpret, and generate human language in a meaningful manner. Within the scope of this literature review, NLP techniques are examined primarily in the context of analysing privacy policy documents, which are authored for human consumption but increasingly processed automatically at scale. Following Table 2.2 gives a quick overview of the Sub Sections included in this section, aligned with methodological evolution of NLP concepts.

Sub.Sec.	NLP Paradigm	Representative Models / References
§2.6.1	Feature-based (classical) NLP	TF–IDF; Naive Bayes; SVMs [99]
§2.6.2	Distributional semantics (Static Embeddings)	Word2Vec [100]; GloVe [101]; FastText [102]; LexVec [103]
§2.6.3	Neural sequence modelling	RNNs [104]; LSTMs [105]; Seq2Seq [106]
§2.6.4	Attention-based architectures	Transformer [107]
§2.6.6, 2.6.7	Pre-trained encoder models	BERT [108]; RoBERTa [109]; XLNet [110]
§2.6.5, 2.6.8	Generative AI & LLMs	GPT family [111, 112, 113, 114]; Llama [115]; Mistral [116]
§2.6.9	Model adaptation and alignment	LoRA [117]; RLHF [118]; DPO [119]; RAG [120]

Table 2.2: Methodological evolution of NLP

2.6.1 Text pre-processing and feature-based NLP

Text pre-processing constitutes a fundamental step in many NLP pipelines, particularly in early feature-based approaches. The primary objective of pre-processing is to transform unstructured textual data into a form that is amenable to statistical or machine learning algorithms while reducing noise and sparsity. Common pre-processing steps include lowercasing, tokenisation, stop-word removal, part-of-speech (POS) tagging, stemming, and lemmatisation.

A token is typically defined as a contiguous sequence of characters treated as a semantic unit for downstream processing [121]. Stemming refers to the heuristic removal of affixes to approximate a word's root form, whereas lemmatisation maps words to their canonical dictionary forms based on morphological analysis.

Early machine-learning-based NLP systems relied heavily on manually designed features derived from such pre-processing steps. A representative example is the work of Pang et al.

[99], who evaluated Naïve Bayes, maximum entropy models, and support vector machines (SVMs) for sentiment classification using unigram features. Their findings demonstrated that, unlike topic classification, sentiment analysis presents additional challenges due to its dependence on subtle linguistic cues and contextual polarity.

Textual features in these models are commonly represented using vector space models such as term frequency–inverse document frequency (TF–IDF), which encode documents as weighted document–term matrices. This process, often referred to as vectorisation, enables the application of standard machine learning algorithms to textual data.

Such approaches are collectively described as bag-of- models, including bag-of-words and bag-of-POS representations. While computationally efficient, these models disregard word order and broader context, motivating subsequent research into representations that better capture semantic relationships between words. This motivation gave rise to distributional semantics and word embedding models.

2.6.2 Distributional semantics and word embeddings

Distributional semantics is grounded in the hypothesis that words occurring in similar contexts tend to have similar meanings. Word embeddings operationalise this principle by mapping each word in a vocabulary to a dense, low-dimensional vector such that semantic similarity corresponds to geometric proximity in the embedding space.

In contrast to one-hot encodings used in earlier NLP systems, embedding representations are significantly more memory-efficient and enable neural networks to generalise across semantically related words. One-hot representations treat each word as independent, requiring models to learn relationships from scratch without shared structure.

The Word2Vec framework introduced by Mikolov et al. [100] popularised efficient neural methods for learning word embeddings. The Skip-gram architecture learns to predict surrounding context words given a target word, whereas the Continuous Bag-of-Words (CBOW) model predicts a target word from its context. Both approaches yield embeddings that capture semantic and syntactic regularities.

GloVe [101] extends these ideas by combining local context-based learning with global corpus statistics. It performs matrix factorisation over word–word co-occurrence counts, focusing on non-zero entries to improve efficiency. This hybrid approach enables GloVe to leverage both local and global distributional information.

FastText [102] further incorporates subword information by representing words as bags of

character n-grams. This design explicitly models morphological structure, addressing limitations of earlier embedding methods when dealing with rare or out-of-vocabulary words. As a result, FastText can generate meaningful embeddings even for words not observed during training.

LexVec [103] proposes a weighted low-rank factorisation of Positive Pointwise Mutual Information (PPMI) matrices using stochastic gradient descent. Unlike methods that clip negative PMI values, LexVec explicitly penalises reconstruction errors for both frequent and infrequent co-occurrences. Empirical results indicate improved performance over GloVe on word similarity tasks, highlighting the importance of modelling negative co-occurrence information.

2.6.3 Neural sequence modelling

While word embeddings capture semantic properties at the lexical level, they do not inherently model sequential dependencies in text. Natural language is fundamentally sequential, and modelling word order is essential for tasks such as translation and generation. Recurrent Neural Networks (RNNs) were introduced to address this limitation by maintaining a hidden state that evolves over time as new tokens are processed.

In RNNs, information from previous inputs is propagated through recurrent connections, enabling the network to model temporal dependencies. Mikolov et al. [104] demonstrated that RNN-based language models can capture longer contextual information than feed-forward architectures. However, standard RNNs are difficult to train due to vanishing and exploding gradients and struggle to retain information over long sequences [122].

Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber [105], mitigate these issues through gated memory cells that regulate information flow. LSTMs have been successfully applied to a wide range of sequence modelling tasks.

The sequence-to-sequence (Seq2Seq) framework [106] employs an encoder–decoder architecture, typically using stacked LSTMs, to map input sequences to output sequences of variable length. Applied to machine translation, this approach achieved performance surpassing traditional phrase-based statistical machine translation systems in terms of BLEU score. Beyond translation, Graves [123] demonstrated that LSTMs are capable of generating complex structured sequences, including realistic handwritten text, through next-step prediction.

A key limitation of early encoder–decoder models lies in the compression of the entire input sequence into a single fixed-length vector. This bottleneck motivated the development of attention mechanisms, which allow the decoder to dynamically attend to different parts of the input sequence by accessing all encoder hidden states. Attention marked a critical conceptual shift in NLP and laid the groundwork for subsequent advances in neural language modelling.

2.6.4 Attention-based architectures and transformer models

The introduction of attention-only architectures marked a decisive shift in NLP, enabling models to capture long-range dependencies while supporting efficient parallel computation. Vaswani et al. [107] proposed the transformer architecture, which eliminates recurrence and convolution entirely in favour of self-attention mechanisms. The model consists of an encoder–decoder structure, where each component is composed of stacked attention and feed-forward layers.

Central to the transformer is the scaled dot-product attention function, which computes a weighted sum over value vectors based on the similarity between query and key vectors. To enrich representational capacity, the authors introduced multi-head attention, allowing the model to attend to information from multiple representation subspaces and positions simultaneously. Since self-attention lacks inherent sensitivity to token order, positional encodings based on sinusoidal functions of varying frequencies are added to the input embeddings, enabling the model to distinguish absolute and relative positions within a sequence.

The transformer was evaluated on large-scale machine translation benchmarks, including English–German and English–French datasets comprising over 40 million sentence pairs. The proposed architecture achieved state-of-the-art BLEU scores, outperforming prior recurrent and convolutional models by a significant margin, while also reducing training costs in terms of floating point operations. Crucially, the removal of recurrence enabled full parallelisation during training, substantially improving efficiency and scalability. This architectural innovation laid the foundation for modern large-scale language models.

2.6.5 Generative pre-trained transformers

Building on the transformer architecture, Radford et al. introduced the Generative Pre-trained Transformer (GPT) [111], which formalised a two-stage learning paradigm. In the first stage, the model is pre-trained on a large corpus of unlabeled text using a standard autoregressive language modelling objective, maximising the likelihood of the next token given previous context. In the second stage, the model is fine-tuned on supervised downstream tasks using task-specific objectives.

GPT adopts a unidirectional, decoder-only transformer architecture, making it naturally suited for text generation. To adapt the model to tasks such as textual entailment, question answering, and semantic similarity, the authors reformulated inputs into a single token sequence using delimiter tokens to separate different semantic components (e.g., premise and hypothesis). This unified input representation allowed GPT to improve state-of-the-art performance on nine out of twelve evaluated benchmarks, demonstrating the effectiveness of large-scale unsupervised pre-training in reducing task-specific labelled data requirements.

2.6.6 Bidirectional encoder representations

Devlin et al. proposed BERT (Bidirectional Encoder Representations from Transformers) [108], which introduced a fundamentally different pre-training strategy based on bidirectional context modelling. Unlike unidirectional language models, BERT leverages a transformer encoder to jointly condition on both left and right context at every layer.

BERT is pre-trained using two self-supervised objectives: masked language modelling (MLM), where a subset of input tokens is randomly masked and predicted, and next sentence prediction (NSP), which encourages inter-sentence coherence modelling. Fine-tuning BERT for downstream tasks requires minimal architectural modification, typically involving the addition of a lightweight task-specific output layer. Empirical evaluations across eleven NLP benchmarks, including the GLUE benchmark, demonstrated substantial improvements over prior approaches, cementing bidirectional encoders as a dominant paradigm for language understanding tasks.

2.6.7 Encoder-only transformer variants

Following BERT, several encoder-only transformer models were proposed to refine pre-training objectives and architectural assumptions. RoBERTa [109] demonstrated that BERT's performance could be significantly improved by removing the NSP objective, increasing training data size, extending training duration, and dynamically masking tokens during training. These findings highlighted the sensitivity of transformer encoders to pre-training design choices.

XLNet [110] addresses the limitations of masked language modelling; specifically data corruption and fixed input constraints, by introducing a permutation language modelling objective. By leveraging autoregressive factorisation over token permutations alongside a recurrence mechanism, the model captures bidirectional context and long-term dependencies without the need to mask tokens. This architecture effectively combines the strengths of unidirectional autoregressive models and bidirectional encoders, allowing XLNet to process lengthy policy documents without the input length limitations of BERT, while achieving strong performance on language understanding benchmarks.

2.6.8 Decoder-only language models: closed-source and open-source

Decoder-only Transformers evolved into large-scale generative language models capable of zero-shot and few-shot learning. Closed-source models, such as GPT-3, GPT-4, PaLM, and Claude, are typically trained on massive proprietary datasets using extensive computational

resources. These models exhibit strong generalisation capabilities across reasoning, summarisation, and code generation tasks but offer limited transparency and reproducibility.

In contrast, open-source decoder-only models, including LLaMA, Mistral, and Falcon, have enabled the research community to study, adapt, and deploy large language models under more transparent conditions. These models facilitate fine-grained analysis of model behaviour and support domain-specific adaptation, making them particularly suitable for research settings that require controlled experimentation.

2.6.9 Training and adaptation paradigms for LLMs

The effectiveness of modern language models depends not only on architecture but also on training and adaptation strategies. **Parameter-frozen paradigms** do not update any of the pre-trained parameters of the original model and do not associate with training costs. Pre-training refers to large-scale self-supervised learning on generic corpora, typically using autoregressive language modelling objectives, which enables models to acquire broad linguistic and world knowledge.

LLMs such as GPT-3 have few-shot learning capabilities through in-context learning [113]. By using a specific text or template, called a prompt, users can steer the model to generate desired outputs, ushering in the "pre-train and prompt" paradigm. In this context, prompts that condition the model with a few specific examples are termed *few-shot prompts*, whereas those relying solely on a template without examples are known as *zero-shot prompts*.

Chain of thought (CoT) prompting refers to steering the model towards generating step-by-step answers and is shown to perform better, especially in multi-step arithmetic and logical reasoning benchmarks, demonstrated by modifying the answers in few-shot examples to step-by-step answers [124]. Kojima et al. [125] propose *zero-shot CoT* that does not rely on showing examples to the model. Instead, they prompt the model twice, first to extract all the reasoning steps and second to extract the final answer. These parameter frozen paradigms have gained attraction due to their applicability in a wide range of NLP tasks.

Fine-tuning adapts pre-trained models to downstream tasks using supervised data and can be referred to as a **parameter-tuning paradigm**. While effective, full supervised fine-tuning is computationally expensive for large models. As a result, parameter-efficient fine-tuning (PEFT) methods have been proposed, including Low-Rank Adaptation (LoRA) [117], which injects trainable low-rank matrices into attention layers while keeping base model weights frozen. Such approaches significantly reduce memory and compute requirements. Beyond supervised learning, reinforcement learning from human feedback (RLHF) incorporates human preference

signals to optimise model outputs using reward models [118]. More recently, Direct Preference Optimisation (DPO) has been proposed as a more straightforward alternative that directly optimises preference likelihoods without explicit reinforcement learning loops [119].

REALM (Retrieval-Augmented Language Model) [126] and RAG (Retrieval-Augmented Generation) [120] are two pioneering frameworks designed to overcome the static nature of traditional language models by integrating a retrieval mechanism directly into the generation process. Instead of relying solely on parameters learned during pre-training, these models dynamically fetch relevant documents from an external corpus to inform their predictions. This architecture primarily addresses the challenge of knowledge obsolescence, allowing models to access new information that became available after their initial training without requiring costly retraining. However, while they excel at factual updates and open-domain question answering, they do not inherently solve the problem of policy understanding which refers to the ability to interpret and reason about complex rules, compliance requirements, or legal frameworks that requires specialised structural comprehension rather than just information retrieval.

Chapter 3

Entailment-Driven Privacy Policy Classification with LLMs

While many online services provide privacy policies for end users to read and understand what personal data are being collected, these documents are often lengthy and complicated. As a result, the vast majority of users do not read them at all, leading to data collection under uninformed consent. Several attempts have been made to make privacy policies more user-friendly by summarising them, providing automatic annotations or labels for key sections, or by offering chat interfaces to ask specific questions. With recent advances in Large Language Models (LLMs), there is an opportunity to develop more effective tools to parse privacy policies and help users make informed decisions.

In this chapter, we propose an entailment-driven LLM-based framework to classify paragraphs of privacy policies into meaningful labels that are easily understood by users. The results demonstrate that our framework outperforms traditional LLM methods, improving the F1 score in average by 11.2%. Additionally, our framework provides inherently explainable and meaningful predictions.

3.1 Introduction

Many online services and apps we use today collect vast volumes of personal data [127]. Beyond the first-party use of this data for purposes such as personalisation, it is often used for advertising, analytics, and user profiling. Additionally, this data can be shared with, or even sold to, third parties without the direct knowledge of users, posing serious privacy risks [127, 128]. Typically, information regarding such data collection and sharing practices is outlined in the service's privacy policies and providing the users with privacy policies is mandatory in many jurisdictions [129]. However, these policies are usually lengthy, complicated, and written in complex legal jargon. As a result, users frequently agree to data collection practices without thoroughly reading the privacy policies or comprehending the potential risks involved.

While some service providers actively attempt to enhance the readability of privacy policies [72, 130], the vast majority of these documents remain incomprehensible to end-users.

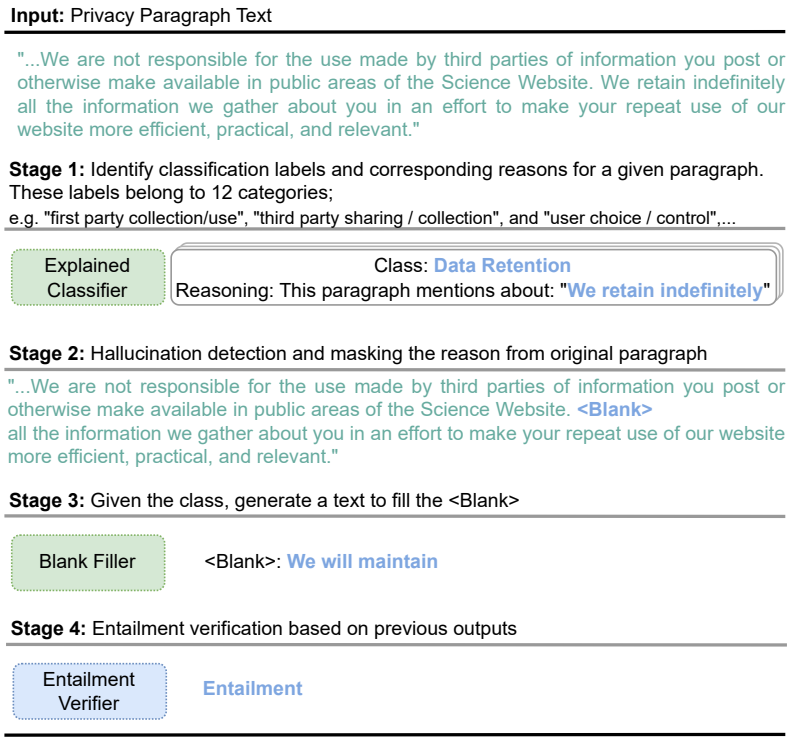

Figure 3.1: Four stages of the entailment-driven privacy policy classification. Depicted paragraph is from our test-dataset and all of the outputs generated at stage 1 would undergo stage 2 to 4 separately.

Consequently, multiple research efforts have explored the possibility of providing users with more user-friendly representations of complex privacy policies. These approaches range from presenting user-friendly labels [131], to designing chatbots to answer privacy-related questions [14]. However, most of the existing work has leveraged classical Natural Language Processing (NLP) techniques and encoder-only language models, such as variants of the BERT model.

Recent advancements in Large Language Models (LLMs), such as GPT [112, 114] and LLaMA [132], have established them as the state-of-the-art for a majority of NLP tasks. These models have demonstrated excellent capabilities in areas like text summarisation and understanding. Moreover, the versatility of LLMs has been showcased in various application domains, including medicine [133], finance [134], and others. These developments in LLMs provide a foundation for building novel solutions that can extract useful information from otherwise complex and nearly unreadable privacy policies, and present it to users in a more user-friendly manner.

In this chapter, we propose a novel entailment-driven LLM-based framework for privacy policy paragraph classification. Traditional LLM approaches are known to suffer from well-known hallucination problems and thus may not always generate the expected result directly (e.g., summarised text, classification label of text). As a result, the onus of determining whether

or not the LLM has made up or dropped facts will be on the users. One key idea behind our approach is to bolster LLM-based classification frameworks with an additional "entailment" phase to filter out the initial classifications by LLMs in an analogous way to how we would select or drop a particular LLM-generated output. Figure 3.1 demonstrates this phenomenon with an example. At stage 1, an *explained classifier* predicts a class output and a corresponding reason for a given privacy text. Then, we mask the reason from the original text and use an intermediate stage 2 with a *blank filler* in an attempt to predict the reasoning text again. An *entailment verifier* receives information from stages 1 and 2 both and decides 'entailment', i.e., the class prediction and the original reasoning are acceptable or vice versa. More specifically,

- We propose an entailment-driven LLM-based framework to classify paragraphs in privacy policies into 12 categories, such as *first party collection/ use*, *third party sharing/ collection*, and *user choice/ control*, that are easier to understand by the users. The key components of our framework include the explained classifier that generates classification thoughts, the blank filler that re-thinks about these original thoughts and the entailment verifier that makes the final decision (analogous to how a human would reason).
- We evaluate the performance of our proposed method using the OPP-115 dataset and compare our results with existing baselines and zero-shot LLM settings. We find that our method performs better than vanilla LLM-methods; 8.6%, 14.5%, and 10.5% higher than the results of T5, GPT4, and LLaMA2, respectively in terms of macro-average F1 score.
- We further analyse the explainability of our method and show quantitatively that it is better than other baseline methods we compare with. Out of 57.9% of the predictions, our method generates reasoning texts that are at least 50% or more overlapping with what a legal-expert would have reasoned. Comparatively, it is only 18.3% for the best performing embedding-based model of PrivBERT.

The rest of the chapter is organised as follows: Section 3.2 discusses related work. Section 3.3 details our framework, while Section 3.4 describes the experiment setup. Results and comparisons with other baselines are provided in Section 3.5. Finally, Section 3.6 concludes the chapter.

3.2 Related Work

We review prior work relevant to this chapter, specifically examining empirical studies of privacy policies, the NLP techniques used to analyse them, and recent advancements involving large language models.

3.2.1 Empirical studies of privacy policies

Users are increasingly concerned about online privacy [135], yet empirical studies consistently show that privacy policy documents have become substantially longer over the past two decades. With median word counts ranging from 1,500 [18] to 2,500 [19], these documents take too long to read [9, 18], resulting in users making little effort to read and understand them [20]. Further, the writing and presentation of these documents often make them inaccessible [14], with the end result often being uninformed consent [15].

Recent regulatory and compliance attempts, such as the EU GDPR, have aimed to make privacy policies mandatory and more user-friendly. While these efforts have led to positive outcomes like a 4.9% increase in the availability of privacy policies [16], a longitudinal analysis by [17] finds that this has caused privacy policies to become even longer, increasing by around 25% globally. This makes the policies even more challenging to read and therefore, to understand.

3.2.2 Analysing privacy policies using NLP techniques

Relying solely on manual analysis of privacy policies or manually crafted rules, as in [41, 42], does not scale. As a result, many researchers have proposed automated methods to provide end-users with meaningful interpretations of privacy documents. Some examples include identifying relevant privacy topics for policy sections [83, 44], summarising sections [92], highlighting subsections where users can make informed choices regarding their personal information (e.g., opt-out choices [86, 97]), developing question-answering systems for policy documents [14], building user-interface tools to identify common data practices (e.g., browser extensions [81]), or detecting privacy policy inconsistencies and non-compliances [95, 17]. The majority of these works have leveraged traditional Natural Language Processing (NLP) techniques together with machine learning methods such as support vector classifiers, logistic regression models, and neural networks, with limited adoption of recent transformer-based language encoder models like BERT [108]. PrivBERT [44] is a domain adapted version of a popular encoder model RoBERTa [109] that performs better than vanilla-encoder models in classification tasks.

3.2.3 Large language models (LLMs)

The recent surge of Large Language models (LLMs) such as GPT4 [114], LLaMA2 [132] has resulted in generative AI models being adapted with a wide range of tasks, including world knowledge, commonsense, and summarisation. The performance of LLMs could be further enhanced through domain adaptation, as demonstrated by initiatives like BloombergGPT [134]

for the financial sector and Code LLaMA [136] for software development. However, such endeavors are resource-intensive, requiring substantial text data and computational power.

PolicyGPT [47] is a recent attempt to explore zero-shot prompting for privacy policy paragraph classification with LLMs. However, our experiments show that PolicyGPT's zero-shot performance falls short when confronted with multi-class-multi-label classification (refer Section 3.4.5).

In this work, leveraging open-source LLMs, we explore how LLMs' unique explainable capabilities can aid in better interpretations of privacy policies. State-of-the-art chain-of-thought (CoT) prompting [124], which maps non-trivial inputs and outputs via intermediate steps, mimics the human thought process by breaking down a complex task into smaller, more interpretable steps. Drawing inspiration from CoT prompting, we investigate how the inclusion of one-step reasoning; *i.e. a 'reason' shown in stage 1 of Figure 3.1*, can improve our classification accuracy and lead to more reliable and explainable outputs.

3.3 Our Framework

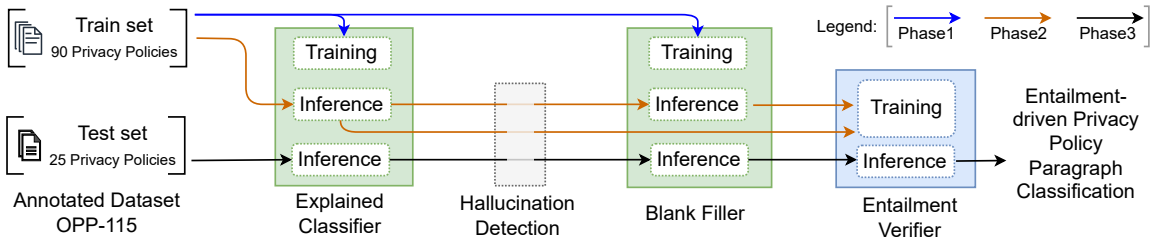

Figure 3.2: End to end pipeline of our method. Shaded in green are decoder models and shaded in blue is an encoder model. Phase 1 represents the training of explained classifier and blank filler. Phase 2 represents the training of the entailment verifier, for which we use the already trained modules from phase 1 in inference mode. Phase 3 represents all three modules working in inference mode with test dataset.

Our framework starts with explainable privacy policy paragraph classification. This is depicted in the first block of Figure 3.2 named as explained classifier. Next, an explanation-masked-out version of the privacy paragraph and previous classification output are given as inputs to the blank filler. It will then generate the most likely token sequence for that 'masked portion' by looking at the broader context of the paragraph. Finally, we check the entailment among these via the last block of our pipeline, entailment verifier. How each module works is explained below in detail.

3.3.1 Explained classifier

The explained classifier's intuition is to generate a sufficient number of class identifications with reasons that are extracted from a privacy policy paragraph. Therefore, this module requires an autoregressive LLM. Formally; an output class label $y_i \in [c_0, c_1, ..c_n]$ will be predicted alongside a subset of original tokenised text, i.e., a reason $t_i \in T$ during its training and inference stages. Here, $c_0...c_n$ are all the class labels defined in the annotated dataset. T is the tokenised privacy policy paragraph text chunk that the model is fed with. $t_i \in T$ condition is evaluated using Python Regex based hallucination detector and such filtered y_i, t_i pairs will be forwarded to the next stage of the model. For each paragraph, there could be any number of such pairs establishing a multi-label setting.

3.3.2 Blank filler

Blank filler is also an autoregressive LLM that takes paragraph T as input where a reason t_i from the explained classifier is masked out and would try to predict the best text chunk t_i' to fill in that masked part of the paragraph. We further provide the explained classifier's output class label y_i as an input indicating the model which kind of text it should try to regenerate. We explore the model's understanding of the paragraph's general landscape here; "if we mask out the main reason for a particular classification output, can the model look at the rest of the paragraph and then predict what kind of text should be there?". It is worthwhile to emphasise that we do not expect the outputs t_i' to be word-for-word identical to t_i. Instead, we expect t_i' to be similar in meaning to t_i.

3.3.3 Entailment verifier

This is an encoder-based language model attached to a neural network classifier head where we feed previously generated outputs y_i, t_i, and t_i' separated by $[SEP]$ tokens. The output of this module is binary, indicating entailment (output:1) or contradiction (output:0). This module acts as the final filter where we can remove contradictory classifications and reasons from prediction outputs for a given policy paragraph T.

We further explain how we train each of these modules using the training dataset and how we perform entitlement-driven privacy policy paragraph classifications for the testing dataset in more detail in Sec. 3.4.3.

3.4 Experimental Setup

This section provides an overview of the experimental setup, including a description of the annotated dataset we use and the train-test split we selected. It then introduces the seven baseline models against which we compare our proposed framework, followed by an outline of the evaluation criteria.

3.4.1 Modules of the framework

We used two 8-bit quantised LLaMA2 models [132] with low-rank adaptation (LoRA) [117] for our explained classifier and blank filler modules. Although our framework can accommodate any large language model, we chose LLaMA2 due to its open-source availability and its superior performance in tasks such as commonsense reasoning, world knowledge, and reading comprehension compared to the state-of-the-art at the time of its release.

For the entailment verifier, we selected a BERT encoder module with 110 million parameters that demonstrated good results for the Multi-Genre Natural Language Inference (MNLI), which is also an entailment classification task [108]. Again, we emphasise that our framework is flexible enough to incorporate any other encoder model as the entailment verifier. We fine-tune the BERT model based on the inference results from the explained classifier and blank filler (**cf.** Sec. 3.4.3).

3.4.2 OPP-115 dataset

To evaluate the performance of our framework, we used the OPP-115 dataset [82], which contains paragraph excerpts of online privacy policies annotated and labelled by legal experts. It is commonly used in comparable work [86, 95, 14, 44]. Each paragraph excerpt in the dataset (extracted from 115 web privacy policies) has a 3-tiered label. The highest level of the labelling tier is called "data practice" (e.g., *first party collection/use*), and there are ten such labels. Each high-level tier subsequently has fine-granular labels according to "data-attribute" (e.g., *personal information type*) and "data-value" (e.g., *contact*). Additionally, the paragraph excerpt also has the corresponding parts that led to specific labelling annotated. It is important to note here that one paragraph can have multiple places annotated with different labels. We show two example paragraphs from the dataset in Figure 3.3.

Comparable to other work [47, 44], the main task we solve in this dataset is given a paragraph, predicting the correct first-tier label(s). One of the ten data practice classes called *Other*, is ambiguous in that a model can not give a reason unless given a definition of what *other* means. Therefore, we augment it with the information from a lower-level "data attribute". This

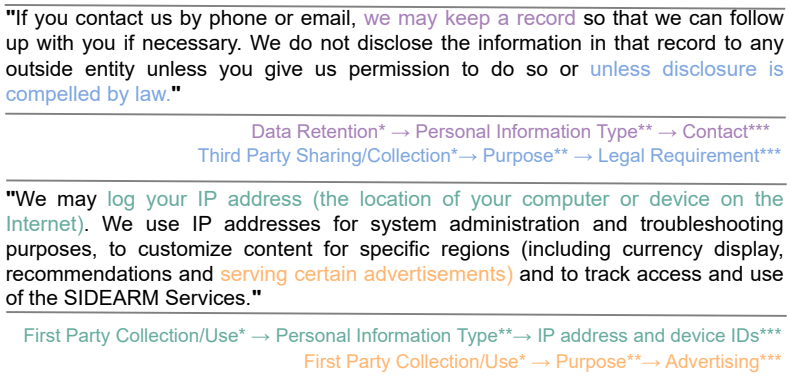

Figure 3.3: Two example paragraphs from train-set depicting the annotated segment and the relevant, data-practice label*, data-attribute label**, and data-value label***

transformation makes our final problem a 12-class classification problem. This is consistent with how other baseline methods, such as PrivBERT [44], used the OPP-115 dataset. Here, we highlight that the exact problem is a multi-class (out of 12 classes) and multi-label setting since one paragraph can contain multiple annotations with different corresponding labels. For the train-test split, out of the 115 privacy policies in the dataset, we used 2948 paragraphs associated with 90 privacy policies for training and the 683 paragraphs associated with the remaining 25 privacy policies for testing. In the training set, 54.5% paragraphs contain only 1 label, 32.1% contain 2 labels, and the remaining with 3 or more labels. Test set consists a similar distribution as well.

3.4.3 Training/testing pipeline

Initially, we train the explained classifier and blank filler separately with the training data in supervised fine-tuning setting over five epochs (Phase1 in Figure 3.2). Next, we create a labelled dataset for training the entailment verifier, according to the inputs outlined in Sec. 3.3.3 and illustrated in Phase2 of Figure 3.2. This dataset is created by running the previously fine-tuned explained classifier and blank-filler in inference mode over the training set. During this inference phase, the explained classifier generates incorrect class predictions that did not exist in the training-set. We denote these made up classes as 'contradictions'. Conversely, any accurate class prediction the explained classifier makes is marked as an 'entailment'. This approach removes the necessity of manually augmenting and curating another dataset, just for training the entailment verifier. Further, it enables us to expose the entailment verifier to realistic errors made by the previous modules, thereby facilitating its training to recognise such mistakes. Once all three modules are trained, we deploy them in inference mode, as shown in Phase3 of Figure 3.2, to evaluate performance on the held-out test set.

3.4.4 Baselines

We compare the performance of our method against seven baselines falling under two broad categories.

3.4.4.1 Embedding based classification models

These models have demonstrated effectiveness in text-classification tasks [108, 109]. Typically, embeddings extracted from a language model are processed by a linear classification head to generate the final predictions. For baselines in this category, we employ two generic encoder models, BERT and RoBERTa, then GPT2, which is adapted to the classification task by using the mean embedding representation of the token sequence of a given text input and finally, PrivBERT, which was further pre-trained on privacy policies for domain adaptation.

3.4.4.2 Language generation based classification models

The next set of baselines are auto-regressive language generation-based classifiers. In other words, these models generate the most likely classification outputs in the form of natural text. These baselines closely resemble our method as our explained classifier operates in this setting. We evaluate the performance of LLaMA2 in a supervised fine-tuning setting as the vanilla LLM, where the model outputs the class labels in natural text. Next, we adapt the encoder-decoder T5 [137] model by feeding policy text in the training set as inputs and fine-tuning it to generate target text that represents the class labels. Finally, we adopt GPT4, the state-of-the-art LLM by OpenAI at the time of experiments, as a baseline model using a prompting structure introduced by [47] to suit the twelve-class, multi-label setting and perform a zero-shot evaluation for our test dataset.

3.4.5 Evaluation metrics

As previously discussed, the challenge addressed by our framework and the relevant baselines involves the assignment of appropriate data practice labels to a given paragraph excerpt from a privacy policy, with a selection available from twelve possible categories (classes). Given that a paragraph may contain multiple categories simultaneously, our method is characterised as a multi-class-multi-label classification problem. We measure the performance of our framework and others using two types of metrics. To measure classification performance, we use the metrics of precision, recall, and F1 score. To measure the quality of explanations of our method, we use two custom metrics; normalised Levenshtein distance and overlap percentage.

(a) Embedding based classification models

	GPT2 Embedding			BERT			RoBERTa			PrivBERT		
Class	P	R	F1	P	R	F1	P	R	F1	P	R	F1
First Party Collection/Use	0.87	0.77	0.82	0.90	0.80	0.85	0.84	0.88	0.86	0.88	0.85	0.87
Third Party Sharing/Collection	0.84	0.83	0.84	0.88	0.85	0.87	0.83	0.86	0.85	0.94	0.83	0.88
User Choice/Control	0.93	0.48	0.64	0.86	0.54	0.66	0.77	0.58	0.66	0.82	0.69	0.75
User Access, Edit and Deletion	0.85	0.80	0.86	0.74	0.85	0.80	0.74	0.78	0.76	0.78	0.76	0.77
Introductory/Generic	0.75	0.56	0.64	0.69	0.53	0.60	0.76	0.44	0.56	0.73	0.69	0.71
Policy Change	0.80	0.55	0.65	0.83	0.52	0.64	0.78	0.48	0.60	0.80	0.55	0.65
Data Security	0.93	0.61	0.74	0.79	0.71	0.75	0.73	0.73	0.73	0.70	0.76	0.73
International & Specific Audience	0.90	0.78	0.84	0.84	0.85	0.84	0.83	0.87	0.85	0.88	0.88	0.88
Practice Not Covered	0.61	0.35	0.44	0.61	0.42	0.50	0.62	0.32	0.42	0.67	0.34	0.45
Data Retention	0.56	0.58	0.57	1.00	0.35	0.51	0.67	0.46	0.55	0.94	0.65	0.77
Privacy Contact Information	0.93	0.68	0.78	0.85	0.80	0.83	0.89	0.88	0.88	0.82	0.89	0.85
Do Not Track	1.00	0.60	0.75	1.00	0.20	0.33	1.00	0.80	0.89	1.00	0.60	0.75
Micro Average	0.84	0.66	0.74	0.83	0.70	0.76	0.80	0.71	0.75	0.84	0.74	**0.79**
Macro Average	0.83	0.63	0.71	0.83	0.62	0.68	0.79	0.67	0.72	0.83	0.71	**0.76**
Weighted Average	0.83	0.66	0.73	0.83	0.70	0.75	0.79	0.71	0.74	0.83	0.74	**0.78**

(b) Language generation based classification models

	T5			GPT4 Prompting			LLaMA2 7B FT			Our Method		
Class	P	R	F1	P	R	F1	P	R	F1	P	R	F1
First Party Collection/Use	0.86	0.78	0.82	0.68	0.91	0.78	0.92	0.66	0.77	0.83	0.72	0.77
Third Party Sharing/Collection	0.87	0.65	0.74	0.70	0.76	0.73	0.92	0.48	0.63	0.82	0.78	0.80
User Choice/Control	0.75	0.43	0.55	0.47	0.71	0.57	0.87	0.39	0.54	0.62	0.67	0.64
User Access, Edit and Deletion	0.86	0.61	0.71	0.47	0.80	0.59	0.83	0.49	0.62	0.67	0.70	0.68
Introductory/Generic	0.72	0.43	0.54	0.69	0.31	0.43	0.43	0.75	0.55	0.56	0.44	0.50
Policy Change	0.93	0.48	0.64	0.43	0.79	0.55	1.00	0.48	0.65	0.54	0.56	0.55
Data Security	0.89	0.52	0.65	0.52	0.77	0.62	0.94	0.48	0.64	0.83	0.69	0.75
International & Specific Audience	0.86	0.70	0.77	0.64	0.87	0.74	0.73	0.72	0.72	0.64	0.67	0.66
Practice Not Covered	0.27	0.03	0.05	0.44	0.25	0.32	0.33	0.01	0.02	0.48	0.33	0.39
Data Retention	0.75	0.12	0.20	0.38	0.46	0.41	1.00	0.19	0.32	0.59	0.50	0.54
Privacy Contact Information	1.00	0.39	0.56	0.49	0.75	0.59	1.00	0.41	0.58	0.73	0.75	0.74
Do Not Track	1.00	0.60	0.75	0.15	1.00	0.26	1.00	0.60	0.75	1.00	0.40	0.57
Micro Average	0.83	0.54	0.66	0.58	0.70	0.63	0.77	0.50	0.61	0.72	0.64	**0.68**
Macro Average	0.81	0.48	0.58	0.50	0.70	0.55	0.83	0.47	0.57	0.69	0.60	**0.63**
Weighted Average	0.79	0.54	0.62	0.60	0.70	0.62	0.80	0.50	0.58	0.71	0.64	**0.67**

*Support for each class from top to bottom; 289, 204, 115, 41, 118, 29, 62, 60, 110, 26, 56, 5

Table 3.1: Performance Comparison

3.4.5.1 Precision, recall and F1 score

In multi-class settings, precision (P), recall (R), and F1 Scores are usually reported as micro, macro, and weighted averages. In *micro-averaging*, the average is calculated globally by counting the total true positives and false positives across all classes, whereas in *macro-averaging*, the average of class-wise performance is calculated. In other words, in macro-averaging, each class, including the minority classes, contributes equally to the final number. The *weighted average* is calculated by taking the performance metric of each class, multiplying it by the number of true instances of that class (i.e., the support), and then dividing it by the total number of instances across all classes. This method provides a way to account for the frequency of each class in the dataset when calculating the overall precision. *When we present our results, we include all these averages for completeness and easy comparison with previous work, but we describe the results in terms of macro averages. This is because the macro-average is the most challenging out of these since, to have a higher value, the classifier needs to perform well in minority classes with fewer samples as well.*

Finally, we also highlight that we consider each annotation and its label as one data point. That is, to obtain a 100% recall and 100% precision for a paragraph, a model must get all labels correctly for all the annotations without producing any additional labels or missing any existing labels.

3.4.5.2 Explainability

Our model's unique approach to producing class labels at the explained classifier stage involves following the idea of one-step chain-of-thought reasoning. This process answers the *explanatory linguistic question of 'why?'* a particular class label is relevant by identifying the most suitable text chunk from the given paragraph. As the OPP-115 dataset contains annotations by legal experts that justify the assigned labels, we leverage these annotations to evaluate the explainability of our framework. To measure the semantic alignment and overlap between the justifications (reasons) generated by our model and the legal annotations, we employ two metrics:

i) Normalised Levenshtein Distance measures the character- level similarity between two strings. Formally, *Levenshtein Distance* is the minimum number of modifications required to convert one string to another using the operations of insertion, deletion, and substitution. In our case, we use it to measure the distance between the reason produced by our method and the ground-truth legal annotation. We normalise it by dividing it by the maximum length of either the reason or the annotation.

ii) Overlap Percentage measures the word-level similarity between the reason and annotation.

We calculate overlap based on *Jaccard similarity* that evaluates the intersection over the union of words present in the two text pairs. A perfectly aligned and legal expert-level-like output from our method would have a zero normalised Levenshtein distance and a 100% overlap with the legal annotation text and vice versa.

While the language generation-based classification models can be evaluated using the above two metrics, the embedding-based models, such as PrivBERT by design, are black-box and, as such, do not provide explanations for their predictions. To this end, we use the popular framework of LIME [45].

LIME: Local Interpretable Model-agnostic Explanations is a framework we can use to identify 'the most important words' that are driving the classification output, and therefore, these LIME words are effectively treated as 'explaining' the model prediction. The LIME technique iteratively perturbs the original text sequence and feeds the perturbed versions to the black-box model, observing how these perturbations positively or negatively influence the output prediction class. We can then quantitatively compare those selected LIME words with legal annotations to understand the overlap percentage. However, there are two inherent limitations to using LIME with an embedding-based black box model.

- LIME being designed for predictive models, the words it highlights do not always belong to a continuous block in the input text. Therefore, we only consider the word-to-word overlap percentage. The Levenshtein distance, being a character-level operation, is not applicable here.
- LIME performs perturbations while observing the effect of the prediction of a single class prediction output's softmax score. Therefore, we can not obtain explanations for a multi-label setting. Instead, we only obtain explanations for the model's most confident class output governed by the highest softmax value over all class labels.

3.5 Results

Next, we present our results, followed by an ablation study to highlight how different components of our overall framework contribute to the final performance. Later, we present the findings on the explainability of predictions.

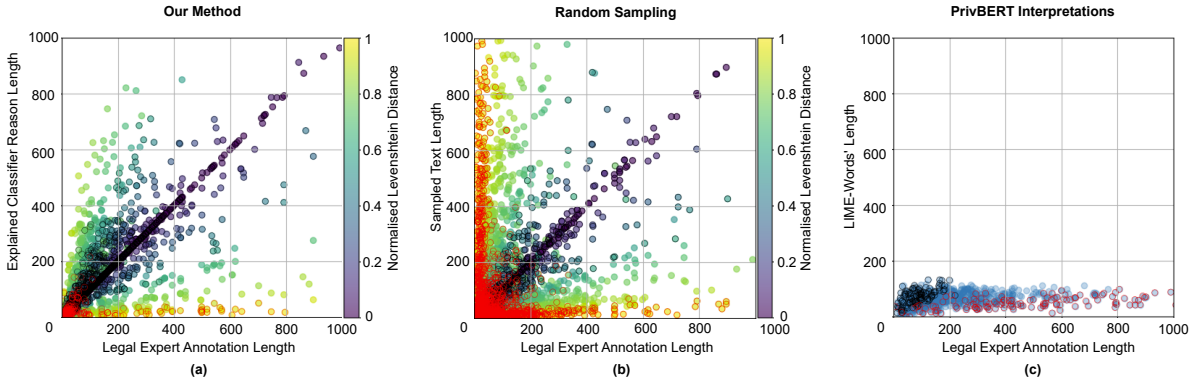

Figure 3.4: Explainability visualised: (a): our method's generated, (b): randomly sampled, (c): LIME based PrivBERT interpretations, compared with legal expert annotations. *Subfigure (c) is not colour mapped. **All sub-figures: outlined in red colour are the samples that have less than 10% overlap with legal expert annotations. Outlined in black are the samples having more than 50% overlap.

3.5.1 Performance comparison

In Table 3.1, we present class-wise Precision (P), Recall (R), and F1 Scores and their micro, macro, and weighted averages. Although the Table presents results for all metrics, our discussions mainly focus on macro averages because they serve as a representative measure of the overall performance of the models, as discussed in Section IV-E.

Embedding based classification: As can be seen from Table 3.1 (a), among the embedding-based classification methods which are not domain fine-tuned (i.e., GPT2 Embeddings, BERT, and RoBERTa), the macro F1 scores are similar with BERT having the lowest score. This is because BERT performs relatively worse in the absolute minority class *"Do Not Track"* with only an F1 score of 0.33. The slightly higher performance of RoBERTa over BERT can be attributed to it being pre-trained on a much larger dataset. Also RoBERTa's slightly higher performance over GPT2 embeddings can be attributed to it being an encoder model rather than a decoder model. It is known in the literature that encoder models perform better than decoder models in text classification tasks [138].

PrivBERT, the privacy policy domain adapted model of RoBERTa, performs well with the highest R value of 0.71 while retaining a high P value of 0.83. Therefore, the resulting F1 score of 0.76 indicates that further pre-training has indeed helped for better privacy policy paragraph classification. However, we should also note that it is only a 5.5% improvement of F1 score compared to RoBERTa. We also observe that it is struggling to recall the class *"practice not covered"*. We also note that our PrivBERT results are lower than those reported by the authors [44]; in this chapter, we record the best results we could reproduce with our train-test split.

Description	P	R	F1
Explained classifier only	0.38	0.85	0.48
Explained classifier + entailment verifier	0.61	0.61	0.59
Full pipeline	0.69	0.60	0.63

Table 3.2: Ablation Study: All values are macro-average scores

Language generation based classification: As can be seen from Table 3.1 (b), the macro averaged F1 scores for T5, GPT4, and LlaMA2 are 0.58, 0.55, and 0.57, respectively. Compared to those, our framework has a significantly better performance with a macro-averaged F1 score of 0.63 (i.e., 8.6%, 14.5%, and 10.5% higher than the original results of T5, GPT4, and LLaMA2), indicating the effectiveness of our proposed framework. While our method does not reach the performance levels of embedding-based methods, our method has the advantage of explaining the classification results (**cf.** Sec. 3.5.3).

We highlight that we tried to replicate GPT4 zero-shot results outlined in the recent preprint PolicyGPT [47] using the same dataset. Despite multiple attempts and communications with the authors, we could not reproduce the results presented in that paper. We believe that authors may have had some pre-filtering of data and different evaluation metrics (e.g., considering a classification as successful even if one predicted label is true among multiple annotation labels per paragraph - in contrast, our evaluation is more rigorous as explained in Sec. 3.4.5.1), making the exact problem they address different from ours and other comparable baselines, such as PrivBERT.

3.5.2 Ablation study

To provide further evidence on the overall effectiveness of our framework and how contributions from individual components of our framework work together, we conducted an ablation study. That is, we evaluate the performance of our framework by progressively adding modules starting from the explained classifier. We show the results in Table 3.2.

The explained classifier could be considered a 'thought generator' where, for each paragraph, it tries to generate multiple 'class output and reason' pairs until the token generation limit is exhausted. In that process, it recalls many of the correct pairs (0.85); however, some of them are not accurate, as indicated by the low precision of 0.38. As soon as we train an entailment verifier to filter out incorrect outputs, our precision improves to 0.61. Nonetheless, it decreases the recall because some correct classifications are filtered, specifically those about which the entailment verifier is not confident. As we employ the full pipeline, including blank filler, we obtain the highest macro-averaged precision of 0.69. Finally, we point out that even

Table 3.3: Overlap percentage with legal expert annotations

Overlap (%)	Our Method	Random sampling	PrivBERT
50 - 100	**57.9**	16.2	18.3
10 - 50	**33.3**	37.9	62.9
less than 10	**8.8**	45.9	18.8

*57.9 indicates that 57.9% samples of our method's predictions overlap with legal-expert annotations by 50 to 100%.

without the blank filler, our model's macro-averaged F1 score is higher than zero-shot GPT4 and LLaMA2.

3.5.3 Explainability

Next, we compare the explainability provided by our method (i.e., best-performing language generation-based method) and PrivBERT (best-performing embedding-based method) using the metrics described in Sec. 3.4.5.2. More specifically, we use normalised Levenshtein distance and overlap percentage for our method and LIME-based overlap percentage for PrivBERT. We report the results for our held-out test set.

First, we present Levenshtein distance results in Figure 3.4 (a) and (b). Each scatter dot represents a data point in our test set. The x-axis represents the legal annotation's character length, while the y-axis represents the generated reason's character length. Each data point is coloured according to the normalised Levenshtein distance between the two texts. A diagonal datapoint with 0 distance indicates a perfect prediction similar to *'what a legal expert would have annotated'*.

We show the results of our method in Figure 3.4 (a). For comparison, in Figure 3.4 (b) we present the same result for a random text generation baseline. That is, we run a separate experiment where, for each prediction, we randomly sample a text from the same paragraph and assume it as the reason generated by the model. This random sampling is done according to the annotation length to paragraph length ratio distribution of the training dataset to mimic a realistic sampling process.

We observe from the results that the generated reason distribution of our method is more positively correlated with the legal expert annotation distribution, unlike a randomly selected sample (from the same paragraph) distribution. In our case, most of the points are around or in the direction of the diagonal. In contrast, in the random case, there are more samples spread near the x-axis and y-axis, indicating significant differences between the two texts. In the figure, we outline all the samples with little or no overlap with the legal expert annotation despite some having a low normalised Levenshtein distance between them, in red. As can be seen, our method

has significantly fewer such points. We further quantitatively analyse overlap percentages later in this section.

Observing the length of the PrivBERT's LIME-words to legal expert annotation length distribution in Figure 3.4 (c), we can visually identify some drawbacks with embedding models. First, LIME can only interpret the explainability of PrivBERT's most confident output; therefore, the number of samples we can analyse is lower. Next, LIME being designed for predictive models, the words it identifies may not necessarily belong to a continuous block, and it can only consider a certain number of perturbations in a selected paragraph. Therefore, the samples are distributed more along the direction of the x-axis with LIME-words' length capped at $\sim$150 characters. As we do not consider normalised Levenshtein distance in this figure, darker shades of blue only represent densely packed sample points. Outlined in red represents the same meaning as with subfigures (a) and (b).

To quantitatively analyse the explainability, in Table 3.3, we present the results for overlap percentages. We observe that with our method, 57.9% of the predictions have at least 50% overlap with legal annotations. However, in contrast to that, nearly 45.9% of randomly selected text had less than 10% overlap with the legally annotated text (these samples with less than 10% overlap are outlined in red colour in all sub-figures in Figure 3.4). These results show that our method's explainability pre-dominantly overlaps with legal annotations and quantitatively, there is at least a 10% overlap for more than 90% of data samples with our method.

When we consider LIME-word based overlap percentage for PrivBERT, only around 18.3% of LIME words overlap 50% or more with the legal annotations and even for random sampling, this overlap count was 16.2%. Also, a similar percentage (18.8%) of samples had less than 10% of overlap. From qualitative observations, we further identified that most LIME words concentrate with class-specific words such as "third" for *"third party sharing/collection"*. This concludes that even when looking at the most confident output of PrivBERT, its quantified explainability is really low compared to our method.

3.6 Conclusion

We proposed an entailment-driven LLM-based framework for privacy policy paragraph classification and for providing explanations behind those predictions. Our training pipeline consists of an explained classifier, blank filler, and an entailment verifier that outperformed other language generation-based baselines such as T5, GPT4, and LLaMA2 by $\sim$8%–14%. The key reason for this is that our framework, inspired by one-step Chain of Thoughts (CoTs) reasoning, avoids the commonplace hallucination problem of LLMs by providing a reason for each classification label. Using the blank filler that re-predicts the same reasoning that subsequently

undergoes the entailment verification process, our method can filter such hallucinated outcomes effectively. As a result, our model has a macro-average precision increase of 38% compared to GPT4. Though the proposed framework does not achieve the performance levels of embedding-based models such as PrivBERT, it provides explanations behind the label predictions, which is useful in the context of privacy and usability. To this end, we showed that our method generates reasoning texts that are likely to be at least 50% or more overlapping with what a legal expert would have reasoned. Overall, our results show that while LLMs can be useful for providing more user-friendlier means to access privacy policies, they are not that useful in their vanilla form. Rather, it is necessary to have auxiliary steps as we proposed in our framework. Building on this foundation for online privacy policies, the next chapter introduces a systematic language modelling framework, *PrivPRISM*, for automatically analysing mobile app privacy policies and detecting discrepancies with Google Play data safety declarations.

Chapter 4

PrivPRISM: Automatically Detecting Discrepancies Between Google Play Data Safety Declarations and Developer Privacy Policies

End-users seldom read verbose privacy policies, leading app stores like Google Play to mandate simplified data safety declarations as a user-friendly alternative. However, these self-declared disclosures often contradict the full privacy policies, deceiving users about actual data practices and violating regulatory requirements for consistency. To address this, we introduce PrivPRISM, a robust framework that combines encoder and decoder language models to systematically extract and compare fine-grained data practices from privacy policies and to compare against data safety declarations, enabling scalable detection of non-compliance.

Evaluating 7,770 popular mobile games uncovers discrepancies in nearly 53% of cases, rising to 61% among 1,711 widely used generic apps. Additionally, static code analysis reveals under-disclosures, with privacy policies disclosing just 66.8% of detected accesses to sensitive data like location and financial information, versus only 36.4% in data safety declarations of mobile games. Our findings expose systemic issues, including widespread reuse of generic privacy policies, vague/contradictory statements, and hidden risks in high-profile apps with 100M+ downloads, underscoring the urgent need for automated enforcement to protect user privacy, trust and platform integrity.

4.1 Introduction

In light of the limitations of traditional privacy policy–based disclosures in mobile application (app) ecosystems, as discussed in previous chapters, developers are now mandated to self-declare summarised labels of their data privacy practices, referred to as 'data safety' (DS) on Android [72] or 'app privacy' on iOS [73] (see Figure 4.1 for an illustration). These declarations offer a more user-friendly approach compared to privacy policies (PPs), which are often criticised for being lengthy and complicated [9, 18, 20]. However, they are not intended to

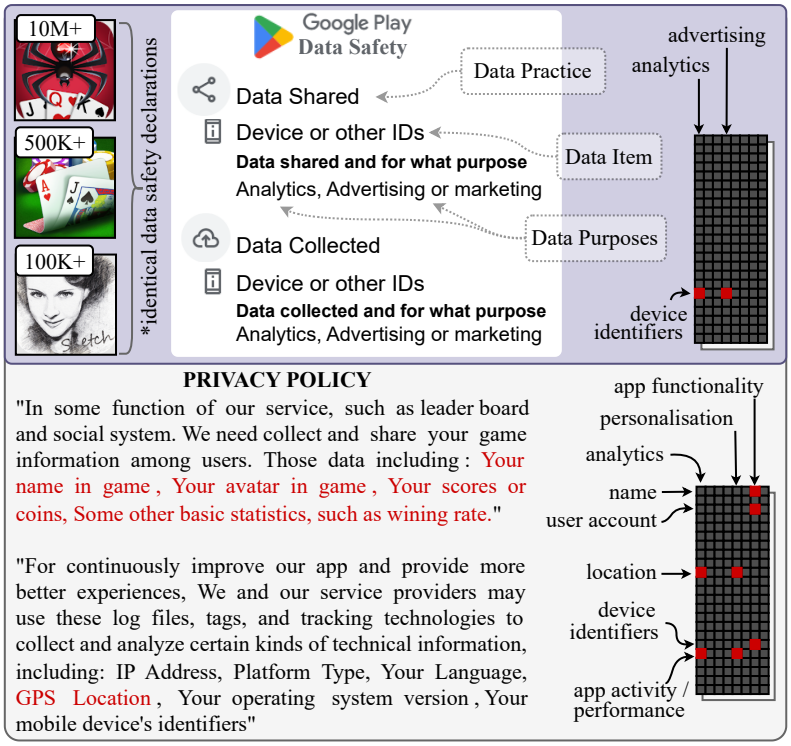

Figure 4.1: Top: Data safety declarations of three games by *Spider Solitaire Card Games* and bottom: corresponding privacy policy highlighting inconsistencies. Data safety labels only report collecting and sharing device identifiers for analytics and ads, yet their privacy policy omits ad-related details and instead disclose access to more sensitive data such as GPS location, user accounts, and app activity. *Identified via PrivPRISM framework deployment to mobile games.*

replace full privacy policies. Regulatory frameworks such as General Data Protection Regulation (GDPR) for EU, California Consumer Privacy Act (CCPA) for US and Australian Privacy Principles (APPs), along with platform operators, require compliance across both forms of disclosure. Despite this, a small-scale manual investigation by the Mozilla Foundation involving 40 apps revealed significant discrepancies between PP and DS declarations in nearly 40% of cases [139]. Given the vast number of apps on platforms like the Google Play Store, a critical question arises: *To what extent do such inconsistencies persist at scale between PP and DS declarations—both self-declared by developers—and what implications do they have for end-user privacy and regulatory compliance?*

Regulatory enforcement against non-compliant apps is typically reactive, triggered by end-user complaints, which highlights the need for automated compliance verification. However, existing methods suffer from two major shortcomings. First, they treat privacy policies and data-safety declarations as independent rather than complementary forms of developer self-attestation and lack metrics for comparing consistency across the two. Second, they depend entirely on either embedding-based or autoregressive language models for text analysis, limiting the frameworks with disadvantages of each category. For example, it is well known that

semantic search on embedding based models is noisy [140] while the precision on decoder based information retrieval is comparatively lower [141].

To address this, we introduce PrivPRISM (Privacy Policy Reasoning and Investigation using Systematic language-Modelling), a novel framework that leverages both encoder and decoder based language models for fine-grained extraction and verification of data practices from privacy policies. Our major contributions include;

- **PrivPRISM framework for robust compliance checks:** Benchmarking demonstrates PrivPRISM's superior performance, achieving a 6 percentage points higher precision than state-of-the-art GPT baselines in data practice classification and reducing mapping errors by 22.3 percentage points.
- **Large-scale empirical analysis:** We apply PrivPRISM to 7,770 popular mobile games on the Google Play Store, comparing extracted PP data practices against DS declarations using tailored compliance metrics. Our findings, including high-profile cases with 100M+ downloads, reveal serious shortcomings in current self-disclosure practices. Fig. 4.1 portrays an example finding where three popular games (10M+ downloads) under-declare in their DS labels. We uncover that 53% of apps exhibit such PP to DS inconsistencies (rising to 61% in our evaluation of 1,711 non-game apps), with developers often providing vague purposes, inaccessible or mismatched policies, and contradictory statements, all of which undermine user trust. *Dataset is available via* `https://github.com/NSS-USYD/PrivCORPUS/`.
- **Uncovering the disclosure gaps:** We observe that 64.9% of apps reuse privacy policies, with 13.4% PPs shared across 2-10 titles, obscuring app-specific data practices. Code-level analysis shows that while PPs account for only 66.8% of sensitive data requests such as location, financial and user-account access, DS declarations cover merely 36.4%, revealing a major compliance gap. A manual audit of 50 apps found that 38% of policy URLs require redirection before reaching the actual policy page, violating Google Play policies and we discuss case studies highlighting ambiguous and contradicting policy texts that not only fail regulatory expectations but also expose end-users to hidden risks.

The organisation of this chapter is as follows. Section 4.2 discusses the literature and Section 4.3 introduces the necessary terminology we use in the rest of the chapter along with the PrivPRISM framework. Section 4.4 highlights the tailored metrics we use to quantitatively characterise the compliance landscape of mobile apps. Section 4.5 provides the benchmarking results for the PrivPRISM where we show the robustness and effectiveness of the framework followed up by the results in Section 4.6. First we analyse the results for 7,770 mobile games

and next we generalise PrivPRISM framework by applying to 1,711 non-game (generic) apps. This section also elaborates an in-the-wild analysis with several selected case studies among popular developers. We conclude the chapter in Section 4.7 and additionally redirect readers to Appendix for supplementary details.

4.2 Related Work

This section reviews prior work relevant to this chapter, focusing first on studies of privacy labels in mobile app markets, and then on advances in natural language processing techniques for understanding and analysing privacy policies.

4.2.1 Privacy labels in app markets

Privacy policy contradictions, non-disclosures, and compliance gaps were well researched [41, 95, 17] before the introduction of iOS App Privacy labels in 2020, which aimed for a user-friendlier disclosure label concept. Following this, new challenges emerged and survey-based studies by [59] and [142] revealed developer misunderstandings and usability issues, while [28] found violations of apps transmitting data without proper disclosure in app privacy labels. Android also launched Data Safety declaration labels for its apps in 2022. Not long after, user surveys, app developer communications and static code analysis started to highlight under-reporting data practices and inconsistencies of the data safety sections against actual behaviour [29, 30, 31, 32, 33, 34]. Relying purely on privacy labels and actual app behaviour similar to the previous work could be inconclusive. In contrast, we motivate our work by treating data safety declarations and privacy policies, which mandatorily elaborate on such labels, as complementary forms of developer self-declarations, aiming to detect and characterise inconsistencies between them. Additionally, we compare these with evidence from app source codes to further elaborate on our findings providing a more comprehensive analysis.

4.2.2 NLP for policy understanding

Traditional NLP techniques have long been applied to simplify privacy policies through summarisation [92], to support user decision-making [86, 97], answer user queries [14], and to build user-interactive privacy tools [81, 90]. The adoption of language models further advanced this space, with encoder-based architectures such as PrivBERT by [44] and related transformer models by Adhikari et al. [43] achieving state-of-the-art performance in tasks of data practice classification and question answering. More recently, generative models such as GPT and open-source counterparts have demonstrated competitive zero- and few-shot performance on

these tasks [47, 46, 141, 96]. LLM-powered tools including CLEAR [48], PRISMe [98] and Privacify [93] demonstrate the potential of generative models to enhance end-user understanding of privacy policies and raise awareness of associated risks.

Despite substantial progress in privacy-policy analysis, most existing tools are not designed for mobile-app–specific information extraction—an important gap given the dominance of mobile platforms in global digital activity; Android alone holding 43.1% of the OS market share [54]. Prior work also do not examine how encoder and decoder models can be jointly leveraged to achieve fine-grained, semantically coherent interpretations of data practices. In contrast, our method uses a systematic encoder–decoder architecture to generate structured, semantically rich policy representations and detect inconsistencies with Google Play Data Safety declarations. It is integrated into a mobile-app–focused analysis framework that supports scalable, fully automated compliance assessment.

4.3 PrivPRISM Framework

We adopt terminology consistent with Google's definitions, a *Data practice* is an activity involving data collection or sharing by a developer, a *Data item* is a specific piece of information about the user (e.g., name, location, email) or the device (e.g., device ID, diagnostics) and a *Data purpose* is the intended reason for processing a data item, such as analytics, or advertising.

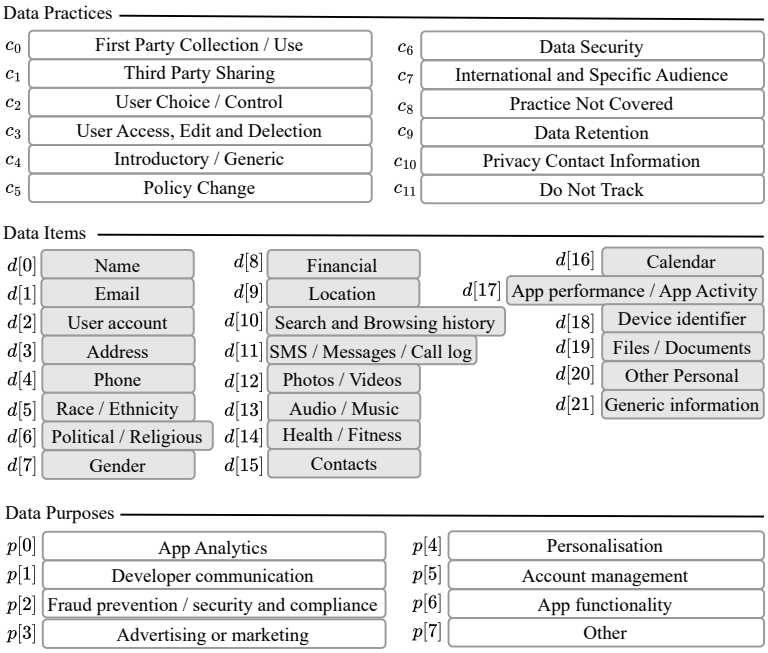

Figure 4.2: Terminology used in PrivPRISM

Figure 4.2 summarises the terminology we use throughout this paper and Figures 4.1 and 4.3

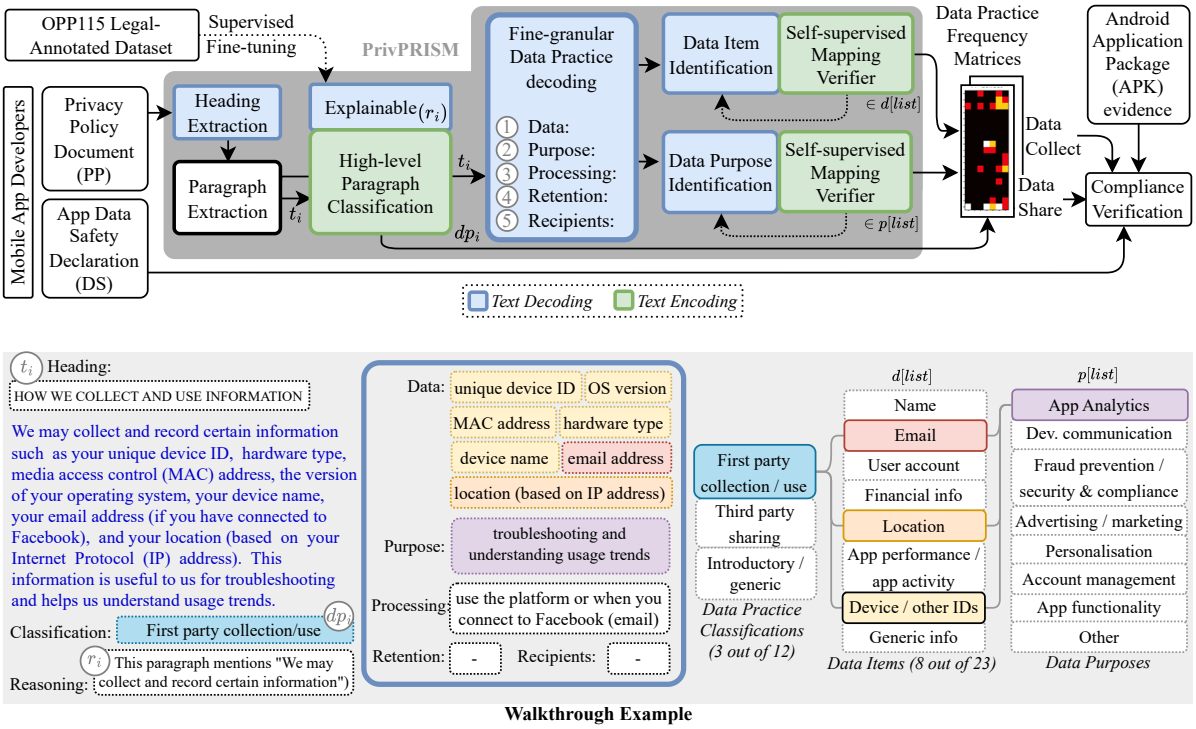

Figure 4.3: End-to-end pipeline of our framework (top) and a walk-through example (bottom)

illustrate how these definitions apply to data safety and privacy policy through examples. We redirect the readers to Appendix Section A.1 for further insights on the terminology we use. PrivPRISM, as illustrated in Figure 4.3, is a systematic and verifiable privacy policy information extraction framework consisting of encoder and decoder models. We next describe various modules of PrivPRISM.

4.3.1 Heading and paragraph extraction

Section headings in a privacy policy, as defined by the developers, are essential structural cues, often summarising the content of the corresponding sections that follow. However, extracting these headings is non-trivial due to the lack of standardised HTML formatting across developers [75, 14]. To address this, we employ a generative language model (LlaMA3.1-8B-Instruct, 128k-token context window, zero-shot setting), prompting it with the textual body extracted from downloaded HTML file to identify and extract primary section headings.

Following heading extraction, we segment each policy into sections bounded by consecutive heading pairs. To ensure robust separation of main sections we run three independent heading extraction trials per policy and select the one with the highest mean and lowest standard deviation in section lengths. Sections may contain multiple paragraphs, separated by newline; short paragraphs (< 512 characters, ~ 64 words) are merged with the following paragraphs. These

merged paragraph texts (t_i in Figure 4.3) often consolidate subheadings with their accompanying content, yielding more coherency.

4.3.2 Explained data practice classification

This module classifies a given paragraph t_i into a data practice category, such as *data collection, data sharing, other, etc.* OPP-115 [82], a popular dataset annotated by legal experts, is used as a benchmark for privacy policy paragraph classification [14] and using of fine-tuned encoder-based language models has already shown promising results [44] as well as PolicyGPT [47] that attempted this task using zero-shot GPT prompting. Our in-depth findings in Chapter 3 emphasises that leveraging decoder models can introduce "explainability" to this classification task in close resemblance with legal-expert level annotations. In order to reach the accuracy levels of encoder-based models and to filter out the most confident class label output, we propose a modified version of this idea, to first predict the most confident label $dp_i \in [c_0, c_1, ..c_{11}]$ for a paragraph t_i via an encoder model (PrivBERT) and next to use a decoder model (Llama3.1-8B-Instruct) to provide explanations (text excerpt $r_i \in t_i$). The textual class label output of the encoder model acts as a prior for the decoder model's next word prediction objective, and we fine-tuned both models using the OPP-115 dataset. This design, confirms that the encoder drives classification accuracy while the decoder adds interpretability. Labels c_o (first party collection) to c_{11} (do-not-track) are consistent with prior work. Benchmark results for this module are summarised in Sub Section 4.5.1.

4.3.3 Fine granular data practice decoding

Based on the terminology provided by the minimum core model in [143], we extract five elements from a selected policy paragraph t_i; *1. Data* or what is processed by a data practice operation, *2. Purpose* of the operation, *3. Processing* or the description of the operation (e.g., disclosure, query), *4. Retention*, which is a description of where the result is stored and for how long and *5. Recipients* who are the entities that can access the result of the operation.

As this task resembles answering explanatory linguistic questions, it is achieved via a decoder model. We use a Llama3.1-8B-instruct to word-by-word extract these elements from privacy paragraphs and allow empty elements, if necessary, with the exception of *data* type.

4.3.4 Data Item/Purpose Keyword Mapping

This stage identifies the most appropriate category for a given {data, purpose} pair decoded previously. We use a Llama3.1-8B-Instruct to map a batch of data items to a pre-assigned set of

23 keywords (8 frequent observations shown in Figure 4.3, with full keyword list in Figure 4.2) and to map a batch of data purposes to a pre-assigned set of eight keywords. All the keywords were selected based the terms defined in DS declarations.

We allow three generalised keywords for data-items: 'other personal information', 'generic information' and 'negatives' (i.e., d[22]: the text segment is not suitable as a data item - to filter noisy inputs decoded previously), and 'other' keyword for data-purpose.

The batched keyword mapping expects N inputs to be orderly mapped to N output keywords. $N = 20$ is selected empirically to speed up keyword mapping (i.e., due to prompt overhead of $\sim 100+$ instruction tokens in the decoding tasks is inefficient when provided with only one input to be mapped to one output) while minimising errors. The errors are twofold. First, any output $\neq N$ produces an error as the model has not produced a one-to-one mapping. Second, hallucinations can occur when the decoder makes up keywords instead of selecting from the pre-assigned list. To avoid such errors, we train an encoder-based self-supervised verifier to be trained on successfully keyword-mapped outputs. The intuition of this is to have a lightweight encoder classifier trained on the decoded outputs of a larger decoder model. An encoder guarantees that each input can be one-to-one mapped to an appropriate class label without hallucinations. We train two such models for the two keyword-mapping tasks and the training data from decoder models are synthetic in nature and are treated as pseudo-labels in self-supervised paradigm.

Based on these outputs, we create two matrices for each privacy policy, one for data collection and one for data sharing (**cf.** Figure 4.1). Each row of the matrix represents a data item keyword and each column represents a purpose. Multiple mentions of a single data-purpose pair increases the frequency. Comparable data matrices are generated for the DS and the sanitisation steps we perform for this are explained in Appendix Section A.2.

4.3.5 Dataset

We analyse 3,400 unique privacy policies linked to 7,770 top-ranked (based on download count) mobile games on Google Play, each with over 1M installs, including 174 titles exceeding 100M downloads. The dataset was constructed via a series of preprocessing steps, including language filtering, format validation, and file size constraints. We observed a policy reuse rate of 64.9% with 1,077 policies covering two to ten games each. Within the tail we observed 21 policies, each representing more than 20 games. A detailed description is available in the Appendix A.4. For the generalisation of our results to non-game apps, we analyse 1,254 unique privacy policies representing 1,711 apps.

4.4 Metrics for Compliance Quantification

In this section, we elaborate on the tailored metrics we use to quantify the compliance landscape of a given mobile app dataset. (Note: We have aimed to be consistent throughout this thesis regarding the notations we use. For example, c_0 represents the data practice category of "first party collection / use" in all technical chapters as well as in the appendix.)

4.4.1 Data practice compliance

When developers self-declare their *data practices*—i.e., any activity involving first-party data collection or third-party data sharing—in the privacy policy (PP) and in the Data Safety (DS) section, it becomes essential to assess the consistency between the two sources. To this end, we define two metrics: PP compliance and DS compliance.

PP compliance quantifies the extent to which data items listed in the DS declaration are also explicitly mentioned in the relavent PP. Formally, we define it as the proportion of declared DS data items that can be verifiably found in the PP.

$$PP\ compliance = n_{(PP \cap DS)}/n_{DS}$$

$$n_\alpha = \sum_{j \in J} f(d[j])\ where\ f(d[j]) = 1\ if\ d[j] \in \alpha\ else\ 0$$

Here, $j \in J$ indicates the data item $d[j]$ belonging to the list of data items J and $len(J) = 23$ in this experimental setup. As a given data item i could follow the high-level classification of $dp_i = c_0$ (i.e., first party data collection) or $dp_i = c_1$ (i.e., third party sharing), we calculate $PP\ compliance$ separately for each data practice classification type.

DS compliance measures the extent to which data items mentioned in the PP are also reflected in the DS declaration. Formally, it is defined as the proportion of data items identified in the privacy policy that are declared in the DS section.

$$DS\ compliance = n_{(PP \cap DS)}/n_{PP}$$

Results of the PP and DS compliance are discussed in Sub.Sec. 4.6.2.

4.4.2 Data purpose compliance

For a given $\{PP, DS\}$ pair, we could observe the purpose compliance by comparing the presence of each individual purpose k when a given data item j is agreed as collected or shared

among the pair. Therefore, we present our results per data item category and when the PP and DS do not agree on a data item, we specifically do not elaborate such results as PP compliance and DS compliance metrics already characterise them. We measure data purpose compliance as the intersection over union (IoU) of occurrences.

$$Purpose\ compliance(k, j) = N_{(PP \cap DS)}/N_{(PP \cup DS)}$$

$$N_\beta = \#\ p[k] \in \beta\ when\ d[j] \in (PP\ \cap\ DS)$$

Results of the data purpose compliance are discussed in Sub.Sec. 4.4.2. The next section benchmarks PrivPRISM framework and highlights robust performance.

4.5 Benchmarking PrivPRISM

This section presents a component-wise evaluation of the PrivPRISM framework where applicable. The analysis commences with the explainable high-level paragraph classifier, followed by a detailed assessment of the data item and purpose identification modules, including the validation of their associated self-supervised verifiers.

	PrivPRISM			F.T. Llama3.1			Z.S. GPT4o			
Class	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Support
c_0 First Party Collection / Use	0.94	0.74	0.83	0.90	0.21	0.34	0.88	0.69	0.78	289
c_1 Third Party Sharing / Collection	0.96	0.66	0.78	0.88	0.25	0.38	0.85	0.68	0.76	204
c_2 User Choice / Control	0.95	0.46	0.62	0.65	0.19	0.30	0.87	0.54	0.67	115
c_3 User Access, Edit and Deletion	0.94	0.37	0.53	0.60	0.15	0.24	0.72	0.63	0.68	41
c_4 Introductory / Generic	0.73	0.41	0.52	0.51	0.25	0.33	0.94	0.25	0.39	118
c_5 Policy Change	1.00	0.45	0.62	1.00	0.21	0.34	0.71	0.69	0.70	29
c_6 Data Security	0.92	0.55	0.69	0.79	0.24	0.37	0.86	0.60	0.70	62
c_7 International & Specific Audiences	0.91	0.72	0.80	0.89	0.28	0.43	0.80	0.73	0.77	60
c_8 Practice not covered	0.64	0.19	0.29	0.14	0.46	0.21	0.71	0.09	0.16	110
c_9 Data Retention	1.00	0.23	0.38	1.00	0.12	0.21	0.90	0.35	0.50	26
c_{10} Privacy Contact Information	0.97	0.66	0.79	0.79	0.20	0.31	0.93	0.45	0.60	56
c_{11} Do Not Track	1.00	0.60	0.75	1.00	0.60	0.75	0.83	1.00	0.91	5
$\mu\bar{c}$ Micro Average	0.91	0.56	0.69	0.41	0.24	0.31	0.85	0.54	0.66	1115
$m\bar{c}$ Macro Average	0.91	0.50	0.63	0.76	0.26	0.35	0.83	0.56	0.63	1115

Table 4.1: Data Practice Classification Results. F.T.:Fine-tuned, Z.S.:Zero Shot

4.5.1 Explainable high-level paragraph classification

We benchmark explainable paragraph classification results against fine-tuned Llama3.1 and zero-shot prompted GPT4o via API access in Table 4.1. When classifying data collection (c_o), our method has a 4 and 6 percentage points better precision than Llama3.1 and GPT4o, respectively and 8 and 11 percentage points better in third-party sharing (c_1) classification. Micro average precision for all classes showcases 6 percentage points better accuracy against GPT4o and 50 percentage points than Llama3.1 (due to poor performance of Llama in minority classes).

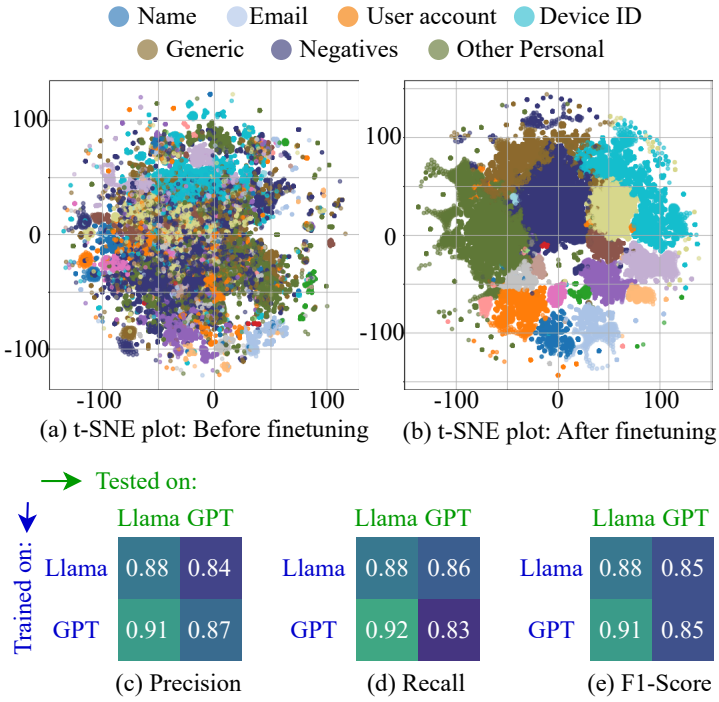

Figure 4.4: Data item mapping verifier embedding alignment and transferability

The OPP-115 dataset allows paragraphs to have multiple labels, however our method (and baselines discussed here) selects the most appropriate label, which reduces recall in benchmark results but prioritises precision for compliance tasks. To further validate the results are consistent with real world policies, we randomly (10 policies per 10KB increments of the file sizes) selected 50 policies with our framework's classification results and manually verified the outputs. We observed first party collection labels are 95.3% accurate while recalling 90.9% instances and third-party sharing labels are 88.6% accurate while recalling 95.4% instances.

With respect to the explainability, we follow the same definition as in [141], where we measure the overlap percentage of a generated reasoning text against legal expert annotations for a given paragraph. Our method recorded an average 62.85% overlap compared to 52.35% of GPT4o and 57.77% of Llama3.1.

4.5.2 Keyword mapping and self-supervised mapping verifier

We select 106,971 data items and 61,877 purposes decoded for 1,000 privacy policies to then benchmark keyword mapping and verifier training. Llama3.1 resulted in 43.7% error rate in data item one-to-one matching and 1.53% hallucinations (e.g., made up class labels such as 'cookies', 'virtual items'). We ran the same mapping experiment using GPT4o, and the error rate was 22.3% with 0.53% hallucinations. Despite GPT4o being better, both of these models struggled with this seemingly straightforward task as an element-wise classification problem. Data item self-supervised verifier is trained to produce a 23-dimensional one-hot encoded output

Class	Trained on: GPT Tested on: GPT P	R	F1	Trained on: GPT Tested on: Llama P	R	F1	Trained on: Llama Tested on: Llama P	R	F1	Trained on: Llama Tested on: GPT P	R	F1	Support
Name	0.97	0.92	0.94	0.98	0.95	0.96	0.97	0.93	0.95	0.96	0.92	0.94	199 149*
Email	0.99	0.98	0.99	0.99	0.99	0.99	0.97	0.98	0.97	0.98	0.99	0.98	347 295*
User Acct	0.95	0.80	0.87	0.96	0.67	0.79	0.84	0.69	0.75	0.93	0.83	0.88	475 436*
Financial	0.95	0.98	0.96	0.94	0.91	0.92	0.94	0.85	0.89	0.94	0.99	0.97	270 205*
Location	0.98	0.94	0.96	0.93	0.95	0.94	0.90	0.92	0.91	0.96	0.95	0.95	220 364*
App Perfo	0.90	0.93	0.91	0.85	0.95	0.90	0.91	0.78	0.84	0.81	0.97	0.88	529 402*
Device ID	0.87	0.96	0.91	0.93	0.95	0.94	0.88	0.92	0.90	0.88	0.95	0.91	744 487*
Gen. Info	0.75	0.48	0.58	0.91	0.96	0.94	0.90	0.94	0.94	0.77	0.40	0.52	1005 2972*
Micro Avg	0.87	0.83	0.85	0.91	0.92	0.91	0.88	0.88	0.88	0.84	0.86	0.85	8271 6210*

Table 4.2: Transferability of self-supervised mapping verifier training

and is trained on the training set of either Llama3.1 (53,021) or GPT4o (74,394) data belonging to 860 policies. Figure 4.4(a) and (b) are illustrative examples of how encoder embeddings belonging to similar data items are aligned with Llama3.1-based training. More importantly, by conducting a transferability evaluation, where micro average precision, recall and F1 results shown in Figure 4.4(c-e), we observed that, irrespective of the method we used to create the training set (i.e., mappings generated by GPT4o or Llama3.1), we get similar F1 scores meaning the smaller encoder has captured the keyword mapping task well. The best advantage of the verifier is the ability to deploy in an inference setting and provide accurate mappings to decoder errors, even among the training dataset (as errors were not used for training). Full transferability results are also depicted in the Table 4.2.

Class	Pr	Re	F1	Support
0 - Analytics	0.96	0.85	0.91	627
1 - Dev. Commu.	0.80	0.88	0.84	320
2 - Fraud Prevention	0.95	0.95	0.95	1530
3 - Advertising	0.97	0.95	0.96	590
4 - Personalisation	0.86	0.95	0.90	796
5 - Account Manage	0.96	0.88	0.92	847
6 - App Functionality	0.92	0.95	0.94	580
7 - Other	0.94	0.91	0.93	1030
Micro Avg	0.93	0.92	0.93	6320
Macro Avg	0.92	0.92	0.92	6320

Table 4.3: Alignment of 110M encoder model against the ground truth assumption from a 8B decoder model

Equipped with this knowledge, we performed the data purpose mapping and the respective self-supervised verifier training, with the exception of training only with Llama3.1-based mapping outputs. Initial error rate for decoder keyword mapping was 6.2% and the hallucination rate was 0.6%. We selected the training and testing datasets similar to the previous setup and referring to Table 4.5, we observed micro average 0.93 precision, 0.92 recall and 0.93 F1 score

and concluded that the finetuned verifier module's outputs are in alignment with the larger 8B parameter decoder model. Next, we used this verifier module to correct the training data errors, similar to before and deployed it in PrivPRISM.

4.6 Results

This section details the deployment of PrivPRISM "in the wild" using the dataset of 3,400 mobile game policies. We begin by analysing the structural composition of these policies and comparing their data practice disclosures against Google Play Data Safety (DS) declarations to evaluate the compliance landscape regarding data collection and sharing. Furthermore, we validate these findings by examining static evidence from Android Package (APK) files and demonstrate the framework's generalisability through its application to non-game apps. The section concludes with a manual audit of the lowest-scoring developers, highlighting specific instances of regulatory non-compliance.

4.6.1 Policy completeness

Analysing paragraph-level classifications across 3,400 policies using the *explained classifier* in the PrivPRISM framework reveals the general structural composition of privacy policies. As shown in Fig. 4.5, policies typically begin with introductory sections and end with contact information ($\sim$15%). First-party collection ($\sim$40%) and third-party sharing ($\sim$20%) dominate the core content. Completeness for these two categories remains high (91.6% and 90.3%), whereas c_8 (64.7%), c_3 (55.8%), c_9 (42.8%), and c_{11} (6.8%) show limited coverage, highlighting sparse user-data-control disclosures. This limited coverage is significant because GDPR and similar regulatory frameworks mandate clear disclosures on user data rights and control mechanisms, meaning that omissions in these categories reflect substantive compliance risks.

4.6.2 Data practice compliance

We evaluated 3,400 policies against each of their most downloaded game and we observed average compliance scores of 82.07% PP data-collect, 23.75% DS data-collect, 68.52% PP data-share, and 20.90% DS data-share. We can observe that developers tend to disclose collection practices better than sharing practices. Out of the 3,400 pairs, 46.7% of them achieved 100% PP and DS compliance scores, indicating the data items declared across them overlapped perfectly. However, note that this is irrespective of indicated purposes.

Table 4.4 showcases how the compliance scores differ according to their popularity. We could see that PP compliance is slightly better at less popular games than with the most popular

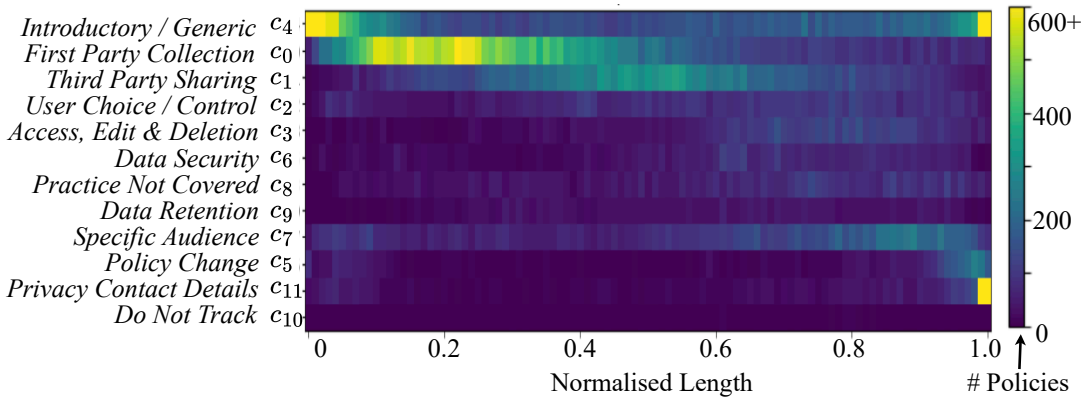

Figure 4.5: Structural composition of a privacy policy - A column represents the data practice category distribution of the policy dataset at a given normalised length.

Downloads:	$\sim 5M+$		$\sim 1-5M$	
	Collect	Share	Collect	Share
PP Compliance	81.23	68.08	83.28	69.17
DS Compliance	25.44	22.58	21.29	18.47

Table 4.4: Overall data practice compliance landscape (%)

ones. However, the DS compliance remained the other way around. We will further elaborate on this trend explaining Figure 4.7.

"One Policy to Rule them all?"; A significantly higher mean PP compliance than DS, confirms higher data disclosures in privacy policies than what is required/declared for each game app. Having a unified privacy policy for all games by a developer justifies this score, at the expense of reducing comprehension and clarity for end-users. To understand its impact, we aggregated all individual data safety declarations into a single *super-data-safety* entry per shared policy. This aggregation led to a modest DS compliance improvement of 9.26 percentage points for data collection and 6.63 for data sharing. However, many policies still disclosed more data types than actually declared in the Play Store that suggest continued misalignment.

4.6.3 Data purpose compliance

We present the data–purpose compliance scores in Table 4.5 for twelve data types. As described in Sub Section 4.6.2, a detected purpose contributes to the compliance score only when both the privacy policy (PP) and the data safety (DS) declaration agree for a given data item. Data items (i.e. rows) not shown in the table fall into categories where PP–DS agreement is limited or entirely absent. Later in the results section, we further examine the observed collection and sharing frequencies, highlighting that despite the wide range of data categories available in DS declarations, developers consistently over-declare in PPs and under-declare in DS forms,

	p[0]	p[1]	p[2]	p[3]	p[4]	p[5]	p[6]
d[0]	0.71	4.00	0.78	1.40	7.50	33.06	20.41
d[1]	2.63	17.09	6.76	1.96	1.55	50.00	11.81
d[2]	20.00	7.49	14.66	5.71	12.88	43.48	35.10
d[4]	0.00	7.69	0.00	0.00	0.00	40.00	7.14
d[8]	16.55	1.45	34.39	6.57	6.37	1.76	25.76
d[9]	26.46	2.05	7.11	7.83	9.33	1.95	17.19
d[11]	0.00	23.91	11.36	0.00	0.00	2.13	11.32
d[12]	0.00	2.17	2.13	0.00	7.50	10.81	35.29
d[15]	10.00	0.00	0.00	12.50	16.67	0.00	44.44
d[17]	47.52	1.65	7.95	8.23	7.84	0.86	35.87
d[18]	37.49	0.91	12.87	14.98	6.05	4.03	36.60
d[21]	50.98	4.76	3.85	3.23	18.60	11.11	34.48

Table 4.5: Data purpose compliance (%): Each row represents k^{th} purpose compliance score for a given valid data item category j.

resulting in limited overlap between the two.

Across all categories, developers most consistently disclose the collection and sharing of data items for *p[6] – App functionality*, indicating that functional-requirement-related purposes are the most reliably communicated. We also observe relatively clearer alignment for *d[8] – financial data* when used for *p[2] – security, compliance, and fraud prevention*, and for *d[9] – location data* when used for *p[0] – app analytics*. Additionally, *d[0–2,4] – name, email, user account, and phone data* show moderate consistency when disclosed for *p[5] – account management* across both PP and DS.

However, almost no cell exceeds a 50% compliance score, underscoring a major gap in how developers articulate why specific data types are collected or shared. For example, we can only observe 7.83% purpose compliance for *location data collected or shared for advertising*, despite this data item category observed in top-5 frequently declared in the privacy policies (cf. Figure 4.6). It is important to note that DS declarations naturally impose a fine-grained taxonomy that mobile app developers are expected to consistently reflect in their privacy policies. When such alignment is missing, users are ultimately presented with vague or overly broad descriptions of data practices, leading to blanket consenting and undermining meaningful transparency.

4.6.4 Code-level evidence based compliance

To examine how inconsistencies between PP and DS declarations manifest in actual app behaviour, we analysed ∼5,000 game Android package files (APKs). Our analysis considered two complementary aspects: (1) the explicitly declared permissions requested in the manifest files, and (2) the sensitive dataflows inferred from method calls referenced in the compiled classes, both linked to specific data items accessed by the apps. Further details are included in

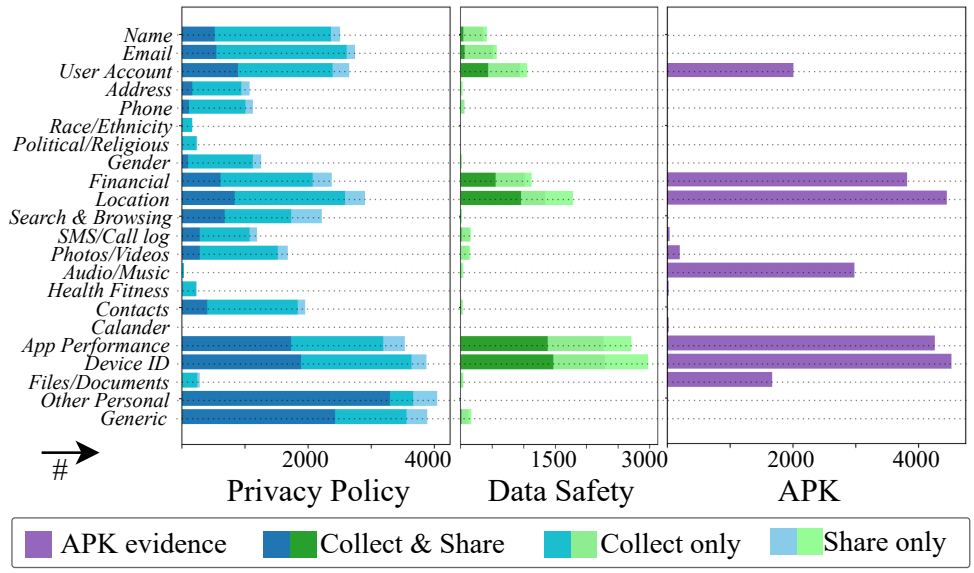

Figure 4.6: Number of data item declarations mentioned in PP, DS and within APK evidence for 4,538 game apps.

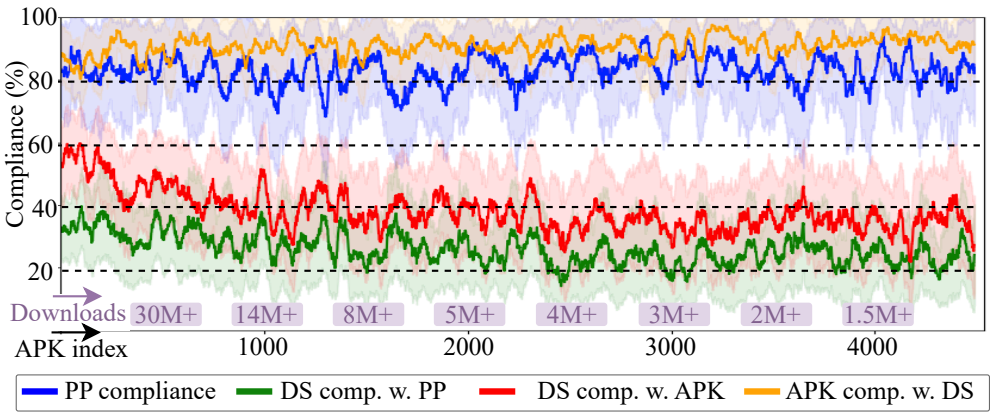

Figure 4.7: Four types of moving average filtered data item compliance scores based on the app popularity. Moving average window = 50, Shade = 0.5*STD

Appendix A.3.

Figure 4.7 and Figure 4.6 highlight three key findings. (1) APK-level evidence compliance w.r.t. DS declarations remains high and even surpasses PP compliance, suggesting that the data types disclosed in DS are generally reflected in both policy text and real app behaviour. (2) However, the converse is not observed!, with many data items declared in PPs absent in DS and not evidenced in APKs, indicating that developers tend to include broad, precautionary disclosures to secure blanket user consent and reduce future liability rather than to mirror actual data practices. (3) Most game APKs access financial (84.3%), location (98.3%), and user account (58.7%) information, yet these categories are only partially acknowledged in PPs (52.6%, 64.1%, 44.5%) and are concerningly under-represented in DS declarations (<40%). In contrast, device identifiers and analytics data show stronger alignment across PP, DS, and APKs—likely

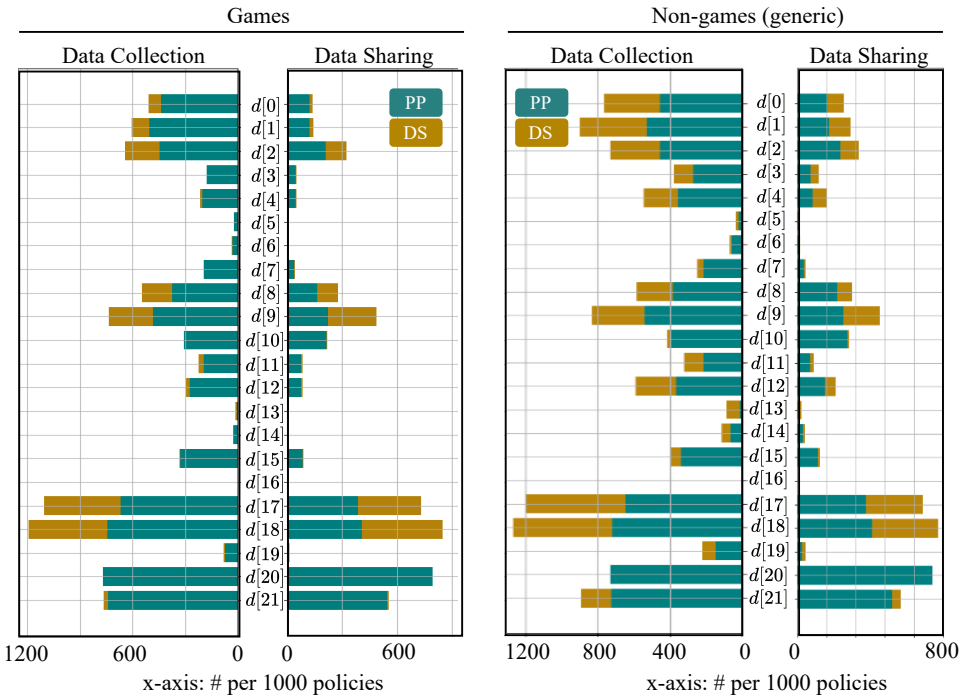

Figure 4.8: Data item collection or sharing frequency for game and non-game apps if mentioned in privacy policy (PP) or data safety (DS). X axis is normalised for easier comparison; i.e., the x-axis is normalised to show how many data items are observed per 1,000 privacy policies.

due to tighter regulatory scrutiny surrounding unique identifiers.

4.6.5 Generalisation of results for non-game apps

We conducted the main study to mobile games as they are less likely to have shared privacy policies with online versions (e.g., Meta privacy policy will cover Facebook app as well as the Facebook website) and are more likely to be downloaded more and used by minors. However, PrivPRISM pipeline is generic and can be deployed to non-game apps. To demonstrate that, we conducted following experiments.

We deployed our framework to analyse 1,254 unique privacy policies of non-game apps. Collectively, they represent 1,711 apps. Similar to the methodology explained in main text, we select the most popular (most downloaded) app for each unique privacy policy for compliance score calculation. The selection of unique privacy policies was based on the app category (e.g. Social, Communication, Tools, Medical, etc.) of their most popular non-game app and we selected 50 per each category when available.

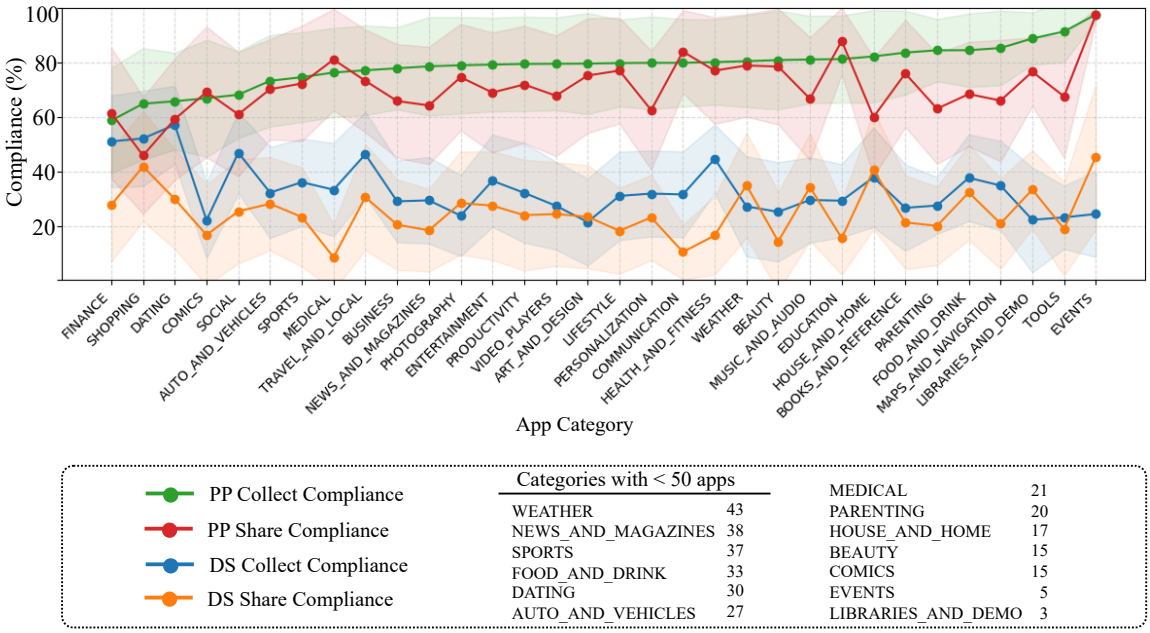

Figure 4.9: Privacy Policy and Data Safety collection and sharing compliance scores with respect to the corresponding non-game app's category. Note that several categories contained less than 50 apps when we selected the most popular non-game app corpus. Data points: mean compliance per category, Shade: 0.5*STD

4.6.5.1 Game versus non-game data practices comparison

For each PP and DS pair, our framework identifies which data items are collected and which data items are shared. While observing this for all 3,400 game related and 1,254 non-game related unique privacy policies, we observed that some data items are more frequent and some rarely come up. To better understand results, we aggregated all the data item collection/sharing results and show the findings in Figure 4.8. Please note that, even if the privacy policy contains multiple instances about a single data item, we only consider one of those instances (low-level purpose classification omitted). Therefore, a data item can at most occur with the value 3,400 for games and 1,254 for non-games. For the easiness of comparisons, we normalise the results such that we show the collection or sharing numbers per 1,000 app-policy pairs. for a given data item, we stack the total number of declarations in DS and PP both to a single box plot and are emphasised using two colours; teal for PP and dark golden for DS.

We can deduce two main findings by observing the plots. First, the collection and sharing distributions for both game and non-game apps are similar with collection numbers generally higher than the sharing numbers. Second, for both games and non-games, the disclosures obtained via PPs are significantly higher than the DS, and is consistent with the main text's finding on higher PP compliance than the DS compliance.

According to the results, DS declarations and PPs rarely declare about $d[5, 6, 13, 14, 16]$

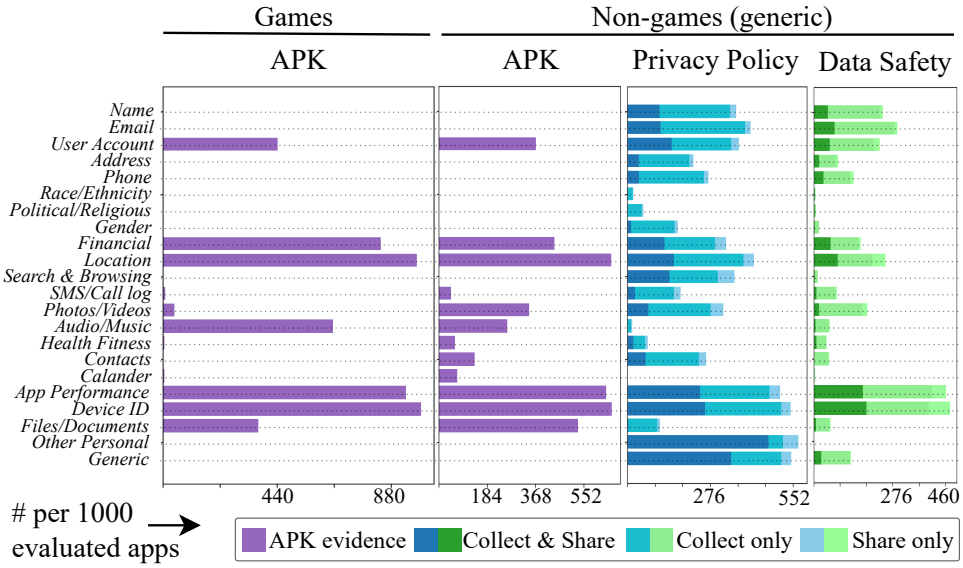

Figure 4.10: APK evidence analysis for non-game apps. Note: the x-axes are normalised for easier observations, therefore are in the same scale.

categories *(race / ethnicity / political / religious / photos / videos / audio / music / calendar).* Classes $d[3, 4, 7, 10, 12, 15]$ *(address / phone / gender / search and browsing history / SMS / Messages / Call log / contacts)* are almost never declared in data safety declarations (dark golden colour) of games but more frequently found in non-game context. This highlights that game privacy policies are not really optimised for game specific data items, rather has a tendency to follow generic privacy policy formats, proving PPs are unlikely to align with end-user comprehension.

4.6.5.2 Games versus non-games compliance score comparison

Among non-game app-policy pairs, 38.5% demonstrated perfect alignment (compared to 46.7% in games) and PP data-collect compliance score declined to 78.2% (-3.9 percentage points), whereas the PP data-share score showed slight improvement (+1.5 percentage points) relative to games. App category wise compliance scores are depicted in the Figure 4.9. Medical and communication app categories demonstrated the lowest DS share compliance scores despite high PP collection and sharing scores, where it is likely that sensitive or high-risk data items are described broadly in privacy policies. This suggests that developers in these domains may prioritise broad legal coverage over accurate disclosure, leading to misalignment between stated and actual data practices and potentially confusing end-users. Highest overall compliance scores were observed across the food and travel categories, representing apps that are frequently used by end-users (we omitted event category as the sample size was low). Lowest overall scores were observed by comics and news categories. Highest to lowest category scores observed a difference of nearly 23 percentage points.

Policy structure analysis shows that both games and non-games consistently disclose first-party collection (91%) and third-party sharing (87–90%). However, coverage of data security and retention is lower, with games lagging behind non-games (73.5% vs. 78.2% for security; 42.8% vs. 55.1% for retention).

4.6.5.3 APK evidence analysis for non-game apps

We conducted APK evidence analysis for non-game apps, following the same procedure described in Sub Section 4.6.4, with results shown in Figure 4.10. Financial and location data collection and sharing continue to show notable under-declaration patterns similar to game apps, whereas user account–related disclosures are comparatively more consistent.

In contrast, non-game apps exhibit a broader range of evidence for categories such as *photos/videos, health/fitness, contacts, calendar, and files/documents*—an expected trend given their functionality. However, these categories often lack explicit justification, especially in DS disclosures, suggesting that developers rely on generic statements rather than app-specific purposes. This highlights that while functional diversity in non-games leads to wider data access, transparency around the rationale for such access remains limited, posing user-awareness challenges.

4.6.6 When games break rules!

From a manual audit of the 50 lowest-compliance games identified by PrivPRISM we uncover six recurring issues. (1) Several developers provided mismatched (e.g. a distant game developer policy provided instead of actual game policy) or placeholder privacy policy links, violating Google Play's guidelines. (2) Popular games shared data with third parties without disclosure, while some omitted financial and performance data in their policies (e.g. in Figure 4.11(a)) (3) Policies were often ambiguous, e.g., claiming names and ages were "collected" but never left the device, and (4) obscuring the implications of collecting IP addresses for approximate location. (5) In 38% of cases, policy URLs redirected to layered sites, or inactive pages, frustrating access to the true policy. (6) Some developers reused identical policies across dozens of games under different URLs, complicating attribution and large-scale audits. Overall, every audited game showed discrepancies, with policies disclosing on average three more data items than their DS declarations, and 33% showing mismatches between developer and policy identities. We will elaborate more on specific findings in the next subsection.

4.6.7 In-depth evaluation of the manual audit

Among the 3,400 game policies we evaluated, we selected the 50 lowest privacy-compliant games (based on mean PP and DS compliant scores) and conducted a targeted manual evaluation to understand the implications. First, we have selected a developer as a case study, and we only discuss publicly available information as of Oct. 2025.

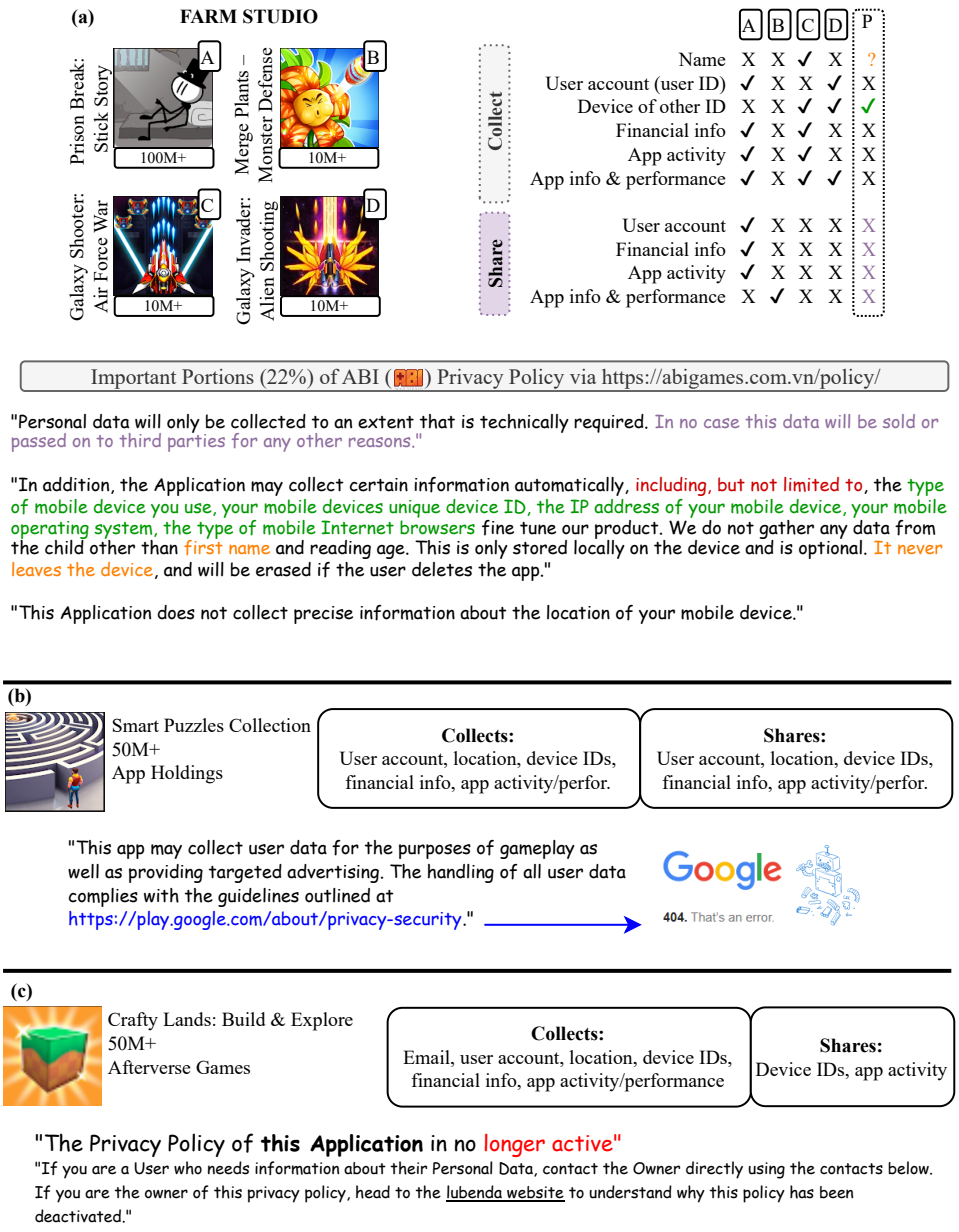

Figure 4.11: (a): A case study of 'FARM STUDIO', (b,c): Existing policy links but invalid contents. Validated 2025.10

4.6.7.1 Case study: FARM STUDIO

Figure 4.11(a) presents four games by this developer, accompanied by respective DS sections and the important portions of the PP. Game A with the highest number of downloads (100M+) resulted in 0% PP and DS compliance scores in PrivPRISM framework, and we also observed that the developer name and privacy policy names do not match; possibly due to citing game developer policy as their own. The developer's website was simply a placeholder URL showing only 'Hello World!', and their GMAIL address offered no further insight. *01. It is against Google Play guidelines to provide a mismatched policy*.

Two of the games (A,B) share data with third parties without disclosure in PP, and the remaining two games had mismatched data collection practices, with some important details missing in the policy, like financial and app performance information collection. *02. This is a data collection and sharing disclosure contradiction.*

Additionally, parts of the PP were highly ambiguous. It indicates the developer gathering first name and age, only to later claim such information never leaves the device. Aiding the contradiction, Game C indicated collecting names in the DS, raising questions about *03. highly ambiguous and contradicting policy text*. Furthermore, the PP mentions not collecting precise location but was unclear about the purpose of collecting IP addresses. *04. IP address collection via internet connections may be used to detect end-user approximate location, where many privacy policies lack clarity*.

4.6.7.2 Case study: BERNI MOBILE

With their most popular games downloaded more than 10M times, Berni Mobile's privacy policy simply states that *"Berni Mobile do not collect any personal user data"* and that *"ad networks may access your unique device identifier through their own technologies and use it to target advertising to you."* However, upon evaluating their popular title *Cruciverba Italiano* (1M+ downloads), we found that the app collects and shares device or other IDs, approximate location, app info and performance data, and app activity for various purposes. Since approximate location can constitute personal data when linked to a unique device identifier, this represents a direct policy contradiction. Nevertheless, the fact that such a popular developer maintains an overly vague privacy policy of only about 100 words highlights a concerning lack of transparency and accountability in user data handling practices.

4.6.7.3 More ways to non-comply?

Figure 4.11(b,c) portrays when policy URLs redirected users to active websites but with invalid privacy policies. The developer in (b) used a Google guideline (which is now invalid) as the privacy policy despite 50M+ downloads, and much sensitive information was collected and shared. This is a clear evasion of compliance; *05. Google Play requires developers to submit an active privacy policy link. However, this requirement can be exploited with links containing privacy policy look-alike text.* Furthermore, 38% of privacy policy links in the manual evaluation required us to read and click several other links to get to the actual privacy policy (hence the original URL contents were flagged as non-compliant with no data). Some of these contained language selections, region selections or were generic websites where users need to find true policy.

Additionally, we found a developer named 'Play Cool Zombie Sport Games' with eight games in this non-compliant list (41 active games in total) with customised and unique policy links for each of their games, only to discover after one level of redirection, lands on the same privacy policy belonging to 'TEN SQUARE GAMES'. We were unable to verify any affiliation between the developer and the policy owner. *06. Reusing identical policy content under different URLs may be a deliberate tactic to evade automated detection and complicate compliance audits at scale.*

Manual verification of the remaining 62% revealed discrepancies in every case, with privacy policies disclosing, on average, three more data items—including location and financial data—than their DS declarations. $\sim 33\%$ showed mismatched developer and policy names, and some policies denied data collection while DS declared otherwise.

4.6.8 Actionable insights

According to Google Play policy, it is mandatory for app developers to provide an active, public and non-geo fenced privacy policy URL in the app listings. Non-compliance could lead to banned apps or banned developer profiles by Google. Such attempts will be identified during three stages; (1) Google's new app publishing phase, (2) For existing apps - during the regular app updating process and (3) For existing apps without valid privacy policy links – during Google's routine ecosystem checks.

Developers could potentially avoid all three of the stages above if there is an active, public and unrestricted page linked as a privacy policy, but the content could be vague, placeholder or boilerplate. PrivPRISM becomes a valuable tool in this context;

01- Non-privacy policy related, place-holder or vague text – The PrivPRISM framework

processes all textual content from such pages and systematically evaluates them for meaningful privacy disclosures. Owing to its capacity to identify high-level data practice categories and fine-grained data items even in short text segments, PrivPRISM effectively detects these inadequate policies—typically reflected as very low PP compliance scores relative to DS declarations and static code evidence. Such cases constitute clear violations of Google Play's developer policy and applicable regional privacy regulations, and are potentially actionable upon regulatory review. Characteristic – *Exhibits low PP compliance despite high DS compliance*.

02- Boilerplate policies that are not-truly representative of app behaviour – This is one of the key artefacts highlighted in our study — the widespread reuse of generic PPs that are written to cover multiple apps or services. As a result, end-users are often overwhelmed by irrelevant or overly broad descriptions of data practices, making it difficult to discern what actually applies to a specific app. While boilerplate text does not necessarily imply non-compliance, such policies rarely align with the app's real data handling behaviours and therefore undermine transparency and meaningful consent. Policy contradictions or non-representative policies are potentially actionable under regulatory and platform rules. Characteristic – *Exhibits low DS compliance despite high PP compliance*.

4.7 Concluding Remarks

We introduce PrivPRISM, a novel framework for fine-granular extraction and analysis of data practices in mobile app privacy policies. Unlike off-the-shelf LLM approaches, PrivPRISM employs a structured pipeline, achieving 6% higher precision in explainable data practice classification compared to state-of-the-art GPT baselines. To support reliable real-world deployment, it integrates self-supervised verification modules to reduce generative errors and we define novel metrics for observing compliance. Applying PrivPRISM to PPs from $\sim 10,000$ of the most-downloaded Google Play games and generic apps ($\sim 5,000$ unique PPs), we identify potential discrepancies in 53% (games) to 61% (non-games) of cases. Code-level evidence analysis shows widespread PP and DS under-declarations among sensitive data-item categories alongside numerous other categories lacking explicit justification for collection or sharing. A manual audit of the 50 least compliant game-apps qualitatively reveals PP-DS contradictions, placeholder URLs, vague or conflicting statements, and tactics that may obscure auditability—all posing clear threats to end-user privacy.

Future work can extend our framework in several directions. First, while our analysis highlights substantial discrepancies between privacy policies, data safety declarations, and static APK evidence, apps flagged by our method could be of false positives where manual audits

supported by explainable indications from PrivPRISM are encouraged alongside a fully comprehensive assessment of dynamic analysis to observe real user interactions, runtime data flows, and network behaviours—elements that static techniques and fuzzing alone cannot reliably capture [144, 145, 146]. Second, our study assumes direct accessibility of privacy policies via developer-provided URLs; however, some policies require navigating complex website structures or hidden links, suggesting the need for more intelligent crawling methods capable of handling dynamic and nested content. However, it is noteworthy to mention that, it is mandatory for developers to provide a direct, non-geo-fenced and public facing URL during the app submission process. Finally, discrepancies may also arise when the same app appears across multiple platforms or markets, where metadata varies and disclosure taxonomies differ (e.g., Android vs. iOS). Extending PrivPRISM to reconcile cross-market versions and platform-specific disclosures would enable a more holistic compliance assessment.

Chapter 5

PrivSTRUCT: Untangling Data Purpose Compliance of Privacy Policies in Google Play Store

Existing research typically treats privacy policies as flat, uniform text, extracting information without regard for the document's logical hierarchy. This disregard for structural cues such as section headings designed to guide the reader, often leads automated methods to entangle distinct data practices, particularly when linking sensitive data items to their specific purposes. To address this, we introduce **PrivSTRUCT**, a novel and systematic encoder and decoder combined framework that leverages developer-defined structural cues to untangle complex privacy disclosures.

Benchmarking against the state-of-the-art tool PoliGraph reveals that PrivSTRUCT extracts more than double the number of data item and purpose excerpts. By applying this framework to a large-scale dataset of 3,756 Android apps, we uncover a critical transparency gap: the probability of developers overstating a data purpose is **20.4%** higher for first-party collection and **9.7%** higher for third-party sharing when they rely on globally defined purposes rather than specific, locally scoped disclosures. Alarmingly, we find that sensitive third-party data flows such as sharing financial data for analytics are frequently diluted and entangled into generic or unrelated categories, highlighting a persistent failure in the current purpose disclosure landscape.

5.1 Introduction

Privacy policies are supposed to inform users about data collection and usage practices, yet they are notoriously complex and lengthy, often leading individuals to skip reading them altogether [9, 18, 20]. Regulations such as the EU General Data Protection Regulation (GDPR), the California Consumer Privacy Act (CCPA), and the Australian Privacy Principles (APPs) have sought to improve transparency. However, these efforts have paradoxically increased policy intricacy, making them even less accessible [16, 17]. In the mobile app ecosystem, where users increasingly rely on applications for daily tasks, this issue is amplified. App marketplaces

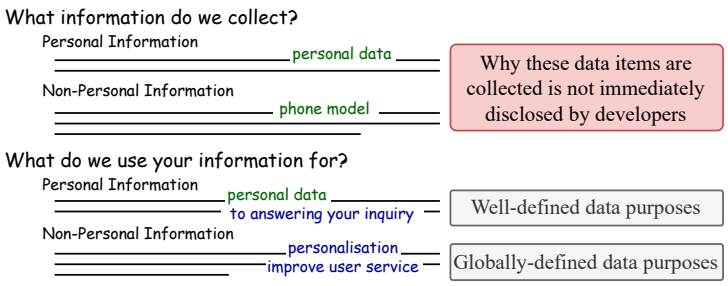

Figure 5.1: Privacy policies not-defining data item to data purpose relationships immediately, prompting end-users to globally match data items with respective purposes

like the Google Play Store require developers to provide summarised data practices through Android data safety labels [72], which outline collected or shared data items and their purposes in a straightforward manner. However, these self-declared labels must align with the legally binding full privacy policy document in order to avoid misleading users and ensure compliance [129]. Yet, as we demonstrate in this work, such alignments are frequently undermined by structural ambiguities in the policies themselves.

Empirical observations reveal that many policies separate descriptions of data items (e.g., "name" or "email address") from their purposes (e.g., "app analytics" or "personalisation"), often across distinct sections, as illustrated in Figure 5.1. This isolation can prompt users to make blanket consents, assuming broad applicability without clear linkages, potentially leading to unfair data processing. Existing natural language processing (NLP) approaches often exacerbate this issue by analysing policies as flat sequences of text, disregarding the hierarchical cues (such as section headings) that developers use to scope these claims. For instance, while frameworks like PoliGraph [49] effectively map explicit semantic relations into knowledge graphs, they often struggle to differentiate between purposes defined locally (tied structurally to specific data items) and those defined globally (broad mandates applied to the entire document). Consequently, these methods risk extracting undistinguished relationships that misrepresent the true granularity of the developer's disclosure.

To address these challenges, we propose PrivSTRUCT (Privacy policy Structural Tagging for Robust Understanding and Compliance Tracing), a novel framework that resolves policy ambiguity by systematically leveraging the developer-intended structural hierarchy. Unlike traditional flat-text analysis, PrivSTRUCT utilises a hybrid architecture combining decoder-based high-level structural extraction with encoder-based classification for granular semantic analysis. This allows us to not only map the logical flow of a policy but to specifically categorise section headings by intent, distinguishing between segments that define what data is collected, why it is processed, why it is shared, etc. By reconstructing these dependency links, PrivSTRUCT can accurately differentiate between locally defined purposes (tied to specific data items) and globally defined mandates, enabling a far more precise audit of Play Store Data Safety Labels

than previously possible. More specifically, our contributions include

- We propose PrivSTRUCT, a novel systematic framework that combines the natural language understanding (NLU) capabilities of decoder-based LLMs with the classification-efficiency of encoder-based models to analyse privacy policies. Unlike prior works that treat policies as flat text, PrivSTRUCT leverages structural cues to untangle the relationships between data items and purposes. We introduce a Direct Preference Optimisation (DPO) pipeline to fine-tune smaller, open-source models (Llama-3.1) for structural extraction, achieving performance comparable to proprietary state-of-the-art models while significantly reducing computational overhead.
- We benchmark PrivSTRUCT against *PoliGraph*, a state-of-the-art NLP tool for privacy policy analysis. Our evaluation on a diverse test set reveals that in average PoliGraph only identifies 52.1% and 89.1% less number of unique data items and data purposes compared to our method.
- We conduct a large-scale empirical analysis of 3,756 privacy policies and their corresponding Google Play Data Safety labels. We characterise the landscape of "Purpose Compliance" and "Purpose Dilution," revealing that the probability of developers overstating purposes is 20.4% higher for first-party collection and 9.7% higher for third-party sharing as they increasingly rely on globally defined purposes. Alarmingly, we find that critical third-party data flows such as sharing financial data for analytics are frequently diluted into generic or unrelated categories within the policy text, undermining the transparency goals of modern app marketplaces. *Dataset is available via* `https://github.com/NSS-USYD/PrivCORPUS/`.

The organisation of this chapter is as follows. Section 5.2 gives an overall motivation to this research idea which highlights a key finding enabling us to use developer-intended flow of the privacy policy to any fine-granular information extraction of the policy texts. Next, in Section 5.3 we introduce PrivSTRUCT framework with the aid of a walk-through example and explain the experimental setup. Section 5.4 showcases the benchmarking results that demonstrate superior performance of PrivSTRUCT to proceed to Section 5.5 that explores the main findings of PrivSTRUCT applied in the wild with real world privacy policies of Google Play Store apps.

5.2 Motivation

Providing meaningful labels to privacy policy paragraphs has been widely addressed in literature and particularly, OPP-115 foundational dataset introduced by Wilson et al. [82] where legal

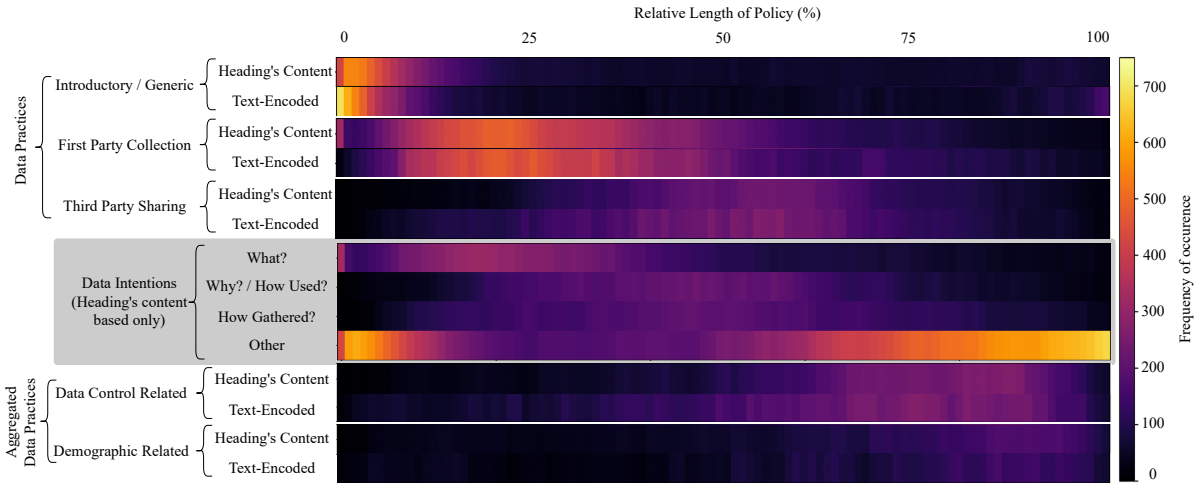

Figure 5.2: Background - Analysing 750 privacy policies based on semantic (text encoder) classification versus structural (heading-content basis) labels

experts annotate 23k segments belonging to 115 online privacy policies has become popular and widely adapted as in [86, 95, 14, 44, 141]. This introduced the *data practice* labels such as *first party collection, third party sharing, introductory or generic, user choice control, etc.* that effectively classify a given privacy paragraph to the most suitable category. PrivBERT [44] is one of the most popular encoder based models being pre-trained on millions of privacy policies and shown strong performance in OPP-115 classification task as a downstream application.

Encoder based language models rely on the embedding vectors generated for a given policy text portion (e.g. a paragraph) to generate classification labels. However, we rarely see a privacy policy written as a continuous block of text that require individual paragraphs to be classified, rather, developers use section headings to direct end-users to specific parts of the policy. Based on this, we conducted an experiment to observe *how much alignment is there if we rely of developer defined section headings to obtain the data practice flow of the policy compared to traditional paragraph classification?*.

To investigate the structural composition of privacy policies, we analyse a dataset of 750 real-world privacy policies from popular mobile apps (we discuss dataset curation in detail in Sub Section 5.3.2). As there is no existing methods in literature, and as HTML tags to identifying section boundaries is argued to be unreliable (e.g. widespread use of JavaScript classes) [75, 14], we employ GPT-5 reasoning model to reconstruct the structural hierarchy. First, we feed the entire privacy policy text as context to extract visible headings via zero-shot prompting.

Subsequently, using *only* the extracted headings, we perform a second zero-shot prompting step to assign two levels of labels: *data-practice-tags* and *developer-intention-tags*.

- **Data-practice-tags:** These correspond to standard categories discussed previously (e.g.,

First Party Collection, *Third Party Sharing*).

- **Developer-intention-tags:** These classify the intent of the heading. We evaluate whether a heading represents "what" data is collected, "why" it is collected/shared, "how" it is gathered/used, or "other".

In summary, we classify the subsequent text content based purely on the semantic signal provided by the section headings. To compare these heading-based labels, we leverage the framework from [44] to assign 12 class labels (e.g., First Party Collection) to the text content itself. We isolate text chunks (excluding the headings identified above) and classify them using a PrivBERT classifier trained on the OPP115 dataset. This allows us to compare the structural (heading-based) distribution against the semantic (content-based) distribution. Note that as no existing encoder-based dataset exists for *developer-intention-tags*, we limit this comparison to data practices.

5.2.1 Evidence of structural cues

Figure 5.2 presents the results of this structural analysis. The x-axis represents the *relative position* within the privacy policy, normalised on a scale of 0.0 (start of the document) to 1.0 (end of the document).

We compare the distribution of labels derived from text-encoding versus those derived purely from headings. We observe near-identical distributions across both methods.

- **0.0 – 0.1 (Introduction):** The initial 10% of the policy length is typically introductory or generic.
- **0.1 – 0.5 (Collection):** Approximately 40% of the document length id dedicated to *First Party Collection*. In this segment, we observed a relative agreement of 98.65% between the heading-based and text-embedding based methods.
- **0.5 – 0.75 (Sharing):** The subsequent 25% of the document predominantly details *Third Party Sharing*, with a method agreement of 91.65%.
- **0.75 – 1.0 (Data control, demographics or other):** *Data Control* practices (an aggregation of user choice, access / edit / deletion, retention, and security) and *Demographic* explanations (e.g., target regions like EU or US) are predominantly located in the final quartile.

Finally, the GPT-5 based *data-intention-tags* reveal that sections specific to "why" data is collected or "how" it is used are structurally distinct from sections describing "what" data is

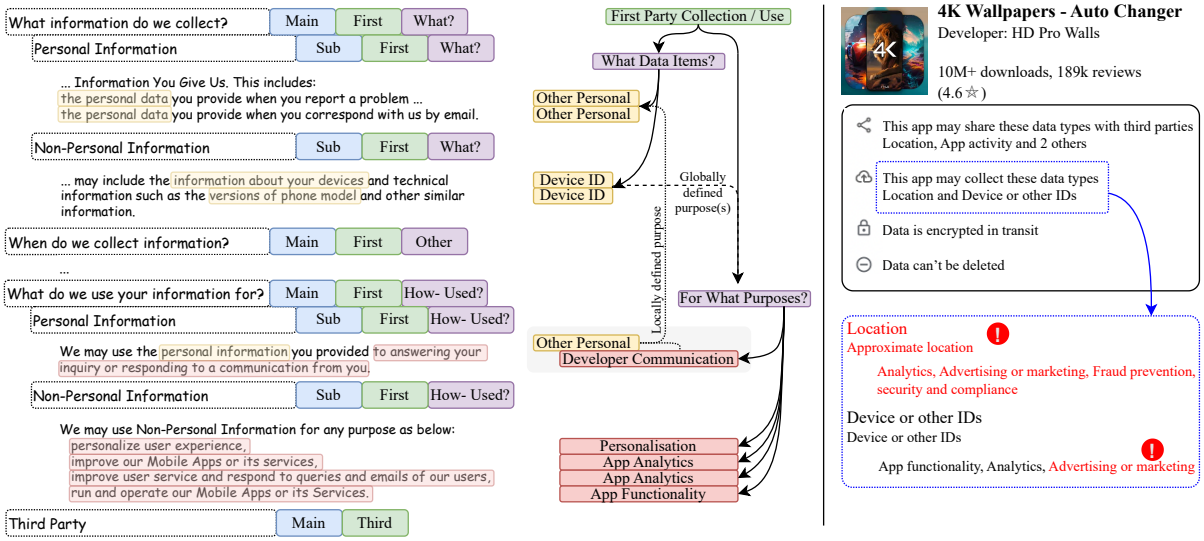

Figure 5.3: Left: Privacy policy portion of HD Pro Walls developer related to first party collection details. Right: The data safety declaration of the developer's app called 4K Wallpapers - Auto Changer, with nearly 10M+ downloads, yet contradicting with the policy itself based on data practices. Validated on December, 2025

collected. This confirms that developers implicitly structure policies with separated intents, a cue that can be leveraged for more granular analysis.

5.2.2 Pathway to PrivSTRUCT

Privacy policies, despite being time consuming and hard to comprehend, contain valuable structural cues that were designed to guide users to respective subsections of the policy. Existing methods do not utilise this knowledge and simply ignore developer-disclosed data intentions, therefore, do not have proper methods to link subsections specifically addressing data purposes. PrivSTRUCT uses these section headings and create a high level flow of the privacy policy, in a similar way how humans would navigate through the policy if they were to read it. More specifically, it allows identifying and mapping globally defined purposes with respective data practices and data items, providing more meaningful representations to the policies.

5.3 PrivSTRUCT Framework

In this section, we introduce the methodology we use to create PrivSTRUCT framework that can fine granular investigate privacy policy data practices with developer intended data items and data purposes. First we explain the methodology using a walkthrough example, and next we discuss each module of the framework.

5.3.1 Walk-through example

Figure 5.3(left) shows a privacy policy being annotated by the PrivSTRUCT framework. First, we identify the main and sub-sections present in the privacy policy text. In the figure, for simplicity, we only show the subsections related to the first party collection information (approximately 20.9% of the total words for this specific policy text). Each heading is given three tags, whether it is a main or sub heading (colour: blue), what is the *data practice* type (colour: green: 'First' indicating *first party collection*) and what is the developer intended *data intention* (colour: purple) Going through the policy text, we observe that section headings identified by PrivSTRUCT as data intention: 'what?' (is collected) do indeed describe all the data items that the developer collects (highlighted in yellow colour). Similarly, we observe the developer disclosing data purposes inside the sections identified as data intention: 'how-used?' by our framework.

Observing such purposes, the developer clearly explains they utilised *personal information: i.e., other personal* they collect *to answering your inquiry: i.e., developer communication* purpose, which we identify as a **locally-defined purpose**. However, there is no such clear definition for *information about your devices: i.e. device ID* and *versions of phone model: i.e. again device ID* and we are compelled to link the remaining data purposes *personalisation, app analytics and app functionality* with those data items. We call such purposes as **globally defined** purposes where the readers have to create an approximate connection. For each identified data items and data purposes, PrivSTRUCT creates a data item and purpose label as we show in the graph in the middle of the figure. In the right is the data safety declaration of the most popular app by this developer with nearly 10M+ downloads, which alarmingly indicates that they collect approximate location from end-users for *analytics, advertising, fraud prevention security and compliance* but this is not defined at all in the privacy policy, indicating a major contradiction when regulatory frameworks require consistency across both forms of disclosure. Additionally, use of *device IDs* for *advertising or marketing* is also not disclosed in policy as well.

This example emphasises how PrivSTRUCT can effectively identify developer disclosure and how it could be utilised for at scale app market integrity checks. Next we formally explain the modules of PrivSTRUCT and the entire pipeline is depicted in Figure 5.4.

5.3.2 Dataset

We analyse a corpus of 3,756 unique privacy policies associated with 6,540 of the most downloaded apps on the Google Play Store, crawled during the first half of 2024. To ensure tractability, we filtered for policies with a text size under 50KB (well above the median: 11KB; average:

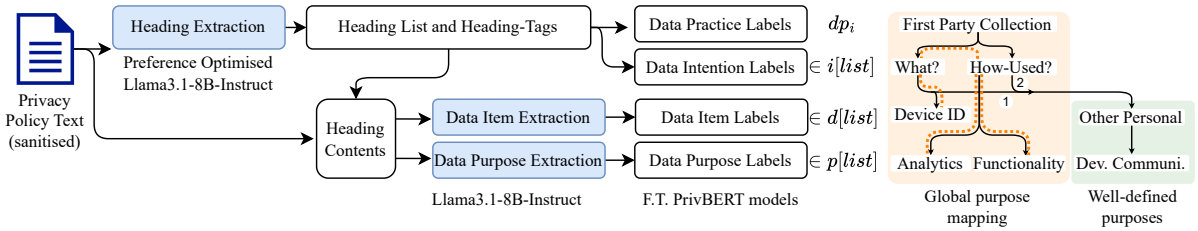

Figure 5.4: PrivSTRUCT framework

18.3KB). Our focus on top-ranked apps aligns with Khandelwal et al. [30], which suggest that developers of high-visibility apps—driven by public scrutiny and greater resources—tend to provide more complete data safety declarations.

A significant observation in this domain is the widespread pattern of policy reuse; a single developer often publishes multiple highly downloaded apps (e.g., Google) governed by a single, unified privacy policy. To attribute purpose compliance accurately, we scope our analysis by associating each unique privacy policy with the developer's single most downloaded app.

Category Distribution: The dataset is predominantly composed of the TOOLS category (20.2%), followed by PHOTOGRAPHY (8.5%) and ENTERTAINMENT (8.4%). This distribution is significant for privacy analysis: 'Tools' apps frequently require broad system permissions that are functionally necessary but often described vaguely in text. Conversely, highly sensitive categories such as FINANCE (3.3%) and MEDICAL (0.6%) constitute a smaller portion of the dataset, reflecting the specialised nature of these apps compared to general-purpose utilities.

Stratified Sub-sampling: From the total corpus, we selected a subset of 750 policies for framework development. To prevent bias toward shorter, generic policies, we employed stratified sampling to ensure a uniform distribution across policy lengths (selecting 150 policies from each 10KB increment up to 50KB). For model development, we utilised a split of 675 policies for training and the remaining 75 for testing.

5.3.3 Heading extraction

First module of PrivSTRUCT is the heading extraction itself and we compare API access based GPT-5 (400k context window) model against Llama3.1 8B Instruct (128k context window) for this particular task given the advantages of locally-hosted pricing advantage and inferencing time advantage for at-scale heading extraction from privacy policies. For the comparison we use the 750 sub-sampled privacy policy subset and we prompt both models to provide a heading list for a given policy text with a suitable `<main>` or `<sub>` tags for each heading indicating a main or sub-heading (c.f. Figure 5.5).

Despite similar performance from both models in majority of policies empirical evaluations depicted that Llama based heading extraction struggles in three instances. 1. Llama based heading extraction often opted for all main or all sub heading tags, 2. Llama headings were observed to extract entire sentences as headings and 3. dropped headings when existed. Therefore, we opted for direct preference optimisation (DPO) of the Llama3.1-8B Instruct model based on GPT5 outputs - to be used as the heading-extractor, and we use this technique due to unique advantages compared to traditional supervised fine-tuning techniques [119]. Figure 5.5 depicts the DPO dataset curation steps. For training we select 675 policies (90%) equally distributed in length with 125 policies per 10KB increment and each contains GPT-5 and Llama3.1-8B (off-the-shelf) inferencing results for heading extraction. Next we chunk down each policy based on newline characters to a maximum of 512 token portions and among such 8,418 portions, we observed 28.64% identical heading identifications among both models. We only select the mismatched instances, and within, we observed 2,436 portions where Llama model failed to identify a heading compared to GPT. As a target of the preference optimised model is to identify headings where Llama failed in comparison to GPT as well as to discourage it to extract paragraph sentences as headings, we randomly augment 50% of these instances with the first non-heading sentence of the respective proportion and use them as rejected outputs.

We use the remaining 75 policies (10%) to quantify the results of the DPO training. β is a prominent hyper-parameter used in DPO to control the deviation of the trained model with respect to reference policy, and we select the most suitable beta based on several metrics we use to quantify heading extraction. The median and inter-quartile (IQR) length of the average of privacy policy content (excluding the identified headings) lengths as a metric characterises the *fitment* of headings across policy, which we can directly compare against Llama3.1 and GPT5. For example, lower quartile getting closer to zero indicates more identifications of sub-headings with no subsequent content, meaning they are likely list items rather than sub-headings. We additionally consider the average number of policy headings each model identifies to further quantify results and we elaborate on the results in Sub Section 5.4.1.

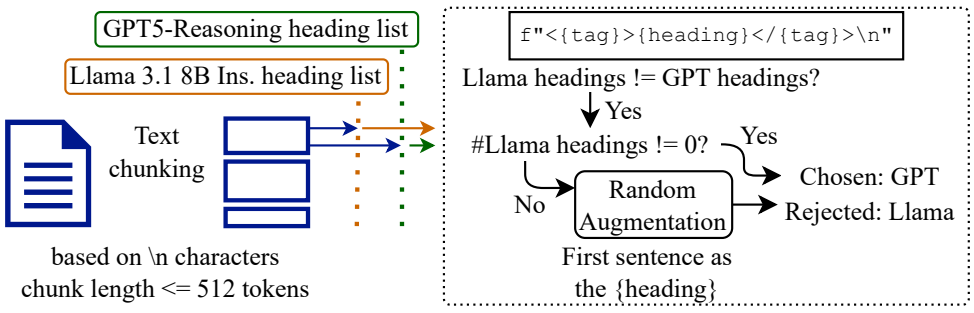

Figure 5.5: Creation of DPO dataset

5.3.4 Decoder based data item and purpose extraction

Literature highlights that embedding based models are not effective in semantic search due to the vector space noise [140]. Comparatively it is straightforward for decoder models as a natural language understanding (NLU) task [147] to extract which segments represent data items and data purposes for a given text chunk with 512 tokens at most. Given the nature of this task, we directly use Llama3.1 8B instruct and represents outputs in a JSON format;

$$[\{`data' : `data\ excerpt', `purpose' : `purpose\ excerpt'\}, ...]$$

and we emphasise that we allow data or purpose keys to contain empty strings if necessary. If we observe any empty purpose for a given data, we later on try to map them globally with a suitable purpose discussed elsewhere, which is a key novelty in this work.

To link **globally-defined** purposes, we create a link between isolated purposes (i.e. no locally defined data item) listed under a *data intention = Why? or How Used?* and isolated data items (i.e. no locally defined data purpose) listed under a *data intention = What?*. An example can be observed in orange colour dashed lines in Figure 5.4. If we observe isolated data items or purposes not belonging to above mentioned data intentions, we still link them but as **weak global (floating) relationships.** Note that the above mentioned two linkages are pre-conditioned on *data practice* category and we do not match first party collection data items with third party sharing data purposes (or vice versa).

5.3.5 Encoder based classifiers

Following the information extraction phase, we employ four fine-tuned PrivBERT models to classify the extracted text into their respective categories. The training of these modules was based on synthetic labels generated via GPT5 prompting for information extracted from the 750 privacy policies. We demonstrate the fine-tuning performance in Table 5.1.

The classifiers for Data Item and Data Purpose operate on a straightforward paradigm: they map each input text excerpt to a single class label using batched inferencing. While the Data Practice and Data Intent classifiers could follow this same approach, we hypothesise that encoder-based models benefit significantly from additional structural context. To test this, we evaluate an input schema that includes the section heading list within the prompt itself. Specifically, we format the input as "`[CLS] target_heading [SEP] context_headings [SEP]`". Due to PrivBERT's 512-token input limit, we include only a subset of adjacent headings (a 'neighbourhood') rather than the full list. We evaluate the impact of this context-aware approach in Sub Section 5.4.2.

A complete list of classification labels:

- **Data Practices:** First Party Collection/Use (c_0), Third Party Sharing (c_1), User Choice/Control (c_2), User Access, Edit and Deletion (c_3), Introductory/Generic (c_4), Policy Change (c_5), Data Security (c_6), International and Specific Audience (c_7), Practice Not Covered (c_8), Data Retention (c_9), Privacy Contact Information (c_{10}), and Do Not Track (c_{11}).
- **Data Intentions:** What (i_0), Why (i_1), How-Collected (i_2), How-Used (i_3), When (i_4), Other (i_5).
- **Data Items (D):** Name (d_0), Email (d_1), User account (d_2), Address (d_3), Phone (d_4), Race/Ethnicity (d_5), Political/Religious (d_6), Gender (d_7), Financial (d_8), Location (d_9), Search and Browsing history (d_{10}), SMS/ Messages/ Call log (d_{11}), Photos/Videos (d_{12}), Audio/Music (d_{13}), Health/Fitness (d_{14}), Contacts (d_{15}), Calendar (d_{16}), App performance/ App Activity (d_{17}), Device identifier (d_{18}), Files/Documents (d_{19}), Other Personal (d_{20}), Generic information (d_{21}) and Negatives (d_{22}).
- **Data Purposes (P):** App Analytics (p_0), Developer communication (p_1), Fraud prevention/security and compliance (p_2), Advertising or marketing (p_3), Personalisation (p_4), Account management (p_5), App functionality (p_6), and Other (p_7).

5.3.6 Metrics for Data Purpose Compliance

In this subsection, *compliance* indicates whether a given data item and its purpose is disclosed in the data safety (DS) declarations and also correctly reflected and described in the corresponding privacy policy (PP) text. More formally, for a given $\{PP, DS\}$ pair, we could observe the purpose compliance by comparing the presence of each individual purpose for and when a given data item is agreed as collected or shared among the pair. To establish a generalised landscape of data purpose compliance (for all pairs of PP and DS instances), we define the probability of observing a given purpose ($p[i]$) being *well-disclosed* for a given data item ($d[j]$) as $P_{WD}(i,j)$ if and only if **locally-defined**.

$$P_{WD}(i,j) = P(p[i] \in (PP \cap DS) \mid d[j] \in (PP \cap DS))$$

$P_{OS}(i,j)$ represents the probability when a purpose $p[i]$ that is **locally** defined in the privacy policy is not observed in data safety, and we consider this as the probability of *purpose over-statement*. Similarly $P_{US}(i,j)$ which is the probability of *purpose under-statement* defines that a purpose is mentioned in the data safety but not in the privacy policy. (Note that the condition of purpose non-existence or $P_{NE}(i,j)$ is not explicitly mentioned due to less significance but can be obtained by $1 - P_{WD}(i,j) - P_{OS}(i,j) - P_{US}(i,j)$).

$$P_{OS}(i,j) = P(p[i] \in (PP \setminus DS) \mid d[j] \in (PP \cap DS))$$

$$P_{US}(i,j) = P(p[i] \in (DS \setminus PP) \mid d[j] \in (PP \cap DS))$$

Note: Probabilities are computed per data item $d[j]$ of a given data practice classification type. In the main context of the chapter, we are mostly interested about data practices of PPs related

with first party collection (c_0) and third party sharing (c_1) which can be directly compared and contrasted against DS declarations regarding compliance.

As we have discussed before, developers are inclined to not locally define data item and data purpose relationships, and rather they tend to disclose via **global definitions**. I.e. data items may be in bulk declared in one section related to collection practices with relevant purposes declared elsewhere and vice versa for sharing practices. As PrivSTRUCT is capable of extracting such relationships, it is important to evaluate the compliance when taken these into consideration as well. The **delta** contribution we receive when these **globally-defined** data-item and purpose pairs are considered to the original P_X is denoted as Δ. For example, based on locally and globally defined data items and purposes, probability of well-disclosed purposes being observed will be $P_{WD} + \Delta P_{WD}$.

For us to globally match such purposes, it is expected that the developers disclose data items inside a heading related to "WHAT" data intention (e.g. What information do we collect from you?) and disclose the data purposes inside a heading related to "WHY" data intention (e.g. Why do we collect information from you?). However, developers could still disclose **floating** data items and data purposes in other sections where the true intent is not clear. In this scenarios, when the data practice category is strictly clear (i.e. collection or sharing), we **weakly global match** such data items and data purposes and the additional contribution we receive from this mapping to the compliance is indicated by Δ'.

5.3.7 Metrics for Data Purpose Dilution

Data Purpose Dilution refers to a phenomenon where the purpose for collecting or sharing a specific data item, as declared in the DS declaration, does not directly align with the purpose stated in the PP. Instead, the purpose in one document (e.g., DS) is "diluted" or spread across a diverse set of other purposes in the other document (e.g., PP) for the same data item, or vice versa. This misalignment creates ambiguity and reduces clarity for end-users about why their data is being processed.

To calculate the data purpose dilution matrix (DM) we deploy the logic given by the following equation where we identify the disagreements between the purposes of DS and PP for a given data item j and dilute them across other purposes with disagreements. We use a 8x7 matrix as the 'other' purpose is non existent for DS declaration for which it always follow $p[7] \in DS = 0$,

$$DM(x, y) = (p[x] \in PP \setminus DS) \wedge (p[y] \in DS \setminus PP)$$

We normalise each dilution matrix for a given data type for the summation of all cells to

be one for a given DS and PP pair. For observing the general landscape of data dilution, we add all dilution matrices (per data item) and normalise again. (Note: DM is different from the confusion matrix in traditional multi-class classification problems, as there are no set ground truth and prediction values for the problem we are discussing.) This metric helps us to identify which data purposes for a given data item are consistently not-disclosed by the developers while disclosing one or more unrelated data purposes.

5.4 Benchmarking PrivSTRUCT

To validate the efficacy of the PrivSTRUCT framework, we conduct a benchmarking analysis. First, we evaluate the structural extraction capabilities of our DPO-trained models against state-of-the-art LLM baselines and next, we assess the classification performance of our encoder-based modules under varying context settings. Finally we compare PrivSTRUCT with PoliGrapher, a state-of-the-art knowledge graph based NLP tool for extracting privacy policy information.

5.4.1 DPO for heading extraction

We show the benchmarking results of the Direct Preference Optimisation (DPO) training in Figure 5.6. Sub Figure (a) represents the fitment of the identified headings across the policy based on IQR and median of the accompanied content of each heading for five β values. We would like to refer the reader to sub figure (b) at the same time, where we depict the average number of headings we identify for each β. We observed that $\beta = 0.1$ the default parameter for the `DPO Trainer`—proved too aggressive for our task, resulting in a collapse where the model failed to identify meaningful headings. Comparing our DPO-Llama3.1 model against the baseline references (standard Llama3.1 Instruct and GPT-5), we observe that GPT-5 generally yields tighter IQRs and identifies a higher volume of headings. However, as we increase β to encourage the model not to deviate excessively from the reference policy, stability improves. Specifically, $\beta = 0.4$ offers the optimal balance, achieving an alignment median closest to both Llama3.1 and GPT-5, while maintaining a lower quartile distinct from zero (a zero value typically indicates list items with no subsequent content misclassified as headings). Sub figure (c) further emphasises the instability at lower values ($\beta = 0.1, 0.2$), highlighting the necessity of hyper-parameter tuning over relying on defaults for structural extraction tasks.

Additionally, we conducted a manual evaluation for the headings we identified, indicating the author preference for the outputs when comparing respective DPO-trained Llama3.1, standard Llama3.1 and GPT5 models. Out of the 75 policies we compare among the test dataset,

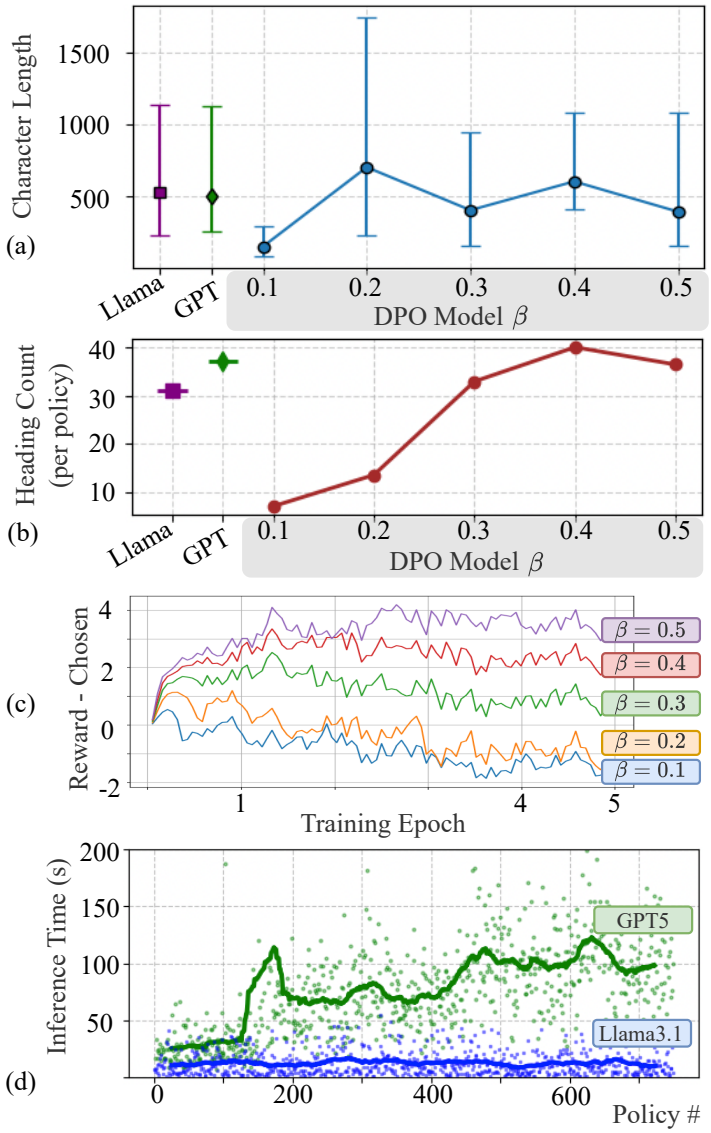

Figure 5.6: DPO training results for heading extraction.

we preferred GPT outputs 50.67% followed up by DPO-Llama at 42.67%. This result is significant as the latter performed well given it a much smaller decoder model compared to the state-of-the-art GPT5 with reasoning capabilities. Moreover, simply compared with Llama3.1 we preferred the DPO-Llama3.1, 65.33% of the times. This benchmarking depicts that, *we can reliably use a DPO trained locally hosted decoder models for heading extraction with no API costs and much faster inference times (c.f. sub figure d).* For the at-scale analysis we use the DPO trained Llama model for heading-extraction task.

5.4.2 Encoder based classifiers

Text classification using encoder models is well-established and proven efficient for privacy policy analysis [44, 141]. Table 5.1 presents the macro-average scores for the four classifiers

Classifier	Feed	#Labels	#Train	#Test	Pr	Re	F1
1. Data Practice	single	12	21k	2795	0.86	0.86	0.86
	multiple	12	21k	2795	0.96	0.96	0.96
2. Data Intention	single	6	21k	2795	0.85	0.85	0.85
	multiple	6	21k	2795	0.87	0.87	0.87
3. Data Items	N/A	23	74k	8271	0.87	0.83	0.85
4. Data Purpose	N/A	8	51k	6320	0.93	0.92	0.93

Table 5.1: Macro average classification results of the four classifiers we use in PrivSTRUCT. The column feed indicates whether we feed surrounding headings in the neighbourhood of the actual heading we are classifying as an additional input to the model.

utilised in PrivSTRUCT. The column #Train indicates how many samples we used to fine tune each model and we followed LLM bootstrapped training, where the training labels are generated via GPT5 models. This stage allows us to use efficient encoder models ($\sim$500M parameters in total) for the label generation task without relying on locally hosted much larger Llama3.1 models or API based generative models, still providing comparable performance.

We specifically highlight the ablation studies for the **Data Practice** and **Data Intention** classifiers (Rows 1 and 2 in Table 5.1). Here, feed denotes whether the model received only the target heading (single) or the target heading accompanied by its surrounding neighbourhood of headings (multiple). The multiple setting enables the model's self-attention mechanism to leverage the structural context of the document. While increasing the MAX_LENGTH parameter to accommodate this context increases computational cost, it yields significant gains: we observed a substantial **10 percentage point improvement** in F1 score for Data Practice classification (e.g., distinguishing First Party Collection from Third Party Sharing). However, the improvement for Developer Intent classification was marginal, showing only a 2 percentage point increase.

5.4.3 PrivSTRUCT versus PoliGraph

We use open source PoliGrapher NLP tool by PoliGraph [49] to parse the test dataset of 75 privacy policies and to compare data item and purpose identification against PrivSTRUCT. The dataset we selected is uniformly distributed across the 0-50KB policy text size range covering smaller and larger privacy policies both. Figure 5.7 displays the comparison with respect to unique identifications of items and purposes from the input.

First we highlight that PoliGrapher tool did not parse 6 (8%) privacy policies (all>25KB) and despite PrivSTRUCT processing all 75 input policies, we only discuss the instances where both frameworks successfully provided outputs. First, in terms of data items, PrivSTRUCT identifies average 79.6 unique mentions per policy (a single sentence excerpt from a privacy policy may contain mentions of multiple data items) while PoliGrapher only identifies 38.1 (52.1%

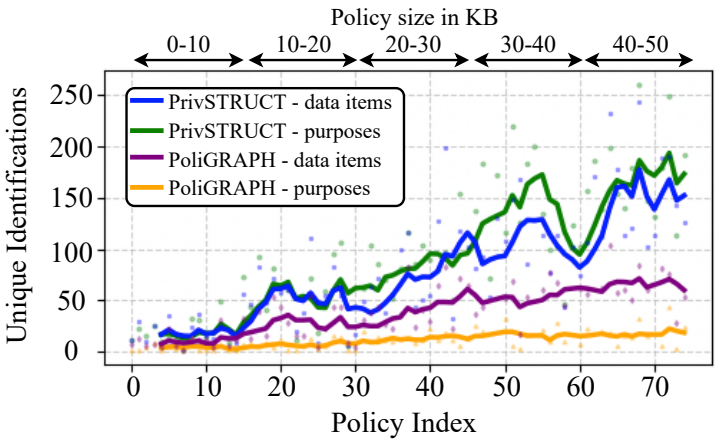

Figure 5.7: Comparison with PoliGraph

less) in average. Simply considering unique data purpose excerpts, PrivSTRUCT identifies 92.8 in average compared to 10.1 (89.1% less) with PoliGrapher. In average, PrivSTRUCT identifies approx. 2.9 more purpose categories providing more information to be used with downstream tasks.

5.5 Results

For discussing the results of the large scale analysis of 3,756 privacy policy documents, we follow the metrics defined in Sub-sections 5.3.6 and 5.3.7. We critically evaluate purpose compliance landscape based on each of the privacy policy (PP) and the data safety (DS) declaration of the most popular app representing the privacy policy. Given that both forms of disclosures (PP and DS) are developer-self declared, our expectation as well as with any end-user is to observe an explanatory text in the PP regarding the data purposes declared in DS. This is also enforced by regulatory frameworks and app market operator mandates under *required consistency*. We observed that, despite the availability of 22 data item categories, majority of them were very sparse; for example, we did not or rarely observed any mention of *d[5]-race/ethnicity* and *d[6]-political/religious* categories. For the rest of this section, we discuss the purpose compliance with respect to eight high-frequent data item categories.

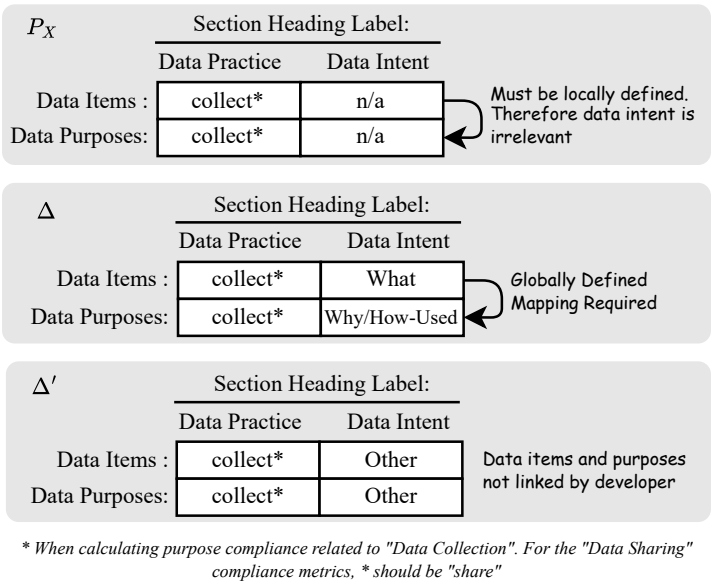

Figure 5.8: Locally-defined, globally-defined or un-defined/floating purposes

Figure 5.8 summarises how developers define data purposes compared to the data items in their disclosures. P_X represents locally-defined purpose statements where it is irrelevant to specifically consider developer data intentions. They could be related to either first party collection or third party sharing based on the data practice classification label. Δ represents globally-defined purposes where an intention based mapping is required to link 'what' data items are disclosed and 'why' these data items are used. Δ' represents floating data purposes that are observed elsewhere in data intention classifications, yet still belonging to the same data practice category.

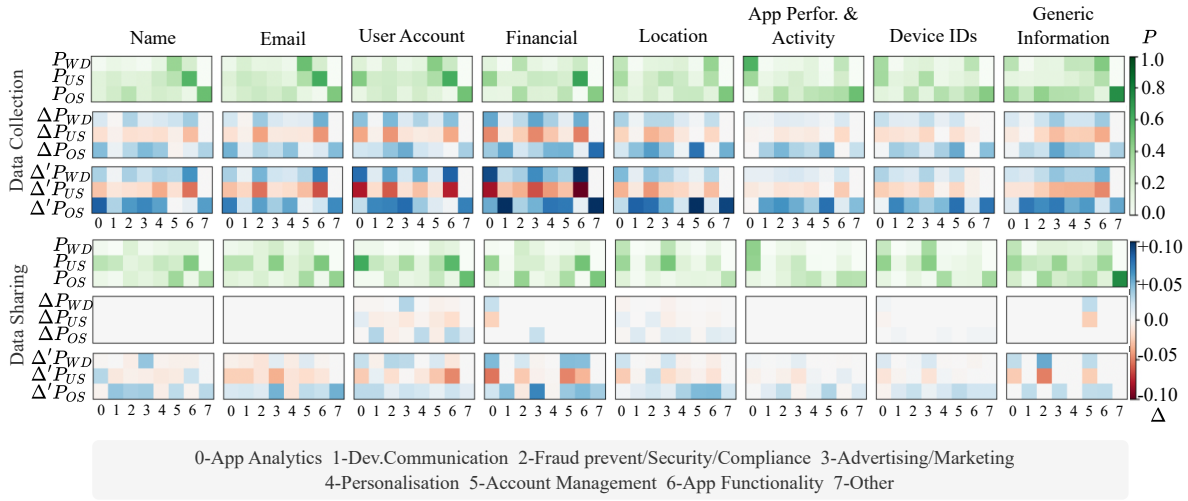

Figure 5.9: Results for Purpose Compliance

		Name	Email	User Acct.	Finc.	Loc.	App Act.	Dev. ID	Generic
collect	P_{WD}	0.111	0.154	0.159	0.122	0.143	0.144	0.135	0.187
	P_{US}	0.214	0.230	0.278	0.253	0.196	0.143	0.165	0.137
	P_{OS}	0.222	0.204	0.190	0.199	0.214	0.264	0.244	0.347
share	P_{WD}	0.073	0.099	0.055	0.063	0.097	0.080	0.099	0.131
	P_{US}	0.219	0.250	0.301	0.199	0.198	0.123	0.172	0.253
	P_{OS}	0.172	0.188	0.172	0.202	0.125	0.188	0.154	0.288

		Avg (P_X)	Δ	Δ'
collect	P_{WD}	**0.144**	+0.017	+0.026 (+18.0%)
	P_{US}	**0.202**	-0.016	-0.026 (-12.9%)
	P_{OS}	**0.235**	+0.028	+0.048 (+20.4%)
share	P_{WD}	**0.087**	+0.001	+0.006 (6.9%)
	P_{US}	**0.214**	-0.001	-0.009 (4.2%)
	P_{OS}	**0.186**	+0.003	+0.018 (9.7%)

Table 5.2: Average probability of occurrence for purpose well-disclosure (WD), under-statement (US) and over-statement (OS) for eight frequently used data item categories. The top table shows the item-specific probabilities, while the bottom table summarises the averages and deltas.

5.5.1 Well-(and not so well)-disclosed Purposes

As shown in Figure 5.9, we can observe that having a well-disclosed purpose for a given data item is not uniform (i.e., due to hot spots if we go through column-wise in the row P_{WD} for each data item). *Name*, *email*, and *user account* have a higher probability of being well-disclosed when collected for account management. Similarly, *financial* information is likely well-disclosed for fraud prevention/security and compliance and *location, app performance, and app activity, device identifiers* for app analytics. All data items have a higher probability of being well-disclosed for app functionality. When *location* and *device identifier* are shared with third parties for advertising, it is likely to be well-disclosed.

Table 5.2 quantitatively discuss the results based on each data when averaged across all data purpose categories. The probability of observing **well-disclosed and locally-defined** purposes among the selected data items remains low at 14.4% for when collected and only 8.7% when shared. Comparatively the purpose over-statement within the policy text remains quite high, approximately with twice the probability. Note that the three probability values in each column do not add up to 1 as we do not show purpose non-existence (which simply dictates that a data item and data purpose are unlikely to be linked via developer disclosures) here.

Both in the figure and in table, we show the change to the original probabilities as soon as we include **globally-defined** purposes as Δ. In the figure, blue shades represent the increase

of the respective probabilities with red shades representing decreases. In the results, we also depict **unlinked/floating data purpose** compliance which are easy to be missed by end-users as the section headings do not properly disclose the developer intention with such data practices. We denote these contributions via Δ'. As we identify more and more data-item linkages that are globally mapped, well-disclosures and over-statements tend to go higher while reducing the under-statements. More specifically, average $P_{WD}|collect$ improves by 18.0% and average $P_{WD}|share$ does not increase as much; only by 6.9%. This indicates that developers are less likely to emphasise on data they share with third parties, and rather tend to explain their collection usages. This is more apparent with nearly 40% of a privacy policy is used for describing collection details and only 25% for sharing details in average. Qualitatively we observed that policies tend to disclose redirections (e.g. 'Refer to the third-party website's privacy policy for ...') instead of clearly indicating purposes. This raises a critical question for end-user privacy whether is it realistic for someone to read all such policies when many apps use third party services for analytics and advertising. The highest probability of understatement is observed with user account data being shared with third-parties.

5.5.2 Purpose Dilution

Figure 5.10 represents the asymmetric purpose dilution matrices for the selected data items. As the data safety does not contain *other* category, it is omitted from the x-axis of all matrices. However, ambiguous PP purposes could still be diluted across DS declarations. A significant purpose dilution can be characterised by horizontal or vertical lines that stand out in these matrices and hot spots are created when they intersect.

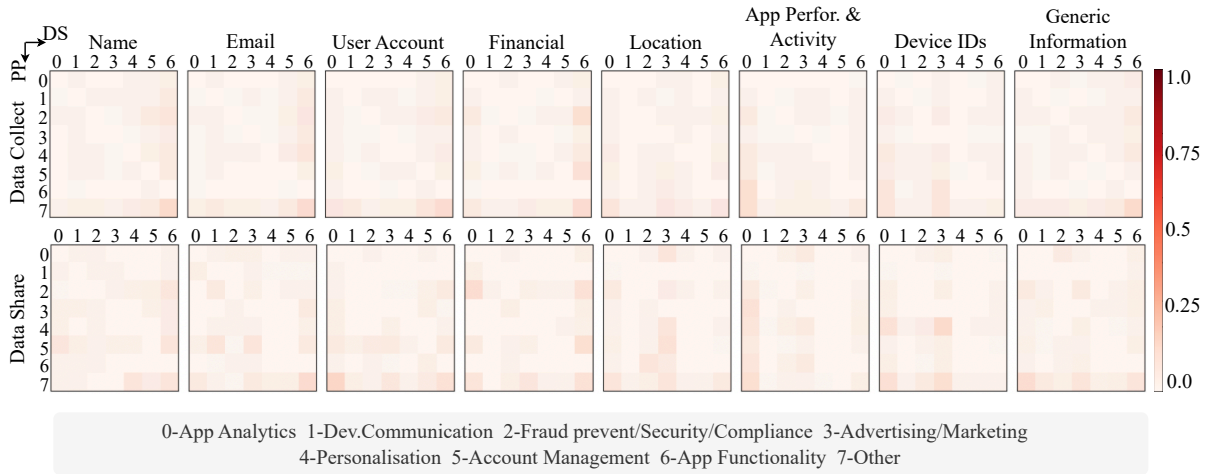

Figure 5.10: Results for Purpose Dilution

A primary observation is that *Financial, location, app performance/activity, device identifier* data collected or shared for *app analytics* is often diluted across other privacy policy

purposes. A qualitative explanation is majority of privacy policies are not optimised based on app based services, rather they are optimised with online-services and traditional privacy policy architectures in mind. We argue that privacy policies representing mobile apps should require modifications or sections specifically mentioning mobile app based data practices. We also observe that *device identifier* data collected or shared for *advertising* is diluted as *personalisation*, *fraud prevention, security or compliance*, or *other* purposes in the privacy policy. User *email* when shared for *advertising and developer communication* is considerably diluted as *account management* which is a serious privacy concern for end-users. *Financial details* when shared for *analytics* are more easier to be misinterpreted as shared for *fraud prevention and security and compliance purposes.*

Our tailored compliance metrics allow this finer grained evaluation and simply observing the hot spots in the figure, we could deduce that purpose dilution is more prominent among the data items that are shared with third parties. This again reinforces the discussion in Sub Section 5.5.1 that more clearer explanations are required about third party data sharing within the privacy policies themselves.

5.6 Related Work

Prior efforts to demystify privacy policies relied on traditional NLP for summarisation and decision support [92, 86, 97, 14, 81, 90]. These methods were eventually superseded by encoder-based transformers and related architectures [43, 44]. More recently, the landscape has shifted toward generative approaches, which are increasingly demonstrating remarkable efficacy in zero- and few-shot policy understanding [47, 46, 141, 96].

Extracting structured information from policy text via semantic analysis is crucial for tasks such as consistency checking. PolicyLint [41] introduced linguistic analysis to extract (data type, entity) tuples from individual sentences, enabling the detection of internal contradictions. Building on this, PurPliance [148] extended the tuple structure to identify processing purpose clauses, allowing for comparisons against actual app behaviour. Similarly, PolicyChecker [149] leveraged semantic analysis roles to audit the consistency between policy disclosures and app code. However, these approaches rely on isolated sentence-level analysis; they cannot resolve dependencies across multiple sections of a policy, often resulting in extracted tuples that are disconnected or ambiguous. Alternative frameworks like PolicyComp [150] attempt to bypass these extraction challenges by benchmarking policies against those of similar 'counterpart' apps, but this merely identifies statistical outliers and does not resolve the fundamental inability of current models to handle complex, cross-section dependencies."

Recent work PoliGraph [49] move beyond linear text analysis by modelling privacy policies

as knowledge graphs. In this architecture, data types and entities function as nodes, connected by two primary edge relations: SUBSUME, which maps generic terms to specific definitions, and COLLECT, which links entities to the data they acquire. This structural approach has proven highly effective for extracting data flows; empirical evaluations showing that PoliGrapher identifies 40% more collection statements than prior state-of-the-art methods, achieving 97% precision. However, while effective for collection, the purpose of that collection is treated merely as a secondary attribute attached to the COLLECT edge. This dependency presents a significant limitation: purposes in complex policy text are frequently ambiguous or syntactically detached from explicit collection verbs. As a result, purposes are not always well-defined solely by the existence of a link between an entity and a data type. This structural rigidity often leads to missed or misattributed purposes, underscoring the need for more robust identification methods.

5.7 Conclusion

In this chapter, we presented **PrivSTRUCT**, a structure-aware framework designed to uncover the complex dependencies between data items and purposes in mobile app privacy policies. By moving beyond traditional flat-text analysis, we demonstrated that leveraging the hierarchical structure; specifically through the extraction and classification of section headings is essential for accurate interpretation. Our methodological contributions include a hybrid architecture that pairs efficient encoder-based classifiers with a decoder-based information extractors. We showed that this approach not only enables cost-effective, locally hosted inference but also significantly outperforms existing state-of-the-art tools like PoliGraph, in average identifying at least twice (or greater) as many data item or purpose excerpts in our benchmark comparisons.

By leveraging robust extraction of data practices, items, and purposes from privacy policies, we characterise the disclosure landscape of popular apps in the Google Play Store. Using our novel framework, PrivSTRUCT, we uncover the relationships between globally defined purposes and specific data items. Our analysis reveals that, compared to relying solely on locally defined purposes, the probability of developers overstating purposes is 20.4% higher for first-party collection and 9.7% higher for third-party sharing. We attribute this to developers relying on 'global' purpose definitions that are difficult for end-users to parse at a glance. Alarmingly, this encourages blanket consenting, as the probability of adequate purpose disclosure remains critically low.

Most critically, our analysis of "Purpose Dilution" highlights that third-party data sharing remains the most opaque aspect of the ecosystem. We observed that highly sensitive data flows, such as the sharing of financial information or device identifiers for advertising, are frequently

diluted into unrelated categories like “Fraud Prevention” or generic “Other” clauses within the policy text. This misalignment suggests that despite the introduction of simplified Data Safety labels, the underlying legal documents remain a barrier to true transparency.

Chapter 6

Conclusion

The central thesis of this work has been that mobile app ecosystems exhibit a persistent "transparency paradox", in which users are expected to make informed decisions based on legally binding privacy policy disclosures that are unrealistic to read and interpret at scale, while regulators and platform operators simultaneously lack scalable means to verify the "privacy compliance" of developer-declared practices. This thesis argues that both challenges of user-facing transparency and system-level privacy compliance can be jointly addressed through automated privacy compliance checks grounded in natural language processing (NLP) techniques.

Throughout this research, we demonstrate that existing NLP solutions are insufficient to capture the semantic nuance, logical structure, and dependencies between data practices, and lack the explainability and verifiability required for analysing legal text in mobile app privacy disclosures, particularly when compliance verification is the end goal. To address these limitations, we developed three complementary frameworks: *Entailment-Driven LLMs*, *PrivPRISM*, and *PrivSTRUCT*. Collectively, these contributions advance the capability to parse, interpret, and evaluate privacy disclosures at scale, enabling both improved transparency assessment and automated compliance reasoning.

This chapter synthesises the contributions of this thesis, reflecting on the implications of our findings for the broader privacy compliance landscape. We subsequently discuss the limitations of the current work and outline a roadmap for future research, with particular attention to how regulators and platform operators can leverage these approaches in transition from reactive enforcement towards proactive ecosystem-wide oversight.

6.1 Implications

The contributions of this thesis extend beyond technical novelty, offering robust frameworks for evaluating privacy policy text and revealing systemic compliance gaps in the mobile ecosystem. We summarise our methodological and empirical contributions below.

6.1.1 Methodological: Systematic Language Modelling

Beyond standard "black-box" embedding methods and unconstrained generative approaches, in Chapter 3, we addressed the critical challenges of verifiability and explainability in privacy policy analysis. We introduced a systematic pipeline that integrates an *explained classifier* to generate candidate data practice labels with reasoning, a *blank-filling module* to re-evaluate those rationales, and an *entailment verifier* to validate the logical consistency between the text and the predicted class. We demonstrated that this architecture outperforms vanilla LLMs (including GPT-4 and LLaMA-2) by an average of 11.2% in F1 score. Crucially, this approach produces natural language explanations that align closely with legal expert reasoning, effectively improving the transparency of generative model outputs in regulatory domains.

Chapter 4 overcomes the noise inherent in vanilla embedding- based semantic retrieval and the heavy reliance on legal expert annotations, which are scarce for fine-grained, mobile-specific data item and purpose characterisation. In PrivPRISM, we leverage the natural language understanding (NLU) capabilities of generative models to extract information through targeted, explanatory linguistic prompts. These extractions are subsequently validated through a lightweight encoder-based verifier, which filters out hallucinations or errors. By utilising an LLM- bootstrapped training strategy where synthetic decoder outputs serve as pseudo-labels for training the verifier, this self-supervised pipeline reduces keyword mapping errors by 22.3 percentage points compared to state-of-the-art unverified LLM outputs.

Chapter 5 addresses the "flat text" assumption, which causes automated methods to entangle data item and purpose interpretations. We utilised preference-optimised LLMs to develop a structure-aware analysis framework, PrivSTRUCT. By explicitly modelling the hierarchical headings of a policy (distinguishing "what" is collected or shared from "why" they are collected or shared), we effectively solved the problem of "floating" data practices. This approach proved superior to flat-text analysis, identifying more than double the number of unique data items and purpose excerpts compared to existing tools.

6.1.2 Empirical: The Compliance Landscape

While consistency between the privacy policy and the Data Safety declaration is a marketplace and regulatory requirement, our analysis reveals it is rarely achieved. We observed discrepancies in over 53% of mobile games, a figure that rises to 61% for generic non-game applications. The nature of these discrepancies follows a clear pattern: developers consistently "over-disclose" in privacy policies (treating them as catch-all liability shields) while "under-declaring" in Data Safety labels. This misalignment is exacerbated by the widespread reuse of generic policy templates, with nearly 65% of analysed apps reusing policies, thereby diluting

app-specific accountability.

The transparency gap widens further when contrasting declarations against ground-truth evidence extracted from application code. Our static code analysis revealed that neither the privacy policy nor the Data Safety labels fully captures the extent of sensitive data access. For instance, while high access rates were confirmed for precise or approximate location (98.3% of apps) and financial data (84.3% of apps) in the code, disclosure rates lagged significantly behind; privacy policies disclosed location access in only 64.1% of cases, and Data Safety labels in less than 40%. This confirms that relying solely on developer self-attestation, whether in legal text or marketplace labels, fails to provide a complete picture of privacy risks.

Finally, we identified a systemic ambiguity of developer intentions. Our analysis of 3,756 policies showed that data processing purposes are frequently diluted or ill-defined. We found that developers who rely on "globally defined purposes" (broad mandates declared once for the entire document) rather than specific, locally defined justifications are significantly more likely to overstate their data practices. Specifically, the probability of purpose overstatement increases by 20.4% for data collection when global definitions are used, effectively rendering the "purpose limitation" principle unenforceable as users cannot meaningfully link specific data items to their intended use.

6.1.3 Regulatory Implications

The findings of this thesis motivate a shift from reactive to proactive auditing. E-safety regulators and platform operators can leverage our frameworks to automatically flag potential non-compliant actors at the point of app submission. For example, automated checks can identify straightforward enforceable actions based on specific discrepancy patterns: (1) *Placeholder or dummy policies*, characterised by low PP compliance scores despite high DS declarations; and (2) *Boilerplate policies*, characterised by high PP coverage (generic text) but low alignment with specific DS declarations or code-level evidence. (3) More in depth evaluations based on our tailored metrics can identify developers with concerning levels of non-compliance indications, to be audited for end-user privacy violations. Operationalising these checks would prevent non-compliant disclosures from reaching the public. Furthermore, app market operators can leverage PrivPRISM and PrivSTRUCT frameworks to assess the alignment of mobile app specific data item and purpose disclosures in online privacy policies to further strengthen transparency to end-users.

6.2 Limitations and Future Works

A primary limitation of our verification methodology is the reliance on static code analysis. While scalable and effective for identifying potential data access, static analysis cannot confirm runtime data transmission. Techniques like code obfuscation or dynamic class loading can hide data practices from static parsers. To address this, future work should isolate the flagged "bad actors" and conduct dynamic analysis. Emerging agentic frameworks can simulate user interactions to trigger and intercept actual data flows, as proposed in recent literature (e.g., AppAgent [151], DroidBot [152]). Using autonomous agents to navigate apps and verify if data is transmitted would provide definitive proof of non-compliance. It is also noteworthy to highlight that inputs taken via user inputs may still not be captured in dynamic analysis and simulated behaviour may not be representative of actual user journey in apps (e.g. games).

Furthermore, our work focuses exclusively on a single platform (Google Play Store). Despite the privacy policy analysis we propose in this thesis being largely platform-agnostic, a comprehensive app privacy analysis requires extending these frameworks to other ecosystems, particularly Apple's App Store. This presents a unique challenge, as Apple employs a distinct disclosure taxonomy that centres on concepts such as *Data Linked to You* and *Data Used to Track You*, rather than the data collection and sharing categories used by Google. Future research should therefore focus on developing robust mappings between legal text and these platform-specific labels to enable inconsistency detection within the iOS ecosystem. Additionally, investigating policy reusage patterns in cross-platform applications where developers use a single policy to satisfy divergent store requirements would provide critical insights into how such practices affect end-user comprehension and regulatory compliance.

Privacy policies utilised in this study are filtered by English language and as further depicted in Appendix A.4, a number of non-English policies were excluded despite being crawled based on Australian geo-location. Future work can parse these non-English policies based on multilingual LLMs and can explore reasoning behind this language-region incompatibility. Furthermore, our analysis targets at most popular apps and future work can explore how compliance metrics vary with respect to the popularity when the apps do not attract millions of users. While this thesis rigorously benchmarks different language models in our systematic designs, future work can also elaborate on how different versions (e.g. based on parameter count, released date, intra-model variants such as reasoning and non-reasoning, etc.) of the same model family contribute to the outputs. Future directions can also leverage the dataset released with this thesis to conduct longitudinal behaviour of privacy policies, especially for the developers obtaining low compliance scores.

Finally, ongoing advances in LLM optimisation, such as Group Relative Policy Optimisation (GRPO) introduced by Shao et al. [153], offer promising opportunities to further strengthen privacy policy information extraction and reasoning. In particular, GRPO objective could be explored to improve the reasoning of explained classifier within the *entailment-driven LLM* framework, as well as the precision of data item and data purpose identification components of *PrivPRISM*. Future research could also investigate how Mixture-of-Experts (MoE) models (e.g., Llama 4 Scout [154]) can be integrated to support cross-jurisdictional or cross–market-operator compliance checks at scale. Finally, recent advances in agentic frameworks present an immediate pathway to address practical privacy policy accessibility challenges, such as automatically resolving redirections and dynamically locating applicable policy documents, thereby extending this work toward fully autonomous compliance analysis.

Bibliography

[1] GSMA, "The mobile economy 2020," Global System for Mobile Communications Association, Report, 2020. [Online]. Available: https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2020/03/GSMA_MobileEconomy2020_Global.pdf

[2] GSMA, "The mobile economy 2025," Global System for Mobile Communications Association, Report, march 2025. [Online]. Available: https://www.gsma.com/solutions-and-impact/connectivity-for-good/mobile-economy/wp-content/uploads/2025/04/030325-The-Mobile-Economy-2025.pdf

[3] Commonwealth Bank, "Annual report 2020," Commonwealth Bank of Australia, Annual Report, 2020. [Online]. Available: https://www.commbank.com.au/content/dam/commbank/about-us/shareholders/pdfs/results/fy20/cba-2020-annual-report.pdf

[4] Commonwealth Bank, "Annual report 2025," Commonwealth Bank of Australia, Annual Report, 2025. [Online]. Available: https://www.commbank.com.au/content/dam/commbank-assets/investors/docs/results/fy25/2025-annual-report.pdf

[5] Roy Morgan Market Research, "7.4 million australians are now using uber compared to around 4.2 million using taxis – a gap of over 3 million," june 2025, finding No. 9911. [Online]. Available: https://www.roymorgan.com/findings/9911-uber-streaks-ahead-of-taxis-june-2025

[6] T. Demetriou, "The impact of mobile ordering platforms on dining establishments," august 2022, epos Now. [Online]. Available: https://www.eposnow.com/au/resources/the-impact-of-mobile-ordering-platforms-on-dining-establishments/

[7] BuildFire, "Mobile app download statistics & usage statistics (2025)," december 2024, last updated December 31, 2024. [Online]. Available: https://buildfire.com/app-statistics/

[8] T. Ermakova, B. Fabian, and E. Babina, "Readability of privacy policies of healthcare websites," in *Wirtschaftsinformatik Proceedings 2015*, 2015, paper 73. Available at: https://aisel.aisnet.org/wi2015/73.

[9] A. M. McDonald and L. F. Cranor, "The cost of reading privacy policies," *I/S: A Journal of Law and Policy for the Information Society*, vol. 4, p. 543, 2008.

[10] E. Okoyomon, N. Samarin, P. Wijesekera, A. Elazari Bar On, N. Vallina-Rodriguez, I. Reyes, Á. Feal, S. Egelman *et al.*, "On the ridiculousness of notice and consent: Contradictions in app privacy policies," in *Workshop on Technology and Consumer Protection (ConPro 2019), in conjunction with the 39th IEEE Symposium on Security and Privacy*, 2019.

[11] Consumer Policy Research Centre, 2018. [Online]. Available: https://cprc.org.au/2018/05/13/research-australian-consumers-soft-targets-big-data-economy/

[12] B. Auxier, L. Rainie, M. Anderson, A. Perrin, M. Kumar, and E. Turner, "Americans' attitudes and experiences with privacy policies and laws," Pew Research Center, Report, november 2019. [Online]. Available: https://www.pewresearch.org/internet/2019/11/15/americans-attitudes-and-experiences-with-privacy-policies-and-laws/

[13] C. Black, L. Setterfield, and R. Warren, "Online data privacy from attitudes to action: An evidence review," Carnegie UK Trust, Report, august 2018. [Online]. Available: https://carnegieuk.org/wp-content/uploads/2024/10/Online-Data-Privacy-from-Attitudes-to-Action-CUKT-2.pdf

[14] H. Harkous, K. Fawaz, R. Lebret, F. Schaub, and K. Aberer, "Polisis: Automated analysis and presentation of privacy policies using deep learning," *Proceedings of the 27th USENIX Security Symposium*, p. 531 – 548, 02 2018.

[15] European Parliament and Council of the European Union, "Protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," *Official Journal of the European Union*, vol. L 119, pp. 1–88, 2016, available at: http://data.europa.eu/eli/reg/2016/679/oj.

[16] M. Degeling, C. Utz, C. Lentzsch, H. Hosseini, F. Schaub, and T. Holz, "We value your privacy... now take some cookies: Measuring the GDPR's impact on web privacy," *arXiv preprint arXiv:1808.05096*, 2018.

[17] T. Linden, R. Khandelwal, H. Harkous, and K. Fawaz, "The privacy policy landscape after the GDPR," *arXiv preprint arXiv:1809.08396*, 2018.

[18] R. Amos, G. Acar, E. Lucherini, M. Kshirsagar, A. Narayanan, and J. Mayer, "Privacy policies over time: Curation and analysis of a million-document dataset," in *Proceedings of the Web Conference*, 2021.

[19] G. Das, C. Cheung, C. Nebeker, M. Bietz, C. Bloss *et al.*, "Privacy policies for apps targeted toward youth: descriptive analysis of readability," *JMIR mHealth and uHealth*, vol. 6, no. 1, p. e7626, 2018.

[20] M. Rudolph, D. Feth, and S. Polst, "Why users ignore privacy policies–a survey and intention model for explaining user privacy behavior," in *HCI International 2018, Las Vegas, USA*. Springer, 2018, pp. 587–598.

[21] J. Valentino-DeVries, N. Singer, M. H. Keller, and A. Krolik, "Your apps know where you were last night, and they're not keeping it secret," 2018. [Online]. Available: https://www.nytimes.com/interactive/2018/12/10/business/location-data-privacy-apps.html

[22] S. Perez, "Developer spams google play with ripoffs of well-known apps...again," 2014. [Online]. Available: https://techcrunch.com/2014/01/02/developer-spams-google-play-with-ripoffs-of-well-known-apps-again/

[23] G. Cleary, "Mobile privacy: What do your apps know about you?" 2018. [Online]. Available: https://www.symantec.com/blogs/threat-intelligence/mobile-privacy-apps

[24] S. Perez, "Consumer advocacy groups call on ftc to investigate kids' apps on google play," 2018. [Online]. Available: https://techcrunch.com/2018/12/19/consumer-advocacy-groups-call-on-ftc-to-investigate-kids-apps-on-google-play/

[25] T. D. et al., "Exploring the far side of mobile health:information security and privacy of mobile health apps on ios and android," 2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4319144/

[26] G. A. Fowler, "It's the middle of the night. do you know who your iphone is talking to?" 2019. [Online]. Available: https://www.washingtonpost.com/technology/2019/05/28/its-middle-night-do-you-know-who-your-iphone-is-talking/

[27] H. Wang, Z. Liu, J. Liang, N. Vallina-Rodriguez, Y. Guo, L. Li, J. Tapiador, J. Cao, and G. Xu, "Beyond google play: A large-scale comparative study of chinese android app markets," 2018.

[28] S. Koch, M. Wessels, B. Altpeter, M. Olvermann, and M. Johns, "Keeping privacy labels honest," *Proceedings on Privacy Enhancing Technologies*, 2022.

[29] R. Baalous, A. Althobaiti, D. Alyoubi, R. Alzahrani, and M. Aljohani, "Detecting the inconsistency between android apps' data collection and google play's data safety using static analysis," *Cybernetics and Information Technologies*, vol. 25, no. 1, 2025.

[30] R. Khandelwal, A. Nayak, P. Chung, and K. Fawaz, "Unpacking privacy labels: A measurement and developer perspective on google's data safety section," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 2831–2848.

[31] R. Khandelwal, A. Nayak, P. Chung, and K. Fawaz, "Comparing privacy labels of applications in android and ios," in *Proceedings of the 22nd Workshop on Privacy in the Electronic Society*, 2023, pp. 61–73.

[32] Y. Lin, J. Juneja, E. Birrell, and L. F. Cranor, "Data safety vs. app privacy: Comparing the usability of android and ios privacy labels," *Proceedings on Privacy Enhancing Technologies*, vol. 2024, no. 2, 2024.

[33] I. Arkalakis, M. Diamantaris, S. Moustakas, S. Ioannidis, J. Polakis, and P. Ilia, "Abandon all hope ye who enter here: A dynamic, longitudinal investigation of android's data safety section," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 5645–5662.

[34] M. Khedkar, A. K. Mondal, and E. Bodden, "Do android app developers accurately report collection of privacy-related data?" in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering Workshops*, 2024, pp. 176–186.

[35] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications magazine*, vol. 48, no. 9, pp. 140–150, 2010.

[36] B. Falchuk and S. Loeb, "Privacy enhancements for mobile and social uses of consumer electronics," *IEEE Communications Magazine*, vol. 48, no. 6, pp. 102–108, 2010.

[37] Katten Muchin Rosenman LLP, "Privacy policies now a must for mobile apps," january 2013, available at: https://katten.com/Privacy-Policies-Now-a-Must-for-Mobile-Apps.

[38] A. Sunyaev, T. Dehling, P. L. Taylor, and K. D. Mandl, "Availability and quality of mobile health app privacy policies," *Journal of the American Medical Informatics Association*, vol. 22, no. e1, pp. e28–e33, 2015.

[39] McNamara Law, "Why you shouldn't rely on ai to write legal documents," january 2025, available at: https://mcna.com.au/why-you-shouldnt-rely-on-ai-to-write-legal-documents/.

[40] T. Willingham, "Can you trust ai to write your website's privacy policy?" september 2025, the Admin Bar. Available at: https://theadminbar.com/can-you-trust-ai-to-write-your-websites-privacy-policy/.

[41] B. Andow, S. Y. Mahmud, W. Wang, J. Whitaker, W. Enck, B. Reaves, K. Singh, and T. Xie, "PolicyLint: Investigating internal privacy policy contradictions on google play," in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 585–602. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/andow

[42] J. Bhatia, M. C. Evans *et al.*, "Automated extraction of regulated information types using hyponymy relations," in *IEEE 24th International Requirements Engineering Conference Workshops (REW)*, 2016.

[43] A. Adhikari, S. Das, and R. Dewri, "Evolution of composition, readability, and structure of privacy policies over two decades," *Proceedings on Privacy Enhancing Technologies*, 2023.

[44] M. Srinath, S. Wilson, and C. L. Giles, "Privacy at scale: Introducing the privaseer corpus of web privacy policies," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 6829–6839.

[45] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you? explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD*, 2016, pp. 1135–1144.

[46] D. Rodriguez, I. Yang, J. M. D. Alamo, and N. Sadeh, "Large language models: A new approach for privacy policy analysis at scale," 2024. [Online]. Available: https://arxiv.org/abs/2405.20900

[47] C. Tang, Z. Liu, C. Ma, Z. Wu, Y. Li, W. Liu, D. Zhu, Q. Li, X. Li, T. Liu *et al.*, "PolicyGPT: Automated analysis of privacy policies with large language models," *arXiv preprint arXiv:2309.10238*, 2023.

[48] C. Chen, D. Zhou, Y. Ye, T. J.-j. Li, and Y. Yao, "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*, 2025, pp. 277–297.

[49] H. Cui, R. Trimananda, A. Markopoulou, and S. Jordan, "{PoliGraph}: Automated privacy policy analysis using knowledge graphs," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1037–1054.

[50] L. L. Porta. (2018) What are third party app stores and are they safe? [Online]. Available: https://www.jamf.com/blog/what-are-third-party-app-stores-and-are-they-safe/

[51] Android, “Android security & privacy 2018 year in review,” Google, Report, march 2019. [Online]. Available: https://source.android.com/static/docs/security/overview/reports/Google_Android_Security_2018_Report_Final.pdf

[52] (2022) Mobile ecosystems - market study final report. [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/1096277/Mobile_ecosystems_final_report_-_full_draft_-_FINAL__.pdf

[53] Statista, “Number of available apps in the google play store from 3rd quarter 2009 to 2025,” 2025, available at: https://www.statista.com/statistics/289418/number-of-available-apps-in-the-google-play-store-quarter/.

[54] S. G. Stats, “Operating system market share worldwide,” https://gs.statcounter.com/os-market-share, 2025.

[55] (2020) Threat intelligence report 2020. [Online]. Available: https://www.nokia.com/networks/portfolio/cyber-security/threat-intelligence-report-2020/

[56] Apple. (2019) Building a trusted ecosystem for millions of apps. [Online]. Available: https://www.apple.com/privacy/docs/Building_a_Trusted_Ecosystem_for_Millions_of_Apps.pdf

[57] P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder, “A” nutrition label” for privacy,” in *Proceedings of the 5th Symposium on Usable Privacy and Security*, 2009, pp. 1–12.

[58] P. G. Kelley, L. Cesca, J. Bresee, and L. F. Cranor, “Standardizing privacy notices: an online study of the nutrition label approach,” in *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, 2010, pp. 1573–1582.

[59] T. Li, K. Reiman, Y. Agarwal, L. F. Cranor, and J. I. Hong, “Understanding challenges for developers to create accurate privacy nutrition labels,” in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–24.

[60] J. Gardner, Y. Feng, K. Reiman, Z. Lin, A. Jain, and N. Sadeh, “Helping mobile application developers create accurate privacy labels,” in *2022 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, 2022, pp. 212–230.

[61] S. Zimmeck, P. Story, D. Smullen, A. Ravichander, Z. Wang, J. Reidenberg, N. C. Russell, and N. Sadeh, “Maps: Scaling privacy compliance analysis to a million apps,” *Proceedings on Privacy Enhancing Technologies*, 2019.

[62] Y. Chen, H. Xu, Y. Zhou, and S. Zhu, "Is this app safe for children? a comparison study of maturity ratings on android and ios applications," in *Proceedings of the 22nd international conference on World Wide Web*, 05 2013, pp. 201–212.

[63] J. Rajasegaran, N. Karunanayake, A. Gunathillake, S. Seneviratne, and G. Jourjon, "A multi-modal neural embeddings approach for detecting mobile counterfeit apps," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 3165–3171. [Online]. Available: https://doi.org/10.1145/3308558.3313427

[64] N. Viennot, E. Garcia, and J. Nieh, "A measurement study of google play," *ACM SIGMETRICS Performance Evaluation Review*, vol. 42, 06 2014.

[65] R. Kumar, A. Virkud, R. S. Raman, A. Prakash, and R. Ensafi, "A large-scale investigation into geodifferences in mobile apps," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 1203–1220. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/kumar

[66] B. Fabian, T. Ermakova, and T. Lentz, "Large-scale readability analysis of privacy policies," in *Proceedings of the international conference on web intelligence*, 2017, pp. 18–25.

[67] C. Jensen and C. Potts, "Privacy policies as decision-making tools: an evaluation of online privacy notices," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2004, pp. 471–478.

[68] G. Meiselwitz, "Readability assessment of policies and procedures of social networking sites," in *International conference on online communities and social computing*. Springer, 2013, pp. 67–75.

[69] J. R. Reidenberg, T. Breaux, L. F. Cranor, B. French, A. Grannis, J. T. Graves, F. Liu, A. McDonald, T. B. Norton, R. Ramanath *et al.*, "Disagreeable privacy policies: Mismatches between meaning and users' understanding," *Berkeley Tech. LJ*, vol. 30, p. 39, 2015.

[70] J. R. Reidenberg, J. Bhatia, T. Breaux, and T. B. Norton, "Automated comparisons of ambiguity in privacy policies and the impact of regulation. 2016," *URL http://papers. ssrn. com/sol3/papers. cfm*, 2016.

[71] T. Libert, “An automated approach to auditing disclosure of third-party data collection in website privacy policies,” in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 207–216.

[72] (2022) Providing a safe and trusted experience for everyone. [Online]. Available: https://play.google.com/about/developer-content-policy/

[73] (2022) App privacy details on the app store. [Online]. Available: https://developer.apple.com/app-store/app-privacy-details/

[74] H. Habib, S. Pearman, J. Wang, Y. Zou, A. Acquisti, L. F. Cranor, N. Sadeh, and F. Schaub, “” it’s a scavenger hunt”: Usability of websites’ opt-out and data deletion choices,” in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.

[75] V. Bannihatti Kumar, R. Iyengar, N. Nisal, Y. Feng, H. Habib, P. Story, S. Cherivirala, M. Hagan, L. Cranor, S. Wilson, F. Schaub, and N. Sadeh, “Finding a choice in a haystack: Automatic extraction of opt-out statements from privacy policy text,” in *Proceedings of The Web Conference 2020*, ser. WWW ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1943–1954. [Online]. Available: https://doi.org/10.1145/3366423.3380262

[76] M. K. Uddin, Q. He, J. Han, and C. Chua, “Mining cross-domain apps for software evolution: A feature-based approach,” in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 743–755. [Online]. Available: https://doi.org/10.1109/ASE51524.2021.9678514

[77] L. F. Cranor, “P3p: Making privacy policies more useful,” *IEEE Security & Privacy*, vol. 1, no. 6, pp. 50–55, 2004.

[78] J. Reardon, Á. Feal, P. Wijesekera, A. E. B. On, N. Vallina-Rodriguez, and S. Egelman, “50 ways to leak your data: An exploration of apps’ circumvention of the android permissions system,” in *USENIX Security Symposium*, 2019.

[79] W. Ammar, S. Wilson, N. Sadeh, and N. A. Smith, “Automatic categorization of privacy policies: A pilot study,” *School of Computer Science, Language Technology Institute, Technical Report CMU-LTI-12-019*, 2012.

[80] E. Costante, Y. Sun, M. Petković, and J. Den Hartog, “A machine learning solution to assess privacy policy completeness: (short paper),” in *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society*, 2012, pp. 91–96.

[81] S. Zimmeck and S. M. Bellovin, "Privee: An architecture for automatically analyzing web privacy policies," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 1–16.

[82] S. Wilson, F. Schaub, A. A. Dara, F. Liu, S. Cherivirala, P. G. Leon, M. S. Andersen, S. Zimmeck, K. M. Sathyendra, N. C. Russell *et al.*, "The creation and analysis of a website privacy policy corpus," in *Proceedings of the 54th ACL*, 2016.

[83] F. Liu, S. Wilson, P. Story, S. Zimmeck, and N. Sadeh, "Towards automatic classification of privacy policy text," *School of Computer Science Carnegie Mellon University*, 2018.

[84] N. Mousavi Nejad, P. Jabat, R. Nedelchev, S. Scerri, and D. Graux, "Establishing a strong baseline for privacy policy classification," in *IFIP International Conference on ICT Systems Security and Privacy Protection*. Springer, 2020, pp. 370–383.

[85] M. Mustapha, K. Krasnashchok, A. Al Bassit, and S. Skhiri, "Privacy policy classification with xlnet (short paper)," in *International Workshop on Data Privacy Management*. Springer, 2020, pp. 250–257.

[86] K. M. Sathyendra, F. Schaub, S. Wilson, and N. M. Sadeh, "Automatic extraction of opt-out choices from privacy policies." in *AAAI Fall Symposia*, 2016.

[87] N. Nisal, S. K. Cherivirala, K. M. Sathyendra, M. Hagan, F. Schaub, S. Wilson *et al.*, "Increasing the salience of data use opt-outs online," in *Symposium on Usable Privacy and Security*, vol. 2017, 2017.

[88] K. M. Sathyendra, A. Ravichander, P. G. Story, A. W. Black, and N. Sadeh, "Helping users understand privacy notices with automated query answering functionality: An exploratory study," *Technical report, CMU-ISR-17-114R*, 2017.

[89] R. N. Zaeem, R. L. German, and K. S. Barber, "Privacycheck: Automatic summarization of privacy policies using data mining," *ACM Transactions on Internet Technology (TOIT)*, vol. 18, no. 4, pp. 1–18, 2018.

[90] R. Nokhbeh Zaeem, S. Anya, A. Issa, J. Nimergood, I. Rogers, V. Shah, A. Srivastava, and K. S. Barber, "Privacycheck v2: A tool that recaps privacy policies for you," in *Proceedings of the 29th ACM international conference on information & knowledge management*, 2020, pp. 3441–3444.

[91] R. Nokhbeh Zaeem, A. Ahbab, J. Bestor, H. H. Djadi, S. Kharel, V. Lai, N. Wang, and K. S. Barber, "Privacycheck v3: empowering users with higher-level understanding of privacy policies," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1593–1596.

[92] A. A. M. Gopinath, V. B. Kumar, S. Wilson, and N. Sadeh, "Automatic section title generation to improve the readability of privacy policies," *USENIX SOUPS*, 2020.

[93] J. Woodring, K. Perez, and A. Ali-Gombe, "Enhancing privacy policy comprehension through privacify: A user-centric approach using advanced language models," *Computers & Security*, vol. 145, p. 103997, 2024.

[94] V. Freiberger, A. Fleig, and E. Buchmann, "Explainable ai in usable privacy and security: Challenges and opportunities," *arXiv preprint arXiv:2504.12931*, 2025.

[95] P. Story, S. Zimmeck, A. Ravichander, D. Smullen, Z. Wang, J. Reidenberg, N. C. Russell, and N. Sadeh, "Natural language processing for mobile app privacy compliance," in *AAAI spring symposium on privacy-enhancing artificial intelligence and language technologies*, vol. 2, no. 4, 2019, p. 4.

[96] J. Chanenson, M. Pickering, and N. Apthorpe, "Automating governing knowledge commons and contextual integrity (gkc-ci) privacy policy annotations with large language models," *Proceedings on Privacy Enhancing Technologies*, 2025.

[97] K. M. Sathyendra, S. Wilson, F. Schaub, S. Zimmeck, and N. Sadeh, "Identifying the provision of choices in privacy policy text," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 2774–2779.

[98] V. Freiberger, A. Fleig, and E. Buchmann, "" you don't need a university degree to comprehend data protection this way": Llm-powered interactive privacy policy assessment," in *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2025, pp. 1–12.

[99] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," *arXiv preprint cs/0205070*, 2002. [Online]. Available: https://arxiv.org/pdf/cs/0205070

[100] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of Machine Learning Research*, vol. 3, no. 6, p. 1137 – 1155, 2003, cited by: 4316. [Online]. Available: https://www.jmlr.org/papers/volume3/bengio03a/bengio03a.pdf

[101] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162/

[102] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the association for computational linguistics*, vol. 5, pp. 135–146, 2017. [Online]. Available: https://direct.mit.edu/tacl/article-pdf/doi/10.1162/tacl_a_00051/1567442/tacl_a_00051.pdf

[103] A. Salle, A. Villavicencio, and M. Idiart, "Matrix factorization using window sampling and negative sampling for improved word representations," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, K. Erk and N. A. Smith, Eds. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 419–424. [Online]. Available: https://aclanthology.org/P16-2068/

[104] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, no. 3. Makuhari, 2010, pp. 1045–1048.

[105] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[106] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014. [Online]. Available: https://arxiv.org/abs/1409.3215

[107] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, p. 5999 – 6009, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[108] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423/

[109] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[110] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[111] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, and Others, "Improving language understanding by generative pre-training," OpenAI, Technical Report, 2018, available at: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[112] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI*, 2019.

[113] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[114] J. Achiam, S. Adler *et al.*, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[115] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[116] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023. [Online]. Available: https://arxiv.org/abs/2310.06825

[117] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[118] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022.

[119] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct preference optimization: Your language model is secretly a reward model," *Advances in neural information processing systems*, vol. 36, pp. 53 728–53 741, 2023.

[120] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[121] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 39.

[122] J. Kolen and S. Kremer, *A Field Guide to Dynamical Recurrent Neural Networks*. Wiley-IEEE Press, 04 2001.

[123] A. Graves, "Generating sequences with recurrent neural networks," *arXiv.org*, 2014.

[124] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *NeurIPS*, vol. 35, pp. 24 824–24 837, 2022.

[125] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[126] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.

[127] R. Binns, U. Lyngs, M. Van Kleek, J. Zhao, T. Libert, and N. Shadbolt, "Third party tracking in the mobile ecosystem," in *Proceedings of the 10th ACM Conference on Web Science*, 2018, pp. 23–31.

[128] H. Habib, Y. Zou, A. Jannu, N. Sridhar, C. Swoopes, A. Acquisti, L. F. Cranor, N. Sadeh, and F. Schaub, "An empirical analysis of data deletion and {Opt-Out} choices on 150 websites," in *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, 2019, pp. 387–406.

[129] P. Robles, "GDPR: What future for first, second and third-party data," 2018. [Online]. Available: https://econsultancy.com/gdpr-what-future-for-first-second-and-third-party-data//

[130] B. Krumay and J. Klar, "Readability of privacy policies," in *Data and Applications Security and Privacy XXXIV: 34th Annual IFIP WG 11.3 Conference, DBSec 2020, Germany*. Springer, 2020, pp. 388–399.

[131] F. Schaub, R. Balebako, A. L. Durity, and L. F. Cranor, "A design space for effective privacy notices," in *Eleventh symposium on usable privacy and security (SOUPS 2015)*, 2015, pp. 1–17.

[132] H. Touvron, L. Martin, K. Stone *et al.*, "LLaMA 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[133] E. Waisberg, J. Ong, M. Masalkhi, S. A. Kamran, N. Zaman, P. Sarker, A. G. Lee, and A. Tavakkoli, "GPT-4: A new era of artificial intelligence in medicine," *Irish Journal of Medical Science*, vol. 192, no. 6, 2023.

[134] S. Wu, O. Irsoy, S. Lu *et al.*, "Bloomberggpt: A large language model for finance," *arXiv preprint arXiv:2303.17564*, 2023.

[135] M. Madden, "Privacy, security, and digital inequality," September 2017.

[136] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin *et al.*, "Code llama: Open foundation models for code," *arXiv preprint arXiv:2308.12950*, 2023.

[137] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *JMLR*, 2020.

[138] Y. Kementchedjhieva and I. Chalkidis, "An exploration of encoder-decoder approaches to multi-label classification for legal and biomedical text," *arXiv preprint arXiv:2305.05627*, 2023.

[139] J. Caltrider and A. Stopper, "See no evil: Loopholes in google's data safety labels keep companies in the clear and consumers in the dark," https://www.mozillafoundation.org/en/campaigns/googles-data-safety-labels/, Feb. 2023, mozilla Foundation Report.

[140] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, vol. 2, no. 1, 2023.

[141] B. Silva, D. Denipitiyage, S. Seneviratne, A. Mahanti, and A. Seneviratne, "Entailment-driven privacy policy classification with llms," in *2024 Conference on Building a Secure & Empowered Cyberspace (BuildSEC)*. IEEE, 2024, pp. 8–15.

[142] S. Zhang, Y. Feng, Y. Yao, L. F. Cranor, and N. Sadeh, "How usable are ios app privacy labels?" *Proceedings on Privacy Enhancing Technologies*, 2022.

[143] P. A. Bonatti, S. Kirrane, I. M. Petrova, L. Sauro, and E. Schlehahn, "The special usage policy language, v1.0," https://ai.wu.ac.at/policies/policylanguage/, 2018, version 1.0.

[144] M. Ahmad, V. Costamagna, B. Crispo, F. Bergadano, and Y. Zhauniarovich, "Stadart: Addressing the problem of dynamic code updates in the security analysis of android applications," *Journal of Systems and Software*, vol. 159, p. 110386, 2020.

[145] Z. Qu, S. Alam, Y. Chen, X. Zhou, W. Hong, and R. Riley, "Dydroid: Measuring dynamic code loading and its security implications in android applications," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2017, pp. 415–426.

[146] S. Zimmeck, Z. Wang, L. Zou, R. Iyengar, B. Liu, F. Schaub, S. Wilson, N. M. Sadeh, S. M. Bellovin, and J. R. Reidenberg, "Automated analysis of privacy requirements for mobile apps." in *NDSS*, vol. 2, 2017, pp. 1–4.

[147] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv e-prints*, pp. arXiv–2407, 2024.

[148] D. Bui, Y. Yao, K. G. Shin, J.-M. Choi, and J. Shin, "Consistency analysis of data-usage purposes in mobile apps," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2824–2843.

[149] B. Andow, S. Y. Mahmud, J. Whitaker, W. Enck, B. Reaves, K. Singh, and S. Egelman, "Actions speak louder than words:{Entity-Sensitive} privacy policy and data flow analysis with {PoliCheck}," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 985–1002.

[150] L. Zhou, C. Wei, T. Zhu, G. Chen, X. Zhang, S. Du, H. Cao, and H. Zhu, "{POLICYCOMP}: counterpart comparison of privacy policies uncovers overbroad personal data collection practices," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 1073–1090.

[151] C. Zhang, Z. Yang, J. Liu, Y. Li, Y. Han, X. Chen, Z. Huang, B. Fu, and G. Yu, "Appagent: Multimodal agents as smartphone users," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025, pp. 1–20.

[152] H. Wen, H. Wang, J. Liu, and Y. Li, "Droidbot-gpt: Gpt-powered ui automation for android," *CoRR*, 2023.

[153] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li *et al.*, "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.

[154] Meta AI, "Llama 4: Multimodal intelligence," https://ai.meta.com/blog/llama-4-multimodal-intelligence/, Apr. 2025, accessed: 2025-12-25.

[155] A. Desnos and G. Gueguen, "Androguard," https://github.com/androguard/androguard, 2011.

Appendix A

A.1 Labels of Google Data Safety Declarations

In this appendix section, we provide a comprehensive overview of the labels used in Google's Data Safety declaration and how our fine-granular interpretations correlate with that. First, we observed all possible combinations of data practice declarations developers can provide for an app during the submission process. As illustrated in Figure A.1, data safety declaration starts with the data practice category, which is either data collection for first-party usage or data sharing with third parties. Also, the developers can disclose data security (request to delete) and whether the data is encrypted or not in transit from the mobile app to the collection party. When it comes to privacy policy, it is more fine-grained in classification. We follow common data practice types introduced by OPP-115 legal expert annotations, which are still widely used. These categories are shown in c_0 to c_{11}.

Next, Google's Data Safety declaration provides detailed labels about what data categories (e.g., location data) and attributes (e.g., approximate or precise location) are shared or collected. These finer levels of detail are not provided for data encryption and deletion request declarations. When deciding the data items suitable to be identified from a privacy policy, we closely followed this structure and we decided on 22 data items $d[0]$ to $d[21]$. The final data item of 'Generic Information' is introduced as we empirically observed many privacy policies using terms such as 'we collect your information', which are highly ambiguous, nonetheless, still require some attention. Also, it classifies terms that are out of domain here, such as 'cookies'. During the keyword mapping stage of our framework, we assigned an additional data item class $d[22] = negative$ that provides a decoder model to observe a text segment and decide as not suitable. This helps in error correction, where a sentence with multiple data items is separated using traditional NLP techniques. E.g. Word 'etc.' after a list of data items needs to be classified as a negative if it was captured.

Lastly, the Data Safety Declaration provides details for each data category (not attribute level) about why this data category is collected or shared and can be one of seven data purposes

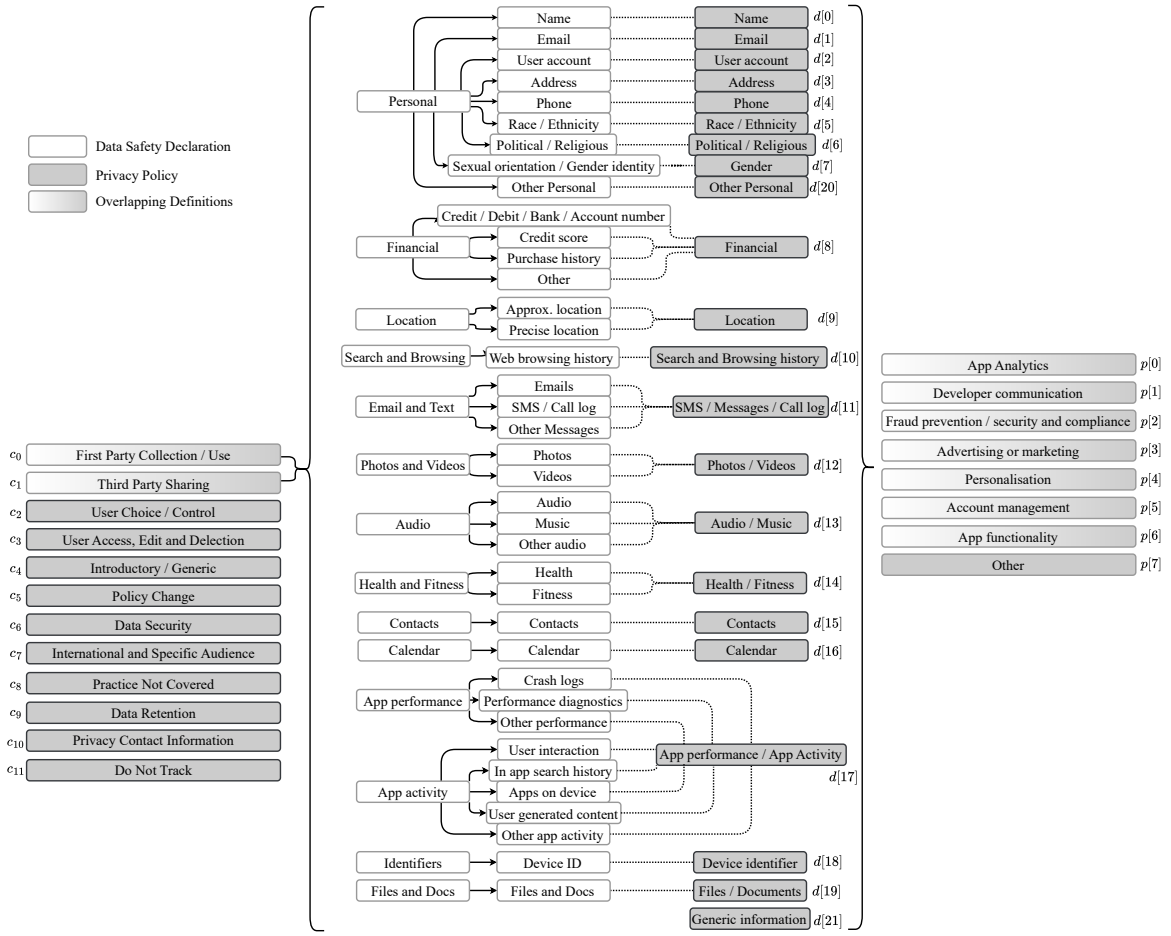

Figure A.1: Combinations of Google Data Safety declaration labels, compared with data practice, data item and data purpose categories we utilise in PrivPRISM and PrivSTRUCT frameworks.

as depicted in $p[0]$ to $p[7]$. App analytics and app functionality purposes are heavily biased towards mobile applications but other categories are generalised, therefore we followed the same structure in privacy policy based purpose identification. Additionally, we introduce the 'other' category to classify any purpose that does not belong to one of these. In summary, this hierarchical structure allows us to have a fine-grained analysis of the privacy policies and to detect potential compliance violations in their respective mobile applications.

A.2 Sanitisation of the Data Safety Declarations

HTML files of the Data Safety (DS) pages were sanitised using a Document Object Model (DOM)–based approach with python library `BeautifulSoup` by referencing necessary division classes, which represents the page as a hierarchical tree of elements. This enabled the identification of primary data practice sections, descriptive statements, and nested categories detailing the types of data collected, their purposes, and any sharing practices. The extracted

components were annotated with lightweight structural tags to retain semantic relationships while ensuring machine-readable consistency across all apps. For example, a disclosure such as "Data Shared → Device or other IDs → Data shared and for what purpose → Analytics, Advertising or marketing" (this is the same example shown in Figure 4.1) is encoded as:

- `<d_prac> Data Shared`
- `<d_cata> Device or other IDs`
- `<d_detl> Data shared and for what purpose`
- `<d_valu> Analytics, Advertising or marketing`

A.3 APK Evidence Extraction

To complement the analysis of developer-declared data practices in Privacy Policies and Data Safety declarations, we conducted static code analysis on the corresponding Android application packages (APKs). Each APK file was parsed using Androguard [155], an open-source reverse-engineering toolkit for Android that supports programmatic inspection of manifests, compiled code, and control/data flow graphs. The goal of this step was to extract code-level evidence of data access that could be aligned against the declared information.

The analysis proceeded in two stages. First, the *AndroidManifest.xml* of each APK was parsed to obtain all explicit permission requests. These permissions describe what categories of user or device data the application claims to access (e.g., *ACCESS_FINE_LOCATION*, *READ_CONTACTS*). Second, the compiled Dalvik bytecode (contained in the `classes.dex` files) was examined through Androguard's `Analysis()` module. The resulting method analysis graph was traversed to identify API method invocations that are known to require or imply certain permissions. These API–permission associations were derived from a precompiled mapping resource (`api_permission_map`), which aligns Android framework API calls to the corresponding sensitive permissions defined by the Android SDK documentation.

Each permission (either declared in the manifest or inferred from API usage) was then mapped to high-level data items. For example;

- `android.permission.ACCESS_FINE_LOCATION → Location`
- `android.permission.ACCESS_COARSE_LOCATION → Location`
- `android.permission.READ_MEDIA_IMAGES → photos/videos`

A.4 Dataset

Developers of top apps tend to place more emphasis on complete data safety declarations due to being more resourceful and concerns over public perception [30]. Therefore, we narrow down our scope to consider top apps and, more specifically, games, as their privacy policies are less likely to be shared between corresponding web-based services. Based on a Google Play Store metadata crawl (approx 1.3M apps) we conducted during Q1 and Q2 of 2024, we select the top apps based on the number of "download counts" and further filter them based on games or non games based on "app category".

We observed that among top apps, games and non-game apps are almost equally distributed, with non-game apps starting to dominate after the fifth percentile. During the crawl, we also downloaded the privacy policies via developer-declared URLs using the `PyWebCopy` Python library that saves the HTML file with linked resources such as images and JavaScripts. Based on the downloaded policies, we observed a 29.42% failure rate due to errors in request refusals and incomplete downloads. Despite the mandatory requirement, 1.52% did not contain a privacy policy link, 1.74% contained PDF or TXT files as the privacy policy and 3.77% of policies were hosted on the docs.google.com domain.

In the subcategory of games, we selected the top most 7,770 successfully downloaded policies and respective their games as our selection for analysis. The lowest download count was 1.3M, with 174 games with 100M+ and 28.7% with 10M+ downloads. The average file size of the text-converted policies was 18.3KB (median 11KB), and we did not consider policies greater than 50KB ($< 3\%$ of total) in length. Furthermore, despite the attempt to crawl the privacy policies via links provided with app metadata based on an English-speaking geo-location, we still observed 506 instances with non-English policies (40 Japanese, 59 Korean, 52 Portuguese, etc.), indicating that end-users are bound to use translation services to read these privacy policies. We also observed that 4,526 of these games shared 1,147 policy links; i.e., multiple games from the same developer sharing the same privacy policy. As an example, there were 67 games governed by voodoo.io privacy policy. After these basic sanitation, we were left with 3,400 unique privacy policies to be used for the experimental setup. Policy HTML to text conversion was done using the `BeautifulSoup4` Python library.

In the subcategory of non-game (generic) apps, we selected the top most 6,540 successfully downloaded policies and their respective apps. Due to widespread policy reuse across apps developed by the same publisher, as discussed earlier, our analysis ultimately covered 3,756 unique non-game privacy policies. We observed the highest degree of policy reuse among Google, with a single privacy policy shared across 68 apps, followed by Samsung and several other developers, each reusing policies across 12 to 18 apps. To evaluate the generalisation

capability of the *PrivPRISM* framework, we further curated a balanced subset by selecting up to 50 apps from each available app category (e.g., 50 tools, 50 photography, etc.). Some categories were underrepresented in the initial selection (e.g., 43 weather, 21 medical, etc.), resulting in a total of 1,254 unique privacy policies used for this evaluation. In contrast, the full set of 3,756 unique policies was utilised when evaluating the *PrivSTRUCT* framework.