DISSERTATION

TOWARDS AUTOMATED SECURITY AND PRIVACY POLICIES SPECIFICATION AND ANALYSIS

Submitted by

Saja Salem Alqurashi

Department of Computer Science

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2024

Doctoral Committee:

Advisor: Indrakshi Ray

Indrajit Ray
Yashwant Malaiya
Steve Simske

ABSTRACT

TOWARDS AUTOMATED SECURITY AND PRIVACY POLICIES SPECIFICATION AND ANALYSIS

Security and privacy policies, vital for information systems, are typically expressed in natural language documents. *Security policy* is represented by Access Control Policies (ACPs) within security requirements, initially drafted in natural language and subsequently translated into enforceable policy. The unstructured and ambiguous nature of the natural language documents makes the manual translation process tedious, expensive, labor-intensive, and prone to errors. On the other hand, *Privacy policy*, with its length and complexity, presents unique challenges. The dense language and extensive content of the privacy policies can be overwhelming, hindering both novice users and experts from fully understanding the practices related to data collection and sharing. The disclosure of these data practices to users, as mandated by privacy regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), is of utmost importance.

To address these challenges, we have turned to Natural Language Processing (NLP) to automate extracting critical information from natural language documents and analyze those security and privacy policies. Thus, this dissertation aims to address two primary research questions:

Question 1: How can we automate the translation of Access Control Policies (ACPs) from natural language expressions to the formal model of Next Generation Access Control (NGAC) and subsequently analyze the generated model?

Question 2: How can we automate the extraction and analysis of data practices from privacy policies to ensure alignment with privacy regulations (GDPR and CCPA)?

Addressing these research questions necessitates the development of a comprehensive framework comprising two key components. The first component, *SR2ACM*, focuses on translating

natural language ACPs into the NGAC model. This component introduces a series of innovative contributions to the analysis of security policies. At the core of our contributions is an automated approach to constructing ACPs within the NGAC specification directly from natural language documents. Our approach integrates machine learning with software testing, a novel methodology to ensure the quality of the extracted access control model. The second component, *Privacy2Practice*, is designed to automate the extraction and analysis of the data practices from privacy policies written in natural language. We have developed an automated method to extract data practices mandated by privacy regulations and to analyze the disclosure of these data practices within the privacy policies.

The novelty of this research lies in creating a comprehensive framework that identifies the critical elements within security and privacy policies. Thus, this innovative framework enables automated extraction and analysis of both types of policies directly from natural language documents.

ACKNOWLEDGEMENTS

I am deeply grateful to my Ph.D. advisor Prof. Indrakshi Ray for her guidance, support, and expertise have been invaluable throughout my doctoral journey. I also would like to thank my committee members, Prof. Yashwant Malaiya, Prof. Indrajit Ray, and Prof. Steve Simske, for generously offering their time and guidance. Also, I am grateful to Dr. Hossein Shirazi for his monitoring and feedback during my research.

Also, I would like to express my sincere gratitude to the Saudi Arabian government and the Saudi Arabian Cultural Mission (SACM) for generously funding my Ph.D. studies. Their financial support has been instrumental in the successful completion of my research and academic journey.

Furthermore, I want to thank the funding agencies for supporting this work: this work was supported in part by funding from NSF under Award Numbers DMS 2123761, CNS 1822118, and from AFRL, ARL, Statnett, AMI, NewPush, and Cyber Risk Research and from NIST under Award Number 60NANB23D152.

DEDICATION

I dedicate this dissertation to my beloved parents and supportive siblings, whose unwavering love, encouragement, and sacrifices have been my guiding light throughout this academic journey. Additionally, I dedicate this dissertation to the cherished memory of my uncles. Their wisdom and love resonate in my life, providing motivation even in their absence.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Security and privacy policies are fundamental principles in information systems, meticulously crafted to ensure the security and privacy of the data. These policies are crucial in building trust, reducing risks, and safeguarding individuals' rights in the fast-changing digital landscape.

Security policies are articulated through Access Control Policies (ACPs), which serve as the foundational step in establishing data security requirements [57]. ACPs manage information flow, preventing unauthorized access and enabling systems to determine whether to grant or deny resource requests. Proper expression of ACPs is vital, as incorrect formulation can allow unauthorized users to exploit these weaknesses to access protected data and resources or prevent legitimate access. ACPs are embedded in security requirement documents that are written in a natural language [16]. ACPs should be derived from natural language documents and converted to machine executable instructions. Usually, ACPs are extracted manually. Moreover, security requirement documents contain unstructured, ambiguous, and implicit information. Thus, manual extraction is tedious, complex, expensive, labor-intensive, and error-prone. Automatic extraction helps system administration to extract ACPs accurately and at a low cost in terms of labor, time, and complexity [8, 9]. This motivates us toward automating ACPs extraction from security requirements documents in natural language, generating enforceable policies and security access control model, and then analyzing the generated model.

On the other hand, a *privacy policy* is a comprehensive legal document intricately designed to outline the processing and utilization of user data, thereby governing the terms and conditions related to data privacy. As the primary legally binding mechanism, privacy policies communicate an organization's data collection and sharing practices to users of their products and services. Analyzing the privacy policy before the compliance process benefits end-users, developers, law enforcement, and organizations. For end-users, it aids in understanding their rights and the organization's data handling practices. Developers aiming to comply with privacy laws benefit by

ensuring their systems align with legal requirements. Law enforcement agencies gain the capability to audit organizations while organizations themselves find assistance in streamlining compliance checks. However, privacy policies often present challenges due to their length and complexity, which can overwhelm novice users and experts [7, 35]. The dense language and extensive content can hinder individuals' comprehension of the data practices that are related to the data collection and sharing that are detailed in these policies. This complexity has spurred recent research efforts to develop methods for extracting critical information from these documents, enabling a more efficient and systematic analysis of data practices in privacy policies [5, 6, 19, 29]. This motivates us to automate the extraction of privacy data practices from natural language statements and analyze the privacy policy.

Therefore, this dissertation delves into the automated extraction and analysis of security and privacy policies articulated in natural language documents. Given such policies' increasing complexity and volume, traditional manual review methods often need to be more efficient and prone to oversight. This research examines and develops automated methods tailored to analyze security and privacy policies. The ultimate goal is to create sophisticated approaches capable of automatically assessing and evaluating these documents' security and privacy policies.

1.1 Challenges

1. **Complexity of Security Policies Expression**: Security must be accurately expressed to avoid introducing security vulnerabilities. Security requirements in natural language often contain unstructured, ambiguous, and implicit information, making it challenging to derive precise and actionable ACPs. Automated methods are required to extract the ACPs effectively.

2. **Length and Complexity of Privacy Policies:** Privacy policies are sophisticated legal documents characterized by their length and complexity. Understanding and analyzing these policies manually can be daunting for both users and developers. Automated analysis techniques are needed to streamline the process and ensure compliance with privacy law.

3. **Cost and Time Constraints:** Manual policy extraction and analysis processes are not only labor-intensive but also time-consuming and expensive. Automating these processes can significantly reduce costs, time, and complexity while improving accuracy and scalability.

4. **Insufficient Data for Automation:** Automation techniques rely heavily on data to train models, develop algorithms, and validate results. However, the lack of comprehensive data that accurately represents security and privacy policies hinders the effectiveness of these techniques.

1.2 Research Questions

- **RQ-1:** How can we automate the translation of Access Control Policies (ACPs) from natural language expressions to the formal model of Next Generation Access Control (NGAC) and subsequently analyze the generated model?

- **RQ-2:** How can we automate the extraction and analysis of data practices from privacy policies to ensure alignment with privacy regulations (GDPR and CCPA)?

1.3 The Proposed Framework

In this dissertation, I present a novel framework comprising two key components designed to address the aforementioned challenges.

The first component, Security Requirements to Access Control Model (*SR2ACM*), is tailored to automate the specification of ACPs from natural language and subsequently generate the NGAC model. This component streamlines the process of translating textual descriptions of ACPs into a structured model, enhancing efficiency and accuracy in policy formulation and analysis.

The second component, Privacy Policy to Practice (*Privacy2Practice*), is geared towards automating the extraction of data practices of privacy policies from natural language and analyzing whether the policy discloses the data practices that are required by the GDPR and the CCPA. This component facilitates the conversion of privacy-related textual content into the corresponding data practices.

These two components offer a comprehensive and automated approach to policy specification and analysis, contributing to the advancement of research and practice in the domains of security and privacy policies.

1.4 Contributions

1. An automated approach to construct ACPs in NGAC specification from natural language.
2. A novel approach approach integrating machine learning with software testing, offering an approach to ensure the quality of the extracted model.
3. A set of formal properties has been proposed to analyze the extracted models.
4. A realistic synthetic annotated dataset comprising access control policy sentences to evaluate the proposed approaches.
5. An automated approach that extracts data practices in privacy policies that are mandatory by privacy regulations such as the GDPR and the CCPA.

1.5 Dissertation Organization

The remainder of this dissertation is organized as follows. Chapter 2 provides an exploration of the key concepts central to this dissertation. Chapter 3 reviews the previous studies. Chapter 4 delves into a detailed linguistic analysis of policy language. Chapter 5 demonstrates the the proposed approach to extract ACPs from security requirements in natural language and automatically generate the NGAC model. Chapter 6 elaborates on the proposed approach to extract data practices and analyze the privacy policy disclosure. Chapter 7 summarizes the research problems, contributions and research outputs.

Chapter 2

Background

2.1 Access Control Models

In the 1970s, the foundational access control systems were developed. As we moved into the 1980s, the access control landscape witnessed the emergence of various flexible mechanisms, including Discretionary Access Control (DAC) [36], Mandatory Access Control (MAC) [51], Role-Based Access Control (RBAC) [24], and Attribute-Based Access Control (ABAC) [34]. These developments culminated in the more recent innovation of Next Generation Access Control (NGAC). This dissertation focuses specifically on attribute-based access control models, with the following subsections providing a comprehensive exploration of both ABAC and NGAC.

2.1.1 Attribute Access Control Model (ABAC)

The core concept of the ABAC model is that a user's access to perform actions on objects is determined by the attributes assigned to the subject, the attributes assigned to the object, and the attributes assigned to the environment. Policies in this model are specified in terms of these attributes [58]. According to the National Institute of Standards and Technology (NIST) documentation on the attribute access control model [34], ABAC is an access control model where authorization is based on the evaluation of attributes associated with the subject, object, requested operations, and environmental conditions [59].

ABAC operates on a set of policies that use attributes as the main building blocks. These attributes can represent various characteristics, such as user roles, department, time of access, and location. The combination of these attributes allows for a more dynamic and flexible access control mechanism compared to traditional models. The evaluation of permissions in ABAC is conducted through a policy decision point (PDP) that evaluates the attributes and makes access control decisions based on predefined policies. The policy enforcement point (PEP) then enforces

these decisions by granting or denying access accordingly. The significant advantage of ABAC is its flexibility and scalability. It allows organizations to create complex access control policies tailored to their specific needs, enhancing security by ensuring that only authorized users can access specific resources under certain conditions.

2.1.2 Next Generation Access Control (NGAC)

NGAC represents a fundamental reworking of traditional access control to better suit the needs of modern, distributed, and interconnected enterprises. The NGAC model conceptualizes access decision data as a Directed Acyclic Graph (DAG), illustrated in Figure 2.1. As per NIST docu-

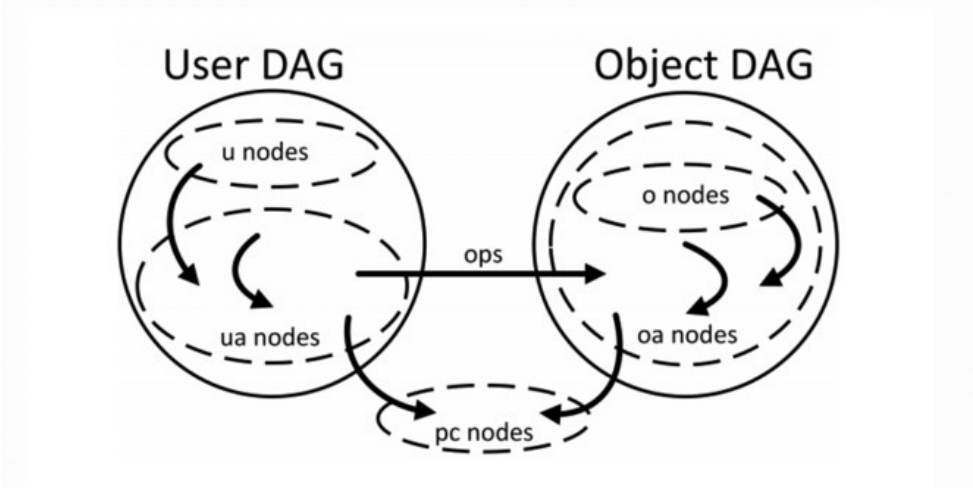

Figure 2.1: Directed Acyclic Graph (DAG) in the NGAC model [58]

mentation, the core policy entities that describe the NGAC model include:

- Users (U): Represents the actors interacting with the system.
- Objects (O): Represents the resources to be protected.
- User Attributes (UA): Represents the characteristics of a user.
- Object Attributes (OA): Represents the characteristics of an object.

- Access Rights (AR): Represents the required rights for users and objects, divided into Administrative Access Rights (AAR) and Resource Access Rights (RAR).
- Operation (OP): Refers to the actions that can be performed on the object, divided into Administrative Operation (AOP) and Resource Operation (ROP).
- Policy Classes (PC): Organizes and distinguishes between classes of user and object policies.

The NGAC model consists of basic entities and relations. The basic entities include users, processes, objects, operations, access rights, user attributes, object attributes, and policy classes [22] [23]. The user attributes, object attributes, and policy classes are containers.

NGAC does not express policy through rules but by using relations, *assignment*, *association*, *prohibition*, and *obligation* relations. The *assignment* relation connects users to user attributes, user attributes to user attributes, objects to objects attributes, object attributes to object attributes, user attributes to policy class, and object attributes to policy class. The *association* relation defines allowable access rights between user attributes and object attributes. The *prohibition* relations disallow access rights that have been granted through the association relation. The *obligation* is an event-pattern relation that tracks event notifications and automatically triggers a predefined response for those events that match a predefined pattern. The *obligation* relation is essential for constructing a dynamic NGAC security model [22].

2.2 Privacy Regulations

The General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) are two landmark pieces of legislation that have significantly reshaped the landscape of data protection and privacy rights in Europe and the United States, respectively. These regulations have set data protection, privacy, and user rights standards, imposing strict requirements on organizations regarding data collection, processing, and user consent. This section provides an overview of the GDPR and the CCPA.

2.2.1 General Data Protection Regulation (GDPR)

GDPR is a comprehensive data protection law enacted on May 25, 2018 in the European Union (EU). It aims to harmonize data protection laws across the EU member states and give individuals greater control over their data. According to Tankard et al. [62], GDPR is not limited to businesses within the EU. It has a broad scope and applies to all businesses, regardless of their location, that process the personal data of EU citizens. This means that even businesses based outside the EU that offer goods or services to EU citizens or monitor their behavior are subject to the GDPR's provisions. It is important for businesses to understand this scope to ensure they are compliant.

One of the GDPR's fundamental principles is the requirement that businesses obtain clear and explicit consent from individuals before collecting and processing their personal data. Consent must be specific, informed, and unambiguous. Alongside consent, the GDPR outlines several other data protection principles businesses must adhere to when processing personal data. These principles include lawfulness, fairness, transparency, purpose limitation, data minimization, storage limitation, security, and accountability.

The GDPR is designed to empower individuals with control over their personal data. It grants several rights to individuals, including the right to access their data, the right to rectify inaccuracies, the right to erasure (commonly known as the "right to be forgotten"), the right to data portability, and the right to object to the processing of their data. These rights are not just legal jargon, but expectations that businesses must meet to ensure GDPR compliance and maintain a positive relationship with their customers. Some businesses must appoint a Data Protection Officer (DPO) to oversee GDPR compliance and serve as a point of contact between the business, data subjects, and the supervisory authority. Additionally, the GDPR mandates that businesses must notify the relevant supervisory authority of a data breach within 72 hours of becoming aware of it unless the breach is unlikely to result in a risk to individuals' rights and freedoms [62].

Non-compliance with the GDPR can result in significant fines. Serious violations can lead to fines of up to 4% of the business's annual global turnover or €20 million, whichever is greater. Furthermore, the GDPR imposes restrictions on transferring personal data outside the EU to ensure

that the data is adequately protected. Businesses are required to implement appropriate technical and organizational measures to demonstrate compliance with the GDPR, including maintaining detailed records of data processing activities and adopting privacy by design and by default principles.

GDPR is not just a set of rules, but a crucial framework that businesses must adhere to. It imposes strict requirements to ensure the protection of personal data and the privacy rights of individuals. Compliance with the GDPR is not an option, but a necessity for businesses operating within the EU or targeting EU citizens. Non-compliance can lead to significant financial penalties and reputational damage, making it a risk that no business can afford to take lightly.

2.2.2 California Consumer Privacy Act (CCPA)

The California Consumer Privacy Act (CCPA), a pivotal state-level privacy law, came into effect on January 1, 2020, in California, United States. This legislation grants California residents specific rights regarding personal information and imposes significant obligations on businesses that collect and process this data. The CCPA has brought about a significant change by introducing several new consumer rights related to personal information.

The CCPA centers on the fundamental "right to know". Under this regulation, consumers are entitled to understand the categories of personal information that a business collects about them, the sources from which this information is obtained, the business purposes for collecting or selling the personal information, and the categories of third parties with whom the information is shared. According to this regulation, consumers have the following entitlements regarding their personal information [63]:

- Right to Opt-Out: Consumers can opt out of selling their personal information to third parties. Thus, businesses are required to provide a clear to facilitate their right and a conspicuous link on their websites titled "Do Not Sell My Personal Information."

- Right to Access: Consumers can request and receive a copy of the specific pieces of personal information that a business has collected about them over the past 12 months.

- Right to Deletion: Consumers have the right to request the deletion of their personal information collected by a business, with certain exceptions.

Therefore, by the CCPA, businesses must implement data protection measures, such as maintaining reasonable security procedures and practices to safeguard consumers' personal information. Additionally, businesses must provide consumers access to their personal information, honor opt-out requests, and delete personal information upon request [63]. Businesses can take several proactive steps to ensure they are in compliance with the CCPA's requirements. These steps include conducting data mapping exercises to understand the personal information they collect, where it is stored, and with whom it is shared; updating privacy policies to incorporate the necessary disclosures and consumer rights; implementing procedures to efficiently respond to consumer requests; and conducting training sessions for employees on CCPA compliance.

2.3 Natural Language Processing (NLP) Techniques

In this section, we introduce the NLP techniques utilized in this dissertation, including *SpaCy*, *SRL*, *BERT*, and *SetFit*. The following subsections provide an overview of each technique.

2.3.1 SpaCy Library

The *SpaCy* library [32] has undergone significant advancements through the integration of deep learning techniques. This library offers a range of linguistic features, including tokenization, lemmatization, Part-of-Speech (POS) tagging, syntactic dependency parsing, and Named Entity Recognition (NER).

When processing a text with *SpaCy*, the input undergoes a series of steps in a pipeline to produce a tokenized document. Initially, the tokenizer breaks the text into tokens. Subsequently, the POS tagger assigns grammatical categories to each token, such as nouns, verbs, adjectives, or adverbs. The parser then constructs a syntactic dependency for each sentence, representing the relationships between words as a navigable tree. This allows for the exploration of how words relate to each other within the sentence's structure. Additionally, the lemmatizer operates to convert

each word to its base or dictionary form, known as its lemma. For instance, it transforms the word "prohibiting" to its base form, "prohibit," aiding in normalization and analysis. Lastly, the Named Entity Recognizer (NER) component identifies various named and numeric entities within the text, such as companies, locations, organizations, and products. These features facilitate the extraction of crucial entities, thereby advancing our aim of extracting and analyzing security and privacy policies.

2.3.2 Semantic Role Labeling (SRL)

Semantic Role Labeling (SRL) stands out as a technique for the shallow semantic parsing of natural language texts, generating predicate-argument structures for sentences [30]. In the semantic theory, predicates encompass verbs, verbal nouns, and select other verb forms. Complementing predicates are arguments, representing phrases that fill meaning slots within a situation described by a predicate, elucidating its essential details. These details address questions such as "who?", "did what?", "to whom?", "with what?" "where?" and "when?" [30]. By assuming semantic roles, arguments play integral parts in defining the meanings of specific slots, with role meanings and inventories varying across semantic theories and annotated corpora [38]. Transforming a text into these shallow semantic structures proves valuable as it allows abstraction from syntactic and morphological representations of sentences. This abstraction is deemed essential in natural language understanding, facilitating a clearer focus on the underlying semantic content of the text.

SRL arguments labeled for the verbs are numbered arguments like ARG0, ARG1, ARG2, and so on. The numbered arguments represent semantic roles related to the predicate. In most cases, the meaning represented by the arguments is the same across different sentences, where the meaning carried by certain arguments as per the prop bank annotation [52]. Apart from these numbered arguments, there are argument modifiers ARGM, which carry functional tags like MNR for manner, LOC for locative, TMP for temporal, and many others, as shown in Table 2.1.

Table 2.1: SRL arguments

Args	Description
ARG0	Agent
ARG1	Patient
ARG2	Instrument, attribute, benefactive
ARG3	Starting point, attribute, benefactive
ARG4	Ending point
ARGM-TMP	When?
ARGM-LOC	Where?
ARGM-DIR	Where to/from?
ARGM-MNR	How?
ARGM-PRP/CAU	Why?

2.3.3 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained deep learning model introduced in 2018 . It has revolutionized the field of the NLP by achieving state-of-the-art results on a wide range of NLP tasks. BERT is based on the Transformer architecture and is designed to understand the context and semantics of words in a sentence by capturing the bidirectional dependencies between them [17].

Below are the advanced key characteristics of BERT [17]:

- Transformer Architecture: BERT is built on the transformer architecture, which uses self-attention mechanisms to weigh the importance of different words in a sentence based on their context.

- Masked Language Model (MLM): Randomly masks some words in a sentence and trains the model to predict the masked words based on the surrounding context.

- Next Sentence Prediction (NSP): Predicts whether a sentence is the next sentence in the original text or a randomly selected sentence.

- Contextual Word Embeddings: Unlike traditional word embeddings like Word2Vec or GloVe, BERT produces contextual word embeddings, meaning the embedding of a word depends on its context in the sentence.

2.3.4 Generative Pre-trained Transformer-3 (GPT-3)

OpenAI, an AI research lab, introduced the GPT series in 2018 and has unveiled three versions: GPT-2, GPT-3, and GPT-4 [50]. GPT-3, the third iteration of OpenAI's GPT model, boasts impressive training with 175 billion parameters, marking a substantial enhancement from its earlier versions. It leverages a diverse range of OpenAI texts, encompassing Wikipedia articles and the expansive open-source Common Crawl dataset [42].

The primary features of GPT-3 are as follows:

- Scale and Size: GPT-3 is one of the most significant language models to date, consisting of 175 billion machine learning parameters. This considerable scale allows the model to capture and understand intricate patterns and nuances in the data, enabling it to generate more accurate and contextually relevant responses.

- Few-Shot and Zero-Shot Learning Capabilities: GPT-3 exhibits strong few-shot and zero-shot learning capabilities. It can perform tasks with minimal examples or prompts (few-shot learning) and generate responses to tasks it has not been explicitly trained on (zero-shot learning), showcasing its ability to generalize across different tasks and domains.

- Adaptability: GPT-3 can be fine-tuned on specific tasks with minimal labeled data, allowing it to adapt to various NLP tasks and applications. This adaptability and fine-tuning capability make it a flexible and versatile model suitable for multiple applications and research studies.

2.3.5 Sentence Transformer Fine-tuning (SetFit)

SetFit, introduced by Tunstall et al. [64], is a cutting-edge model in the field of few-shot learning. *SetFit* employs a self-supervised pre-training approach, training the model to predict missing or masked tokens in a given text without relying on external labels or prompts.

SetFit is an efficient and prompt-free algorithm designed for few-shot fine-tuning of Sentence Transformers (ST). These Sentence Transformers are modifications of pretrained transformer models that incorporate Siamese and triplet network structures to produce semantically meaningful sentence embeddings. The primary goal of these models is to minimize the distance between pairs of semantically similar sentences and maximize the distance between sentences that are semantically distant. The primary advantages of *SetFit* are as follows:

- Simplicity and Efficiency: *SetFit* simplifies the few-shot learning process by eliminating the need for prompts or task-specific architectures, making it more efficient and easier to implement.

- Adaptability: By leveraging self-supervised pre-training and fine-tuning , *SetFit* enables the model to adapt better across different few-shot learning tasks and domains.

- Reduced Computational Overhead: *SetFit* reduces the computational complexity and resource requirements associated with traditional few-shot learning methods by leveraging self-supervised pre-training and efficient fine-tuning techniques.

Chapter 3

Literature Review

Previous efforts have been made to address the complex task of automating access control and privacy policy extraction and analysis. This dissertation chapter explores relevant literature that specifically focuses on the automated specification and analysis of ACPs and privacy policies. This chapter provides a comprehensive analysis and summary, including a comparative evaluation of various approaches taken by different studies. Subsequently, we pinpoint the existing research gaps identified through the literature review. The discussion of the related work is organized chronologically to present a historical perspective on the evolution of research in this domain.

3.1 Access Control Policies Specification and Analysis

In this section, we concentrate on the research initiatives related to the ACPs, which involve an in-depth exploration of their *specification* (the process of extracting ACPs) and *analysis* (the evaluation and assessment of ACPs). The emphasis is on providing a comprehensive overview of the most recent scholarly efforts in understanding and advancing the field of specification and analysis of the ACP.

3.1.1 Access Control Policy Specification

Xiao *et al.* [71] introduce an approach called *Text2Policy* to extract ACP sentences from natural language documents. This research marks the initial endeavor of highlighting the challenges in the automated extraction of the ACPs that are written in natural language, such as security requirements documents. *Text2Policy* employs NLP techniques to automatically extract ACPs and action steps from natural language documents and generate formal specifications. The three main steps of this approach involve linguistic analysis, construction of model instances, and transformation into formal specifications. The approach is based on eXtensible Access Control Markup Language (XACML), a machine-enforceable language designed to implement ABAC. *Text2Policy*

motivated researchers to use NLP to extract ACPs from security requirements in a natural language. This research discusses technical challenges such as anaphora resolution, semantic structure variance, negative-meaning implicitness, transitive actor issues, and perspective variance, proposing corresponding techniques to address these challenges. The authors also present evaluations of *Text2Policy* on the iTrust dataset [44], demonstrating its effectiveness in identifying ACP sentences and extracting ACP rules and action steps with high accuracy. However, the *Text2Policy* is limited to a fixed set of predefined access control patterns and cannot automatically learn new ACP patterns. It also produces incomplete ACPs when conditions exist in the document specifications.

Slankas *et al.* [60] introduce an approach, Access Control Rule Extraction *(ACRE)*, designed to enable organizations to infer Access Control Rules (ACRs) from existing, unrestricted natural language documents such as security requirements documents. *ACRE* integrates NLP and Information Extraction (IE) techniques. Internally, *ACRE* represents natural language sentences using a type dependency parse graph, depicting words as vertices and their relationships as edges. *ACRE* identifies a basic noun-verb-noun pattern in graphs and creates a frequency table for discovered subjects and direct objects. Subjects or direct objects occurring more frequently than the median count are assumed to be legitimate for the system. *ACRE* then explores all combinations of these subjects and objects, extracting Minimal Spanning Trees (MSTs) and associating verbs within the trees with corresponding actions. The approach refines its ACR patterns by expanding the set to include wildcarded subjects or objects matched with words of the same part of speech in other sentence graphs. *ACRE* searches for instances of these patterns across all sentences, assuming found instances represent ACRs, and extracts their elements. To minimize erroneous patterns, ACRE employs a naïve Bayes classifier based on domain-independent features found in other documents. The evaluation results demonstrate ACRE's effectiveness in identifying ACR sentences with 81% precision and 65% recall while extracting ACRs from identified sentences with an average precision of 76% and an average recall of 49%. However, the proposed approach faces challenges when sentences contain events and response patterns. In other words, *ACRE* is limited in detecting policies involving conditions, such as obligations policies.

Narouei et al. [46] propose an SRL approach to identify ACP elements in natural language sentences. The proposed methodology involves the following steps: (1) lexical parser to identify and separate sentences using sentence segmentation and tokenization tools; (2) coreference resolution to determine whether different expressions in a document refer to the same entity or event; (3) ACP sentence identification to identify sentences containing ACP content using a (k-NN) classifier; (4) semantic parser that utilizes the SRL to automatically identify predicate-argument structures in ACP sentences, which allows for the extraction of ACP elements such as subject, object, and action; (5) post-processing to refine the extracted ACPs to increase specificity. This proposed approach utilizes the ACP components extracted by the SRL tool to define new policies for different access control models, such as RBAC or ABAC. Nevertheless, it is essential to note that individual ACPs encompass multiple attributes about the subject, action, and object. However, the SRL faces limitations in identifying the specific attributes associated with these crucial elements.

Alohaly *et al.* [3] propose a deep learning-based automated approach to extract ABAC attributes from natural language policy sentences. The authors developed a framework that leverages NLP, Relation Extraction (RE), and Convolutional Neural Networks (CNN) to automate attribute extraction. The framework locates the modifiers of each policy element by analyzing the security requirements, extracting the attribute value and corresponding policy element, and identifying the attribute's category, short name, and data type. The authors use CNN to capture subject-attribute, object-attribute, and element-specific relations. The proposed framework resulted in an average F1-score of 85% for extracting the subject's attribute values and 71% for extracting object attribute values. However, the framework's accuracy depends on the Named Entity tagger's performance, which can lead to potential inaccuracies in attribute data type determination.

Abu Jabal *et al.* [1] propose a framework called Polisma to learn ABAC policies from different resources, such as a log of access requests and corresponding access control decisions and context information provided by external resources such as Lightweight Directory Access Protocol (LDAP) directories. The approach uses data mining and machine learning techniques to generate the ABAC rules. The Polisma approach excels in acquiring ABAC rules from diverse sources,

achieving an 80% F1-score on established data sets. While Polisma emphasizes extracting ABAC policies from various resources, our approach explicitly targets the extraction of NGAC ACPs from security requirements articulated in natural language documents. Furthermore, a notable limitation of Polisma is its inability to effectively address conflict, where conflict resolution plays a pivotal role in access control policy management.

Xia *et al.* [70] introduce a methodology that employs the predicate arguments structure of the SRL integrated with Bidirectional Encoder Representations from Transformers (BERT) to identify the ABAC rule structure (subject, action, and object). The SRL predicate arguments (expressed as ARG_n) include ARG0 (represents the entity acting), VERB (represents the action itself), and ARG1 (represents the entity toward which the action is directed). The authors utilized SRL to label subject and object attribute values from ARG0 and ARG1. They also employ ARGM-TMP (represents time) and ARGM-LOC (represents location) to identify environment attributes. The process involves two main modules: ACP identification and ABAC Rule Extraction. For ACP identification, SRL architecture is built on BERT to recover predicate-argument structures in sentences, assigning correct semantic role labels to argument spans. The proposed approach achieves an 85% F1-score for ACP identification and a 72% F1-score for rule extraction. However, the generalized architecture of SRL poses challenges in accurately labeling attributes related to security policies.

3.1.2 Access Control Policy Analysis

Ray *et al.* [56] present a formal spatio-temporal model suitable for commercial applications, expanding upon the RBAC model. The authors extend RBAC to incorporate time and location dimensions as it is policy-neutral and simplifies access management. The correct behavior of the proposed model is specified in terms of constraints that any application using this model must satisfy. This work utilizes the Alloy analyzer tool for formal modeling and analysis. The proposed access control model allows permissions to be derived using spatial and temporal attributes. Introducing spatial-temporal information may lead to conflicts within the access control policy. The authors show how formal analysis of this access control model can identify such conflicts.

However, the complexity of the proposed model makes it challenging to implement or scale in real-world applications.

Fernández *et al.* [21] propose a method of specifying and analyzing ABAC policies in first-order logic using a Category-Based Meta-Model of Access Control (CBAC). The authors introduce a meta-model that categorizes and represents the components of ABAC policies, providing a structured framework for policy specification. The proposed meta-model includes various components of ABAC policies, including subjects, objects, actions, and conditions. It defines categories within each element, allowing for a more granular and organized representation. The proposed approach allows for the evaluation and verification of ABAC policies to ensure correctness. This study considers the following properties for security analysis: totality, consistency, soundness, and completeness. The formal methods facilitate rigorous verification and validation of policy specifications, ensuring they adhere to security requirements. The category-based meta-model is a foundation for creating a formal model that enables systematic analysis of ABAC policies. However, like others, implementing or scaling formal model analyses presents considerable challenges in real-world applications..

Chen *et al.* [13] present an approach to analyzing the quality assurance of NGAC policies. This work introduces a method for conducting mutation analysis on NGAC policies to assess the efficacy of testing techniques in uncovering potential faults within the policies. The approach automatically generates a series of policy mutants from a given NGAC policy, each subject to a designated test suite representing a testing method under evaluation. A mutant is essentially a modified version of the original policy, wherein a specific policy element is altered based on a fault model encompassing various types of errors commonly found in NGAC policies. A mutant is classified as "killed" if it fails one or more tests; otherwise, it is considered a live mutant. The testing effectiveness is gauged by the mutation score, calculated as the ratio between the number of mutants killed and the total count of non-equivalent mutants. This work demonstrates how the mutation analysis effectively revealed various faults and obligations during the initial configuration of

the NGAC. Despite the complexity of the proposed testing approach, it exhibits limited scalability for initial configurations and falls short in detecting the majority of obligation faults.

3.2 Privacy Policy Specification and Analysis

In this section, our focus is directed toward research endeavors concerning privacy policy. This entails a thorough investigation into the identification of privacy data practices, which involves extracting the data practices employed by data collectors. Additionally, we delve into analyzing these practices, encompassing the evaluation and assessment of privacy policies. The objective is to offer a comprehensive overview of the latest scholarly initiatives aimed at comprehending and advancing the domain of privacy data practices and analyzing privacy policies.

3.2.1 Privacy Policy Specification

Sunkle et al. [61] use specialized vocabularies and semantic similarity models to translate regulatory requirements into organizational operational procedures. They introduced the Semantics of Business Vocabulary and Rules (SBVR) model, which consists of several stages: (1) creating a business context vocabulary by defining semantic communities and sub-communities affected by regulations; (2) identifying critical terms within regulatory rules, where noun concepts represent entities or categories, verb concepts illustrate behaviors involving noun concepts, and binary verb concepts depict relations between two concepts; (3) constructing logical expressions for each policy outlined in the regulations; and (4) developing a terminological dictionary model that includes diverse representations used by semantic communities for concepts and regulations. However, the proposed approach relies on semi-automated processes for modeling concept vocabularies, which encounters challenges in scalability and adaptability. As regulatory landscapes evolve and organizational processes change, maintaining and updating the concept vocabularies to reflect these changes could be labor-intensive, requiring ongoing monitoring and revision. Additionally, their approach is based on semantic similarity between texts, which may not fully capture the nuances of regulatory language and operational details.

Elluri et al. [19] present an automated compliance framework to align cloud service providers' privacy policies with the GDPR. The research aims to establish semantic similarity between GDPR guidelines and organizational policies, thereby creating a comprehensive knowledge representation. The authors used the Doc2Vec approach to derive similarity scores and extract semantically similar keywords. Initially, the authors retrieved controller, processor, and common obligations from GDPR and collected privacy policies. Subsequently, they analyzed twenty privacy policies by computing similarity scores. They then constructed a comprehensive data compliance comparison ontology using the Protegé tool (an open-source ontology editor), effectively encapsulating similarity scores and semantically similar keyword details derived from regulation and organizational privacy policies. Finally, the authors validated the knowledge graph by cross-referencing the semantically similar keywords extracted from privacy policies with the obtained similarity scores. These similarity scores benefit organizations that do not adhere to GDPR rules, enabling them to revisit the regulation and incorporate any missing rules into their privacy policies. However, their approach relies on semantic similarity, which may not fully capture the intricacies of legal language and policy documents. The effectiveness of the semantic analysis techniques utilized in the research could vary depending on the length of documents, complexity of language, and contextual nuances.

Mousavi et al. [45] present an automated approach called *KnIGHT*, designed to extract pertinent information from privacy policies and align it with the corresponding GDPR articles [62]. *KnIGHT* analyzes privacy policies and GDPR articles to establish correlations at both the sentence and paragraph levels. The process begins with *KnIGHT* identifying candidate sentences using the Privacy Policy Gazetteer is a GATE pipeline [15]. This Gazetteer combines common preprocessing steps in NLP with a list of crucial terms extracted from privacy policies. It is created from an input text file generated through TermRaider (a term extraction tool) and an intermediate converter, resulting in annotations for tokens, sentences, and important terms. Sentences containing at least three important terms are flagged as candidates. Then, each candidate sentence is matched with the most relevant GDPR article, and the optimal paragraph match within that article is determined

using a semantic similarity-based approach. Finally, the most relevant GDPR article is retrieved, and the best paragraph match within it is identified. The authors evaluate the performance of *KnIGHT*, demonstrating its effectiveness in accurately identifying GDPR-related clauses within privacy policies. The automated mapping capabilities of *KnIGHT* indicate potential for enhancing compliance efforts. However, experts' ratings indicate that the mappings made by *KnIGHT* are not fully correct; some are partially correct. Nevertheless, it is essential to acknowledge the limitations of this evaluation, notably the absence of consideration for false negatives.

Hamdani et al. [28] present a method for automating compliance checking with the GDPR. The proposed method is to monitor GDPR compliance in the data supply chain through a document-centric approach. The proposed framework analyzes the compliance function of privacy policies and defines the tasks required to determine if a privacy policy complies with the GDPR. The authors construct a two-module system to assess the comprehensiveness of privacy policies concerning mandatory information. The initial module automatically (machine learning) extracts coarse-grained and fine-grained data practices. The second module (rule-based) examines the extracted data practices and verifies the inclusion of mandatory information as per GDPR provisions. Two experiments are conducted to predicate data practices. Firstly, a local multi-label classifier is constructed using CNN and XLENT algorithms. Secondly, the Hierarchical Multi-label Classification (HMTC) task is approached by transforming it into two text-to-text tasks, allowing for a more detailed analysis and prediction. Following this, the extracted information is input into a rule-based module. This component applies predefined rules derived from GDPR requirements to detect potential compliance violations. These predefined rules were constructed by legal experts who manually converted rules from Articles 13 and 14 into code. The authors choose JsonLogic to serialize the rules obtained as a JSON file. The authors encode 10 GDPR rules from Articles 13 and 14. The proposed approach was evaluated using a dataset comprising 30 privacy policies, where legal experts confirmed the presence of mandatory information. While the proposed approach showed promise in capturing some required information, its performance in extracting data practices needed to be sufficiently high, indicating significant room for improvement. On the other

hand, legal experts manually translated rules from GDPR articles 13 and 14 into code using the OPP-115 taxonomy. However, due to the incomplete coverage of GDPR concepts within the OPP-115 taxonomy, only a subset of the articles could be encoded. Therefore, there is a pressing need to encompass all GDPR rules within the taxonomy to ensure comprehensive coverage.

Amaral et al. [4] introduce an automated method for verifying the compliance of Data Processing Agreements (DPAs), crucial contractual agreements between data controllers and processors. Their approach is grounded in a comprehensive conceptual model comprising 63 distinct information types, encompassing all expected content elements within GDPR-compliant DPAs. This model consists of two key artifacts: artifact (A), containing 45 compliance requirements expressed in natural language about GDPR data processing, and artifact (B), delineating 45 common information types found in DPAs. The authors utilize DERECHA's machine-learning pipeline to evaluate compliance, which is short for DPA sEmantic fRamE-based Compliance cHecking Against GDPR. DERECHA processes a DPA, controller(s), and processor(s) names and requirements to produce a compliance report summarizing its findings. The process begins with creating representations of compliance requirements, which allows for detailed compliance checks at a granular level. These representations categorize each requirement in GDPR, distinguishing between metadata and explicitly stated requirements. Ten semantic roles relevant to DPA compliance are identified, including action, actor, and object. Each requirement is linked with a predicate representing the main action and a set of arguments representing associated participants. Subsequently, the authors employ NLP-based parsing and preprocessing of the DPA text, automatically generating a representation of the DPA content. The textual content is then enriched with semantically related terms. Finally, compliance with GDPR is assessed, resulting in a comprehensive compliance report detailing the findings. Evaluation findings from a dataset comprising 30 DPAs demonstrate the effectiveness of the proposed approach in accurately assessing DPAs for compliance. The approach exhibits proficiency in identifying GDPR violations, achieving a precision of 83.9% and a recall of 83.0%. However, it relies on a predefined conceptual model for compliance verification,

which, while comprehensive, may overlook certain nuanced or context-specific aspects of GDPR compliance.

3.2.2 Privacy Policy Analysis

Andow et al. [5] introduce *PolicyLint*, a tool designed to analyze privacy policies of mobile apps on Google Play and identify contradictions within them. The study focuses on contradictions that could lead to incorrect interpretations of data sharing and collection practices, revealing concerning trends through manual analysis of 510 contradictions across 260 policies. These contradictions are categorized into logical contradictions, which can cause harm if users are unaware of conflicting statements, which may lead to inconsistent decisions made by automated techniques. *PolicyLint* analyzes the policies of 11,430 apps. *PolicyLint* reveals that 14.2% of them contain contradictions, ranging from misleading presentations to redefining standard terms and conflicts in regulatory definitions. However, *PolicyLint* cannot distinguish between entities responsible for data collection or sharing, which could lead to inaccurate analyses of privacy policies.

Andow et al. [6] delve into the detection of privacy-sensitive data leaks within the data flows of mobile applications called *Policheck*. They highlight that data flows, if disclosed in privacy policies, are not considered "leaks." Recent research has merged program analysis with privacy policy analysis to evaluate flow-to-policy consistency and identify violations. The authors formally define a flow-to-policy consistency model for mobile apps, encompassing two types of consistencies (clear and vague disclosure) and three types of inconsistencies (omitted, ambiguous, and incorrect disclosure). *Clear disclosure* refers to a data flow disclosed when a privacy policy statement directly addresses the exact type of data and entity involved in the flow without any contradictory statements in the policy. *Vague disclosure*, on the other hand, occurs when the statements in a privacy policy matching a data flow use broad terms for the data type or entity, lacking specificity. *Omitted disclosure* indicates a data flow that lacks any policy statements discussing it, leaving it undisclosed in the privacy policy. *Ambiguous disclosure* arises when a data flow matches two or more contradictory policy statements, leading to uncertainty about whether the flow will occur.

Incorrect disclosure denotes a data flow inaccurately disclosed in the policy, such as when a policy statement wrongly indicates that the flow will not occur. *Policheck* is deployed to analyze 13,796 applications and their privacy policies, revealing that 42.4% of applications either inaccurately disclose or omit details regarding their privacy-sensitive data flows. *Policheck* exhibits higher precision in accurately determining whether applications disclose their privacy-sensitive behaviors. However, *Policheck* faces challenges in extracting complete policy statements and fails to capture policy statements spanning multiple sentences, treating each paragraph's policy separately.

Bui et al. [11] present *PurPliance*, which is an automated approach that detects inconsistencies between the stated purposes in an app's privacy policy and the actual data usage behavior within the app. The authors analyze the predicate-argument structure of policy sentences and categorize purpose clauses into a taxonomy of data purposes. *PurPliance* then infers actual data usage from network data traffic and utilizes a formal model to represent and verify data usage purposes in privacy statements and data flows. *PurPliance* involves analyzing semantic arguments and syntactic structures of data practices. Thus, *PurPliance* utilized SRL to uncover the latent predicate-argument structures of sentences. Then, *PurPliance* categorizes purpose clauses by identifying patterns of n-grams (such as words and bigrams) on lemmas of predicates and objects in the pairs of purpose clauses. As observed by the authors, these patterns are context-insensitive, allowing for precise classification when matching such n-gram patterns. Evaluation results demonstrate that *PurPliance* significantly enhances detection precision (from 19% to 95%) and recall (from 10% to 50%) compared to a state-of-the-art method. Analyzing 23.1k Android apps, *PurPliance* detects contradictions in 18.14% of privacy policies and flow-to-policy inconsistencies in 69.66% of apps. These findings underscore the prevalence of inconsistencies in data practices among mobile apps and the importance of automated tools for effective detection. Therefore, the research emphasizes the importance of verifying the consistency of privacy policy statements with legal regulations.

3.3 Research Gap

After reviewing the existing literature, I have pinpointed several significant research gaps that demand further attention:

- **Lack of research on the NGAC policy extraction**: Current research lacks sufficient exploration into the extraction of the NGAC policies. NGAC represents a modern approach to access control that adapts to evolving technological landscapes and security requirements. The absence of comprehensive studies in this area hinders the understanding and implementation of advanced access control mechanisms.

- **Absence of an automated framework for analyzing extracted access control models**: Automated frameworks need to be designed to analyze the correctness of access control models extracted from natural language documents.

- **Opportunities for enhancing identification of security policy attributes in natural language sentences**: Recent advancements in NLP techniques present a hopeful prospect for enhancing the identification of security policy attributes within natural language sentences. By harnessing these state-of-the-art NLP methodologies, we can develop more effective algorithms for extracting and interpreting security-related information from textual data.

- **Opportunities for enhancing the extraction of data practices in the privacy policy in documents**: The complexity of privacy policy documents presents challenges for end users and developers. These documents are often lengthy and filled with dense language, legal terms, and lengthy clauses, making it difficult for individuals to understand their rights and obligations regarding data privacy. By leveraging cutting-edge NLP techniques, we can create more efficient algorithms for extracting data practices from privacy policies.

- **The significance of distinguishing between first-party and third-party entities in privacy data practices**: A crucial gap exists in the differentiation between first party and third party entities in the context of privacy data practices. This distinction is vital for a com-

prehensive understanding of data processing and sharing. However, prior research [5] has demonstrated a lack of clarity in defining these roles. This lack of differentiation led to inaccuracies in their analyses. By treating all entities involved in data processing similarly, regardless of whether they are first party or third party entities, these studies fail to fully grasp the nuances and implications of data handling practices. Consequently, their conclusions may not accurately reflect the privacy implications for users.

- **Absent of identifying the type of data and purpose**: Previous studies [5, 6, 29] have not sufficiently defined the type of the purposes for sharing data or the types of data collected by organizations. Establishing clear categories for data-sharing purposes and determining the types of data collected would prove advantageous for both end users and developers. End users would gain a clearer insight into the data collected from them and its intended purposes, thereby enabling them to make informed decisions regarding their privacy and engagement with the service or platform. Similarly, developers stand to benefit from understanding the specific types of data being collected and the purposes for which it is being used, developers can design systems that prioritize user privacy.

Efforts to fill these gaps can be advanced by enhancing automated analysis systems. This involves designing specialized algorithms capable of handling security and privacy policy documents, employing advanced natural language processing techniques and machine learning models trained on diverse security and privacy policies to improve accuracy.

Chapter 4

Policy Language Linguistics Analysis

In this dissertation, we study the automation process of security and privacy policy specification and analysis. We start this research by investigating the linguistic patterns in policy sentences. Then, we study how we can leverage those patterns to automate the policy extraction from the natural language documents. In this chapter, we delve into a comprehensive linguistic analysis of policy sentences, employing the advanced capabilities of a *Dependency Parser* within the NLP field.

In this chapter, we explore the *Dependency Parsing*, explaining its concept in Section 4.1. Following this, we test two approaches within the *Dependency Parsing*: Namely, SpaCy (discussed in Section 4.3) and SRL (detailed in Section 4.4). Then, we present the results and discuss the limitations and the valuable lessons from the analysis.

4.1 Dependency Parser

Dependency Parsing is the process of analyzing the grammatical structure of a sentence to identify related words and determine the type of relationship between them [48, 49]. *Dependency Parsing* refers to examining the dependencies between the phrases of a sentence to ascertain its grammatical structure. The sentence is segmented into various sections primarily based on these dependencies. This process operates under the assumption that there exists a direct relationship between each linguistic unit within a sentence, with these connections termed dependencies [48].

Consider the following statement: "This document is a PhD dissertation." In a written dependency structure, the relationships between each linguistic unit, or phrase, in the sentence are expressed by directed arcs, which serve as tree graphical representations of the syntactic dependencies between words as shown in Figure 4.1

These arcs reveal various syntactic relationships within the sentence, such as subject-verb relationships, modifiers, and complements. For instance, in the above example, the arc from "disser-

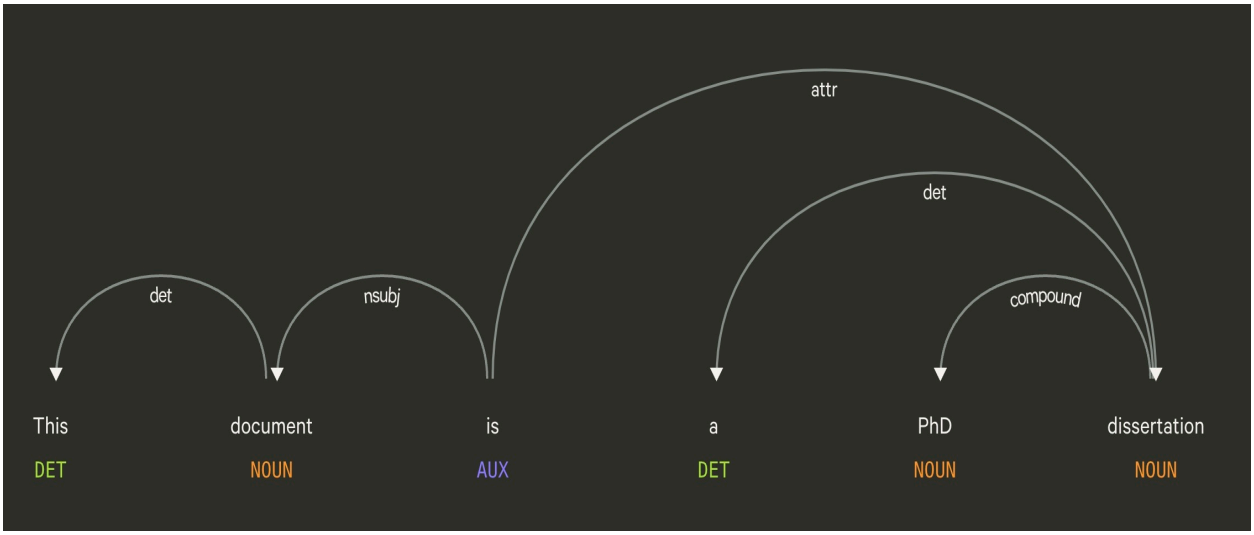

Figure 4.1: Dependency Parser Tree for Sentence "This document is a PhD dissertation"

tation" to "PhD" indicates that "dissertation" modifies "PhD," illustrating the relationship between the two words. By analyzing these dependencies, linguists and natural language processing systems can uncover the hierarchical structure of the sentence and discern how each word contributes to its overall meaning. This process is crucial for understanding the syntactic organization of language and extracting semantic information from text.

Furthermore, each dependency in the parsing process is labeled with a specific tag, which indicates the nature of the relationship between the associated phrases. For example, in the dependency "PhD -> dissertation," the tag "Compound" signifies that "dissertation" modifies "PhD" implying that "PhD" describes the type or nature of the dissertation. These dependency parsing tags help categorize and understand the various syntactic relationships present within the sentence. The meaning of the dependency parsing tags that are in the above example is shown in Table 4.1

Overall, dependency parsing plays a vital role in linguistic analysis and natural language processing, enabling researchers and systems to dissect and comprehend the intricate structure of sentences and texts. Significant progress in NLP has been made, driven by advancements in the dependency parser. This has led to the development of tools like *SpaCy* [32] and *Semantic Role Labeling (SRL)* [43], both stemming from improvements in the dependency parser. These tools, utilizing deep learning techniques, play a crucial role in extracting detailed linguistic information.

Table 4.1: Dependency Parser Tags

Tag	Description
DET	determiner
nsubj	nominal topic
NOUN	noun phrase
compound	compound phrase

We employ the NGAC model as the formal model for the ACPs extraction process, it becomes imperative to elaborate on how *SpaCy* and *SRL* extract NGAC ACPs from natural language sentences. This extraction process entails identifying two main components: NGAC entities and NGAC relations.

In this chapter, we delve into the linguistic analysis of policy sentences, leveraging the capabilities of *SpaCy* and *SRL* for processing policy language. Section 4.3 and Section 4.4 conduct a comprehensive linguistic analysis of policy sentences using *SpaCy* and *SRL*. Finally, we discuss the limitations of both approaches to provide a comprehensive understanding of their applicability and constraints in policy language processing.

4.2 Dataset

Examining the dependency parser approaches (*SpaCy* and *SRL*) requires test cases of ACPs. We utilized the iTrust dataset test cases [44, 71] for their widespread use and public availability. For simplicity, we selected four test cases from the *iTrust* dataset to demonstrate the types of NGAC relations scenarios. Table 5.12 illustrates these ACPs (sentences 1-4). To demonstrate users accessing the objects, we added three sentences, shown in Table 5.12 sentences 5-7, that assign two users (Bob, Jack, and Alice) as a doctor, HCP, and nurse, respectively. We also added two sentences (7 and 8) to demonstrate objects assigned to object attributes (John and John.Smith.Record).

Table 4.2: Access Control Policies and Users

$iTrust$: **Access Control Policies**	
1	An HCP creates patients.
2	Doctor is an HCP.
3	A nurse is prohibited from creating patients.
4	Whenever a HCP changes a patient record, an email must be sent to the administrator.
$Users$	
5	Bob is a doctor.
6	Jack is an HCP.
7	Alice is a nurse.
$Object$	
9	John.Smith.Record is a patient record.
9	John is a patient.

4.3 Access Control Policy in Terms of SpaCy

To identify the NGAC policies (entities and relations) using *SpaCy* which is based on the convolutional neural network (CNN) model, we follow a two-step process. Firstly, we scrutinize the tokens and tags of the ACP sentence to pinpoint the subject, predicate, and object. Next, we analyze the syntactic dependency of the ACP sentence to ascertain the NGAC entities and relations in the following manner:

Method.

- **NGAC entities:** To identify the NGAC entities, we utilize a tokenized ACP that includes POS tags and a dependency tree or sub-tree as input. The process involves iterating over each token in the ACP to extract the root verb, subjects, and objects. If a token's dependency is "ROOT" and the POS tag is "VERB," it indicates that the token points to the root verb of the sentence, and it is extracted as a "predicate." Similarly, if a token's dependency is "subj"

and the POS tag is "PROPN" or "NOUN," it indicates that the token points to a subject and is extracted and added to a "subjects" list. The *"subjects"* list carries two types of elements: proper noun-subjects, which represent users, and common noun-subjects, which represent user attributes. Suppose the token's dependency is *"obj"* and the POS tag is *"PROPN"* or *"NOUN"*; it means that the token points to an object and it is extracted and added to a *"objects"* list. *"object"* list carries two types of elements: proper noun-object, which represent object, and common noun-object, which represent object attributes.

- **NGAC Obligation**: If the syntactic dependency includes an adverbial clause, the ACP sentence is marked as an obligation relation with two parts (event and response). The part containing the root verb, the independent clause, represents the response, while the other part, the conditional clause, corresponds to the event.

- **NGAC Prohibition**: If the syntactic dependency contains a negation pattern (e.g., a user cannot view, or a user is not allowed to view an object), the ACP sentence is marked as a prohibition relation. If the negation pattern is not found, the root verb is checked against a list of prohibition words (prohibit, deny, disallow, block, ban, constrain, forbid, halt, restrain, restrict, inhibit, impede, prevent, proscribe, refuse, reject, revoke, rebuff, oppose, refute, discard, disclaim, disprove, dismiss, decline, retract). The ACP sentence is marked as a prohibition relation if the root verb matches any.

- **NGAC Association**: If the ACP sentence is not marked as obligation or prohibition, it is examined as an association relation. Syntactic dependency is used to identify the association relation of the sentence entities in the form (subject, predicate, object), whenever the sentence is active voice.

- **NGAC Assignment**: If the syntactic dependency shows that the ACP sentence describes a subject or object (e.g. Tom is a user), it is marked as an assignment relation.

Table 4.3: Extracting NGAC Policy using the *SpaCy* Approach

	Access Control Policy Sentence	NGAC Policy
[1]	An HCP creates patients.	HCP–(create)–patient
[2]	Doctor is HCP.	Doctor–(assign to)–HCP
[3]	A nurse is prohibited from creating patients.	(nurse, $deny\_create$, patient)
[4]	Whenever a HCP changes a patient record, an email must be sent to the administrator.	(HCP–changes–patient record) (email–send)

Results.

Table 4.3 presents the output generated by *Spacy*, utilizing the iTrust dataset [44].

In ACP 1, "create" is the root verb indicating the action of granting access, "HCP" is the user attribute representing the user being granted access, and "patient" represents the object attribute being accessed. Since there are no adverbial clauses, negation patterns, or assignment descriptions present in the sentence, it is treated as an NGAC association relation. The association relation is:

$$(Jack \rightarrow HCP \rightarrow [create] \rightarrow patient \rightarrow John)$$

In ACP 2 sentence: "*Doctor is HCP."* "is" is the root verb indicating the assignment relation, "Doctor" is the user attribute, and "HCP" represents the user attribute. This is treated as an NGAC assignment relation. The assignment relation is:

$$(Doctor \rightarrow [assign\_to] \rightarrow HCP)$$

In ACP 3 sentence: "Doctor is prohibited from creating patients." Since the verb "prohibited " is one of the prohibition verbs listed, we consider this ACP as a prohibition relation; the prohibition relation is:

$$(Alice \rightarrow nurse \rightarrow deny\_create \rightarrow patient \rightarrow John)$$

In ACP 4 sentence: "Whenever an HCP changes a patient record, an email must be sent to the administrator." Since the presence of the adverbial clauses, the obligation relation can be represented as:

$$(Jack \rightarrow HCP \rightarrow change \rightarrow patient\ record \rightarrow JohnSmithRecord); (email \rightarrow be\ sent)$$

4.4 Access Control Policy in Terms of SRL

To identify the NGAC policies (entities and relations) using *SRL*, we employ *AllenNLP* [26], an open-source natural language processing platform, features a pre-trained SRL model, which is a re-implementation based on a deep Bidirectional Long Short-Term Memory (BiLSTM) model. This SRL model is designed to recover the latent predicate-argument structure of a sentence, contributing to *AllenNLP's* suite of reference implementations for various NLP tasks [27].

The choice of *AllenNLP* is rooted in its utilization of a transition-based neural network for identifying semantic roles in input sentences. This architecture has demonstrated state-of-the-art performance across various SRL benchmarks [27]. Leveraging AllenNLP assists us in discerning the predicate-arguments (Args) structure of a sentence and mapping them to NGAC entities in our context. The ACP in terms of SRL is the following pattern ($Arg0 \rightarrow V \rightarrow Arg1$) where $ARG0$ represents the agent or subject, V represents the main verb, and $ARG1$ represents the patient or Object.

Method.

- **NGAC Entities**: To identify the *Verb*, we look for tokens tagged as verbs in the POS tagging step. These tokens represent the main verbs of the sentence. To identify the *Subject*, we search for the nominal phrase (NP) or noun phrase that typically precedes the verb and performs the action described by the verb. We use dependency parsing information to find the word that is the subject of the verb. The subject usually has a syntactic relationship with the verb, such as being the head of a subject-dependent clause. To identify the *Object*, we look for nouns or noun phrases directly affected by the verb's action. These typically

follow the verb. We use dependency parsing information to find the word or phrase that is the object of the verb. The object usually has a syntactic relationship with the verb, such as being dependent on a direct object dependency relation.

- **NGAC Association**: We define a pattern between a user attribute, access rights, and an object attribute. For example, a user (identified by their attribute) may have certain access rights (e.g., read, write) to a specific object (identified by its attribute). The pattern for NGAC association can be represented as (user attribute–access rights–object attribute). In terms of Semantic Role Labeling (SRL), we can represent this pattern as:

  $user$ ($Arg0 \rightarrow V \rightarrow Arg1$); where $Arg0$ represents the agent or subject (user), V represents the main verb, and $Arg1$ represents the patient or object.

- **NGAC Prohibition**: We specify that a user is denied certain access rights to a resource. The pattern for NGAC Prohibition can be represented as $user$ _ $deny$ (user attribute–access rights–resource attribute). In terms of SRL, this pattern can be represented as $user$ _ $deny$ ($Arg0 \rightarrow V \rightarrow Arg1$). Where $Arg0$ represents the agent or subject (user), V represents the main verb (deny verb), and $Arg1$ represents the patient or object.

- **NGAC Obligation**: We establish an obligation pattern based on an event pattern and a corresponding response. For example, when a specific event pattern occurs, there is an obligation to respond in a certain way. The pattern for NGAC Obligation can be represented as (Event Pattern (EP), Response (R)). In terms of SRL, this pattern can be represented as:

  ($ARG:TMP$, $ARG:ADV$): $When(TMP) \rightarrow ADV$ ($Arg0 \rightarrow V \rightarrow Arg1$) where $ARG:TMP$ represents the event pattern, $ARG:ADV$ represents the response, $When\ (TMP)$ represents the temporal aspect of the event pattern, and ADV represents the manner of the response. $Arg0$ represents the agent or subject. V represents the main verb, and $Arg1$ represents the patient or object.

- **NGAC Assignment:** We define a pattern between a user and user attribute and an object and object attribute. To illustrate, a user is identified by their attribute, and an object is

identified by its attribute. The pattern for NGAC Assignment can be represented as (user- user attribute),(user attribute- user attribute), (object- object attribute), or (object attribute- object attribute). We identify a list of verbs (contain, includes, is) describing the NGAC assignment relation. In terms of SRL, we can represent this pattern as: $user$ $(Arg0 \rightarrow [contain, include, is] \rightarrow Arg1)$

Results.

Table 4.4: Extracting NGAC Policy using Semantic Role Labeling Approach

Access Control Policy Sentence	**NGAC Policy**
An HCP creates patients.	HCP–[ars:creates]–patients
Doctor is an HCP.	Doctor–is–HCP
A nurse is prohibited from creating patients.	$user\_deny$ (nurse, create, patients)
Whenever a HCP changes a patient record, an email must be sent to the administrator.	Event (HCP–changes–patient record); Response(email–send)

Table 4.4 presents the output generated by *AllenNLP*, utilizing the iTrust dataset [44].

ACP 1 sentence: "*An HCP creates patients.* "Subject (ARG0): "HCP", Verb (V): "creates", Object (ARG1): "patients". The subject ("HCP") corresponds to the user attribute. The verb ("creates") implies an action of creating. The object ("patients") represents the object attribute. The extracted components align with the NGAC association pattern: (user attribute–access right (ars) –object attribute).

$$(Bob \rightarrow HCP \rightarrow [ars : create] \rightarrow patient \rightarrow John)$$

ACP 2 sentence: "*Doctor is an HCP.* "Subject (ARG0): "Doctor", Verb (V): "is", Object (ARG1): "HCP". The subject ("Doctor") corresponds to the user attribute. The verb ("is") implies an assignment relation. The subject ("HCP") represents the user attribute. The extracted

components align with the NGAC assignment pattern: (user attribute–user attribute).

$$(Doctor \rightarrow [is] \rightarrow HCP)$$

ACP 3 sentence: "*A nusre is prohibited from creating patients.*"Subject (ARG0): "nurse", Verb (V): "prohibited from creating", Object (ARG1): "patient". The subject ("nurse") corresponds to the user attribute. The verb ("prohibited from creating") implies a denial of action. The object ("patients") represents the object attribute.The extracted components align with the NGAC prohibition pattern: deny (user attribute,(ars), object attribute).

$$deny(Alice \rightarrow nurse \rightarrow [create] \rightarrow patient \rightarrow John)$$

ACP 4 sentence: "*Whenever an HCP changes a patient record, an email must be sent to the administrator.* "*Event*: [(ARG0): "HCP," Verb (V): "changes," (ARG1): "patient record." The ("HCP") corresponds to the user attribute. The verb ("change") implies an action of modifying. The ("patient record") represents the object attribute]. *Response*: [(ARG0): "email," Verb (V): "be sent." The ("email") corresponds to the object attribute. The verb ("be sent") implies an action of notifying]. The extracted components align with the NGAC obligation pattern: (event, response).

$$(Jack \rightarrow HCP \rightarrow change \rightarrow patient\ record \rightarrow J.M.Record); (email \rightarrow be\ sent)$$

4.5 Limitations and Lessons

Limitations. The primary limitation of *SpaCy* in the context of access control policy language is its inherent difficulty in customizing its general model to accommodate the specific requirements associated with access control policies. Access control policies often employ language constructs unique to the security domain. Since *SpaCy*'s model is not inherently tailored to understand these specialized linguistic patterns, it struggles to accurately process and extract key elements from

access control policy sentences. Particularly, *SpaCy* faces challenges in precisely defining prohibition within the context of access control policies, such as terms like "prohibited" or "denied.", Thus, we made a list of prohibition verbs. *SpaCy* has difficulty identifying and interpreting words that indicate a prohibition of access. This limitation arises from the complex linguistic structures within access control policies. Prohibition-related terms may not be explicitly labeled or recognized by the *SpaCy*. Consequently, *SpaCy* fails to effectively capture the nuanced language used in access control policies to express prohibitions, highlighting the need for additional customization to comprehend and accurately interpret access control policies containing prohibition-related language.

Moreover, relying solely on the *SRL* to extract NGAC patterns proves ineffective. While *SRL* excels at identifying semantic roles within sentences, it fails to capture the intricate patterns of NGAC assignment relations out of the list of assignment verbs. These assignment relations are pivotal in the NGAC model as they define the relationships between users and their attributes and between objects and their associated attributes. *SRL* aids in identifying different parts of a sentence and their functions, such as determining who is doing what to whom. However, regarding access control policy language, which describes how users can access objects and delineate their relationships, *SRL* lacks the detail and specificity to understand these complexities.

Lesson. Our experience with SpaCy and SRL has highlighted significant limitations in capturing essential NGAC components, like assignment and prohibition, within the NGAC model. This underscores a major challenge in applying current NLP techniques, primarily designed for general language and lacking specificity for policy language and vocabulary.

Recognizing this distinct gap in the existing NLP methodologies, there is an imperative need to develop a customized language architecture. Such an architecture would be specifically tailored to enhance the comprehension and interpretation of security and privacy policy language, catering to the nuances and intricacies of policy-related vocabulary and structure.

Given these challenges, this dissertation presents a framework that harnesses customized models explicitly crafted to interpret and understand security and privacy policies. This framework

aims to bridge the gap between general NLP techniques and the specialized requirements of policy language, facilitating more accurate and effective policy analysis.

Chapter 5

Automated Access Control Policy Specification and Analysis

In this chapter, we present our approach to extracting and analyzing security policies, aimed at automating the translation of ACPs from natural language expressions to NGAC formal model. The chapter proposes methods for deriving the NGAC model directly from natural language security requirements. The process involves identifying ACP statements, discerning relation types, capturing entities, generating graphical representations of NGAC ACPs, and analyzing the derived policies. The subsequent sections provide a detailed elaboration of our proposed approach.

5.1 Proposed Approach

We developed an approach named Secuirty Requirements to Access Control Model *(SR2ACM)* that automatically extracts NGAC ACPs directly from security requirement documents written in natural language and converts them into a formal NGAC model. *SR2ACM* identifies ACP sentences, then extracts the NGAC elements, including entities and relations between those entities. It then forms these elements and relations as a formal NGAC model. SR2ACM comprises five tasks: ACP identification, NGAC relations identification, NGAC attributes identification, NGAC graph generation, and NGAC graph testing.

SR2ACM is shown in Figure 5.1. The five tasks in the proposed *SR2ACM* as follows.

- *Task 1 ACPs Identification:* We identify which sentences in security requirement documents are ACPs using BERT as a binary classifier.

- *Task 2 NGAC Relation Identification* We identify the NGAC relation types in each ACP sentence, including association, assignment, prohibition, and obligation, using BERT as a multi-label classifier.

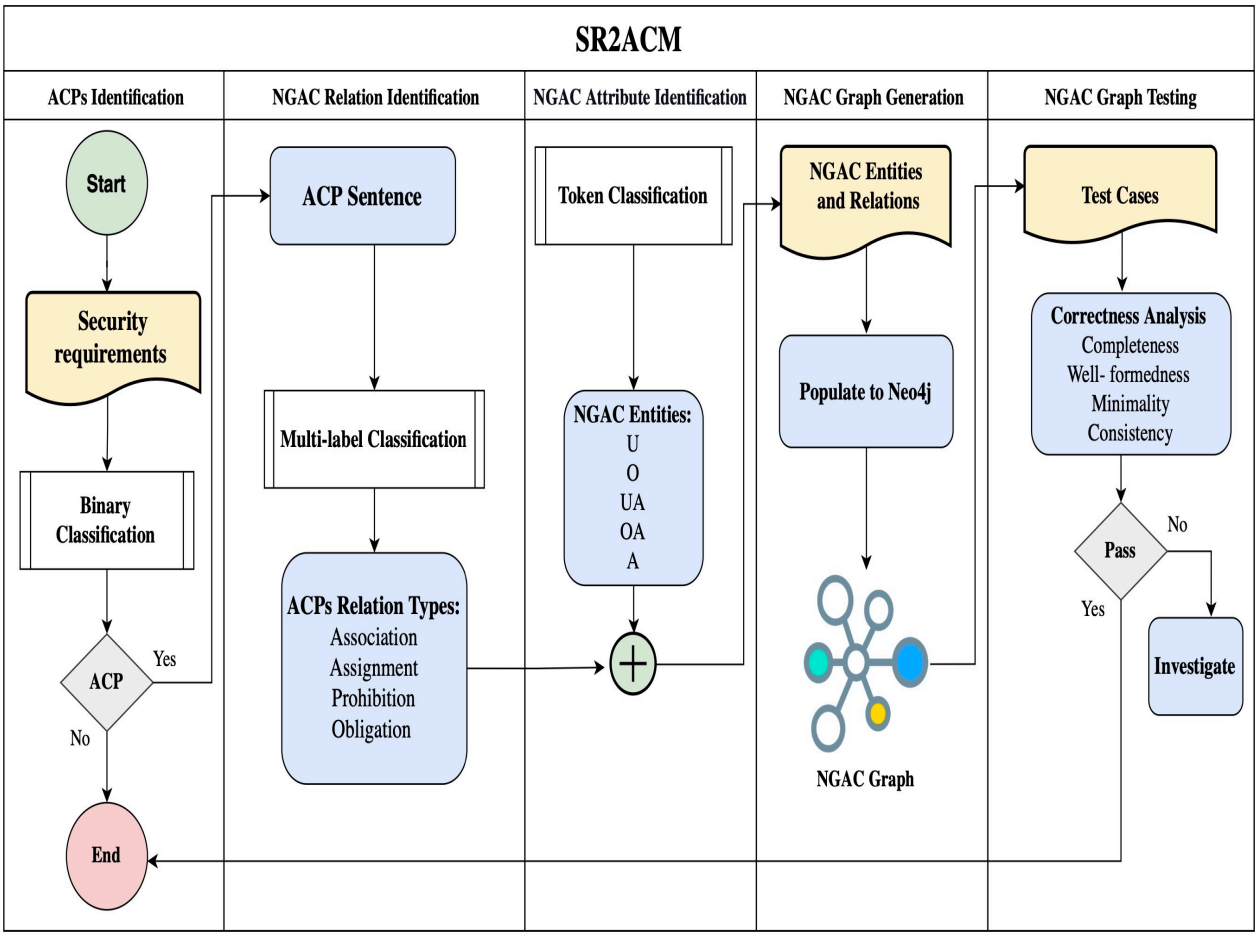

Figure 5.1: A Cross Functional Flowchart Presents SR2ACM

- *Task 3 NGAC Attributes Identification:* We use the extracted ACPs to identify the NGAC entities includes User (U), User Attribute (UA), actions (A), Object (O), and object attribute (OA). We propose a model called NGAC Entity Recognition (NGAC-ER). We use BERT as a token classifier to label words in each ACP as one of the NGAC entities, including {U, UA, A, O, OA}.

- *Task 4 NGAC Graph Construction:* We represent the extracted entities and relations as NGAC graph using *Neo4j*.

- *Task 5 NGAC Graph Testing:* We formally analyze the correctness of the extracted NGAC model using graph-based testing. We verify the satisfaction of *completeness*, *well-formedness*, *minimality*, and *consistency* properties.

5.2 Datasets

Constructing the proposed approach (*SR2ACM)* requires access to ground-truth datasets, enabling the development of the models tailored to accomplish specific tasks. Therefore, we aim to establish a valuable ground-truth dataset. This work created the manual annotation dataset in ACPs. We create a dataset for the ACPs in the educational domain. We started collecting data and then annotated those data manually. However, we do not use only the original data we collected; we also use the augmented version. The following sections explain data collection, annotation, and augmentation.

5.2.1 Data Collection

Prior research [47, 60, 71] on the identification of ACP sentences provide several noteworthy datasets to be leveraged for supervised learning in our work. These datasets are iTrust-v1 (DS1) [44], iTrust-v2 (DS2) [44], IBM (DS3) [60], CyberChair (DS4) [65], and Collected-ACPs (DS5) [60]. These datasets include ACPs and non-ACPs sentences.

In addition to these datasets, we have created a corpus containing about a thousand ACP sentences about access control in information technology from US universities' websites. We named our dataset DS6. To create this dataset, we created a web crawler to collect all text in Information Technology (IT) access control policies on the university's websites. Our crawler collects the text in a paragraph. We segment the paragraph into sentences and associate it with an ID that provides a reference of the paragraph that contains the sentence to preserve the context. Table 5.1 shows the data collected from the universities used in DS6. This chapter uses six corpora under the monikers DS1, DS2, DS3, DS4, DS5, and DS6.

DS1: It is collected from iTrust [44], an open-source healthcare application that includes patients' medical history, primary caregivers identification, recording communications with doctors, and sharing the results. DS1 is used in [71].

DS2: It is the largest version of DS1, which contains more than a thousand sentences. DS2 is used in [47].

DS3: IBM dataset collected by IBM course registration system. This is a dataset in the education domain that uses [60] in their research.

DS4: Cyberchair dataset is in the conference management domain. This dataset includes different policies from conferences and workshops [65].

DS5: This dataset is a combination of 114 ACP sentences collected from 18 sources, including published papers and websites [60].

DS6: We collected US universities' information technology access control policies.

Table 5.1: Number of the Collected Security Sentences

University Name	Number of ACPs
Colorado State University	66
University of Colorado Denver	165
University of Colorado Boulder	35
University of Northern Colorado	350
Harvard University	165
Georgia State University	25
University of Arizona	36
New York University	139
Rochester Institute of Technology	27
University of California	26
University of Massachusetts	23
Tennessee State University	34

5.2.2 Data Annotation

We did three types of annotations: (i) uses a binary label to indicate if a sentence is *ACPs* or *non-ACPs*, (ii) annotate a sentence as one or more of NGAC relation types *association*, *assignment*, *prohibition*, or *obligation*, and (iii) annotate each word in each ACP as *user*, *action*, *object*, *user attributes*, or *object attributes*. We did these three types of annotations manually on DS6. For DS1,

DS2, DS3, DS4, and DS5, we ran the last two types of annotations, as the first annotation type has been conducted and published by [47, 65, 71].

The annotation task is formulated by posing three questions, shown in Table 5.2, along with their labels. The first question requires binary yes/no labels while the second question requires one or more of four labels: *association (A)*, *assignment (AA)*, *prohibition (P)*, *obligation (O)*, and the third question requires one of five labels: *user (U)*, *action (A)*, *object (O)*, *user attribute (UA)*, *object attribute (OA)*.

For each annotation task, the three annotators independently labeled the data. Then, the supervising expert reviewed their annotations, who validated them, and resolved any discrepancies between the annotators. This rigorous process ensured the consistency and accuracy of the annotations across our dataset. Regarding online dataset we found, while the first type of annotation (identifying ACPs) had been previously conducted by [47, 65, 71], we applied the latter two annotation types to these datasets, enriching them with detailed NGAC relation and entity labels. Each annotator had to assign the labels as shown in Table 5.2.

Table 5.2: The three questions answered by annotators, and the corresponding labels: Yes/No, or Association [A], Assignment [AA], Prohibition [P], Obligation [O], or User [U], Object [O], Action [A], User attribute[UA], Object attribute [OA]

Question	Total	Label
Does the sentence contain an access control policy specification?	3482	Yes/No
What are the relation types between subject and object in the ACP sentence?	2234	[A],[AA][P],[O]
Where is the User (U), Object(O), Action (A) and User Attributes (UA), Object Attributes (OA) in the ACP sentence?	38223	[U],[UA],[A],[O],[OA]

5.2.3 Data Augmentation

Data augmentation effectively reduces a model overfitting by creating a different modified data version [69]. In this work, we use various augmentation techniques, including back translation, combining other datasets and the GPT3 Model Generator [72].

Back Translation: In ACP identification, we used the back translation augmentation technique. This technique provides us with more ACP sentences that are similar semantically to the original ACPs but are syntactically different. We use Google translator API [1] to develop our back translation script. The steps of back translation are described below.

1. *Temporary translation:* translates each of the original training labeled data into a different language. We translate ACP sentences from English to a random language, then translate them to another random language for 10 times for each sentence.

2. *Back translation:* Each translated sentence is translated back into the original language, which is English. This step is repeated 100 times for each sentence in each dataset.

3. *Duplicate removal:* at the end of the process, we have a corresponding back-translation for each original text data. The goal here is to keep only one occurrence of each duplicate.

Table 5.3 shows each dataset's ACPs and non-ACPs after back-translation augmentation is completed. These datasets are used in the ACPs identification task in Section 5.3.

Table 5.3: A Statistic of Data Sets Before and After Augmentation

DS_n	**ACPs**	**Aug. ACPs**	**non-ACPs**	**Aug. non-ACPs**
DS1	419	4034	52	4538
DS2	528	5526	643	10116
DS3	163	1700	239	2977
DS4	124	2805	179	1454
DS5	115	1126	27	1385
DS6	828	4770	109	8440

Combined DataSets: In the identification of NGAC relations, we need a dataset containing all four NGAC relations types (association, assignment, prohibition, and obligation) to build a model.

[1]Googletrans: https:/pypi.org/project/googletrans

However, not all datasets contain all the four NGAC relations as shown in Table 5.4. Thus, we combined all original datasets to have all NGAC relations types in the training dataset used in the NGAC relation identification task in Section 5.4.

Table 5.4: Number of NGAC Relations in each Dataset

DS_n	**Association**	**Assignment**	**Prohibition**	**Obligation**
DS1	408	269	6	0
DS2	484	317	4	12
DS3	124	105	1	1
DS4	177	51	0	28
DS5	89	76	8	10
DS6	474	68	28	47
Combined DS_n	1756	886	47	98

We observed an imbalance between sample numbers of (association, assignment) compared with (prohibition, obligation) as shown in Table 5.4. This significant discrepancy arises due to the nature of security requirements that contain association and assignment relations more than prohibition and obligations. The reason is that prohibited actions are often implicit in the sentences, while obligations are mentioned in security requirements documents when certain exception happens.

To overcome the lack of samples related to prohibitions and obligations, we augmented the dataset by creating additional instances for both categories, similar to the original samples. Each type was duplicated twice, ensuring sufficient representation of all four NGAC relation types while preserving their distinctive proportions. Our approach involved leveraging the GPT-3 Model Generator, elaborated in the subsequent paragraphs.

GPT-3 Model Generator: In the NGAC relations and attributes identification, we use the GPT-3 model to generate new ACP samples. We follow an approach similar to that in [72]. This approach

is based on the GPT-3 model to produce synthetic sentences with hyper-realistic text samples from a mixture of actual samples.

The procedure for generating augmented ACPs is as follows: extracting a few ACP sample sentences from the training data, embedding these samples in the prompt, and generating an augmented sentence influenced by the sample sentences. As a result, we generated a thousand ACP sentences that varied between simple and complex sentences. Table 5.5 shows examples of synthetic ACPs generated by the GPT-3 Model Generator. Those data are used in the NGAC relations identification and NGAC entity recognition task in Section 5.4 and Section 5.5.

Table 5.5: Five Synthetic ACPs Sentences Generated by GPT-3

(1)	Only administrators, not data owners, make changes to the security label of a resource.
(2)	LHCP cannot cancel old appointments.
(3)	Inactive patients cannot be changed or logged into the system and can only be reactivated by the administrator.
(4)	The professor can change a student's grade at any time.
(5)	If a teacher is signed up to teach the course in the current stream, the system will not delete the course.

5.3 Task 1: Access Control Policies (ACPs) Identification

This task aims to extract the ACPs from NL documents. To do this task, we fine-tune the BERT and BERT family models to classify a sentence as ACPs or non-ACPs as a sequence classification problem.

Sequence Classification Problem

In NLP, text classification assigns text to different categories. A large pre-training language model is sufficient for learning natural language representations by leveraging copious unlabeled data. The most distinguished examples of those large pre-training language models are Bidirectional Encoder Representations from Transformers (BERT) [17], Embeddings from Language

Model(ELMo) [55], Generative Pre-trained Transformer(GPT) [41], and Universal Language Model Fine-tuning (ULMFiT) [33]. These large language models are trained on text data using unsupervised approaches.

We use BERT model because it performs well in text classification tasks due to its language comprehension capabilities. BERT is based on a multi-layer bidirectional transformer and is trained on a large amount of plain text for two main tasks: masked word prediction task and next-sentence prediction task. We must fine-tune these models using task-specific training data to apply a pre-trained model to a specific task. Thus, we need to design additional task-specific layers after the pre-training model.

For the text classification task, BERT adds a simple layer after the pre-trained model to fine-tune the model on a particular dataset and create state-of-the-art models for text classification tasks.

This task aims to classify the given sentence as ACPs or non-ACPs as a binary classification problem. Thus, we leverage BERT for the ACPs identification task, as shown in the following section.

Fine-Tune BERT for ACPs Classification

ACPs identification involves the following steps. First is segmentation, which divides a string of written language into its component sentences. So we have $S = \{s_1, ..., s_i\}$ where s is a sequence and i is a sequence number. Second, we must transform each sentence into a sequence of S[SEP] tokens. Third, reformat that sequence of tokens by adding [CLS] and [PAD] tokens before using them as input to our BERT model [37]. So, inputs are designed as { [CLS],$s_i[SEP]$,[PAD] } then fed into the BERT encoder. Fourth, the BERT encoder will output an embedding vector of size 128 for each token. Finally, we use these vectors as input for binary text classification. The embedding vector from [CLS] token (which stands for classification) is used as an input for the binary classifier, which then will output a vector of size the number of classes in our classification task {ACP, non-ACP}.

Challenges

The main challenge in the ACPs identification task is the unbalanced data. We have fewer ACP sentences than non-ACP sentences. This unbalance makes the model training overfitting on a specific class, which leads to a biased model. To avoid bias, we use an augmentation technique to balance the data. In this task, we used the back translation technique to achieve our goal as described in section 5.2.

Experimental Results

Table 5.6 shows the average *precision*, *recall*, and *F1-score* across the resultant of BERT text classification that represents ACPs sentences in test datasets. In the majority of cases, the performance is promising. However, we found that the model's performance after data augmentation is better, proving the importance of data size and quality in large NLP language models such as BERT. Furthermore, this study examines other BERT-based models like distilling BERT (DistillBERT), Robustly Optimized (RoBERTa), and an eXtreme Language understanding NETwork (XLNET). Our goal is to compare the performance of different BERT models on the ACPs identification task as shown in Table 5.6. In most cases, we found that the XLNET model performs better than other models.

A Discussion of the Experimental Results

The experimental results show the performance of various models on the test set of datasets. Overall, the XLNET shows significantly better performance in terms of *precision*, *recall*, and *F1-score* because XLNET operates in an autoregressive manner. In other words, it generates tokens sequentially while considering the preceding tokens' context. In contrast, BERT is a masked language model (MLM) in which random tokens are masked, and the model is assigned the task of predicting those masked tokens [37, 40]. Furthermore, our proposed ACP identification model overall outperforms the prior work Tex2Policy [71], ACRE [60], SENNA [47], and BERT based SRL [70] for ACP identification in terms of F1-score as shown in Table 5.7.

Table 5.6: Comparative Analysis of the ACPs Identification Performance.This table shows Precision (P), Recall (R), and F1-score (F_1) to compare between four models BERT, DistilBERT, RoBERTa and XLNET

	BERT			**DistilBERT**			**RoBERTa**			**XLNET**		
Dataset	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
DS1	92%	91%	88%	78%	88%	83%	78%	88%	83%	93%	92%	91%
DS1-Aug	98%	98%	97%	98%	98%	97%	98%	98%	98%	98%	98%	98%
DS2	63%	79%	70%	63%	79%	70%	63%	79%	70%	63%	79%	70%
DS2-Aug	93%	80%	79%	78%	80%	74%	63%	79%	70%	87%	87%	87%
DS3	93%	91%	91%	88%	86%	85%	78%	66%	58%	93%	92%	92%
DS3-Aug	98%	98%	98%	97%	97%	97%	99%	99%	99%	99%	99%	99%
DS4	75%	75%	75%	83%	79%	77%	31%	56%	40%	72%	72%	72%
DS4-Aug	86%	85%	85%	87%	86%	86%	87%	86%	86%	88%	88%	88%
DS5	92%	92%	92%	90%	90%	90%	90%	90%	90%	95%	95%	95%
DS5-Aug	96%	96%	96%	95%	95%	75%	96%	95%	95%	95%	95%	95%
DS6	77%	88%	82%	77%	88%	82%	77%	88%	82%	89%	89%	89%
DS6-Aug	89%	90%	90%	89%	90%	89%	89%	90%	89%	88%	90%	89%

The results also indicate that the BERT-based methods, including our approach, performed better than Text2Policy [71], ACRE [60], SENNA [47]. This showcases the proficiency of BERT transformers in accurately identifying ACP sentences, especially compared to alternative machine learning algorithms. In particular, the self-attention mechanism within transformers enables BERT models to focus on relevant parts of the input text, effectively capturing long-range dependencies and contextual information essential for understanding ACPs sentences [37].

BERT-based Semantic Role Labeling (SRL) approach proposed by [70] demonstrates promising performance in identifying ACPs. However, our method achieves an average 93% F1 score, while [70] achieves 85%.

5.4 Task 2: NGAC Relation Identification

This task aims to extract the relation type between user and object attributes in access control policy sentences. To do this task, we propose a model able to classify ACP sentences as one or

Table 5.7: The ACP Identification Performance Compared with the Related Work

	Text2Policy [71]	**ACRE** [60]	**SENNA** [47]	**BERT based SRL** [70]	**Our Approach (BERT)**
Dataset	$F_1(\%)$	$F_1(\%)$	$F_1(\%)$	$F_1(\%)$	$F_1(\%)$
DS1	87%	98%	80%	95%	98%
DS2	–	88%	58%	82%	87%
DS3	97%	87%	59%	89%	99%
DS4	–	64%	82%	77 %	88%
DS5	–	89%	70%	78%	95%

more of the NGAC relation types, including association, assignment, prohibition, and obligation. We fine-tune the BERT based on a multi-label classification problem.

Multi-Label Classification Problem

In a multi-label classification problem, the training set is composed of instances, and each one can be assigned multiple categories represented as a set of target labels. Thus, the task is to predict the label set of test data. We aim to construct a model that can predict multiple types of NGAC relations in an ACP sentence. We propose an approach based on the BERT transformer to solve the multi-label classification problem [12].

FineTune BERT for ACPs Multilabel Classification

In the NGAC model, ACPs can fall into multiple categories of relations. For example, the admin can add a student's record. This ACP sentence contains two categories: an association and an assignment simultaneously. The assignment is the relation between the user and user attributes $\{User \leftarrow (role : Admin)\}$, and the association whereas admin associated with student's record by read and write operation:{admin$\rightarrow$ (read, write) $\rightarrow$ record}. In this task, we build a model to predict a multi-type of NGAC relations to obtain the relation between user attributes and object attributes in ACPs. NGAC model includes association, assignment, prohibition, and obligation.

After we extract ACP sentences in Task 5.3, we aim to identify the NGAC relation types to define the relation between user attributes and object attributes in each ACP sentence. To accom-

plish this task, we need to classify the relation in each ACP as a multi-label classification problem. BERT can be easily applied to downstream NLP tasks in a fine-tuned manner after obtaining the sentence vectors. We fine-tuned the BERT model to be able to classify ACPs for one or more types of NGAC relations *(association, assignment, prohibition, and obligation)*.

In this task, we utilize the BERT base model and other BERT family models (RoBERTa, DistilBERT, and XLNET) with the following hyperparameters: text sequence length = 128, train epochs = 10, train batch size = 64, evaluation batch size = 66. Given that the output is multi-label *(association, assignment, prohibition, and obligation)*, it is appropriate to utilize both a Sigmoid activation function and a Binary Cross-Entropy loss (BCELoss) function for the final output. However, PyTorch documentation recommends employing the BCEWithLogitsLoss() function, which integrates a Sigmoid layer and the BCELoss within a single class, for more numerical stability than using a plain Sigmoid followed by a BCELoss. [39]. The classifier outputs a vector that has a probability of each label. However, we need to roll it over to a 1 or 0 depending on whether it is above or below a threshold value. Thus, we set threshold = 0.5. This binary classification approach tests each class to determine whether it belongs to the association or not, prohibition or not, obligation or not, or assignment or not.

Challenges

To predict all NGAC relation types, we need to train the model on a dataset that contains all NGAC relation types. Since not all datasets in section 5.2 contain all types of relations, we combined all datasets. As a result, we have a dataset that contains all NGAC relation types, as shown in Table 5.4. Even though there are unbalanced classes between the four types, this is the nature of access control policies. In general, we find ACPs associate users with resources more than prohibit users from resources. To illustrate, faculty and teacher assistants can modify the grades. That means users other than faculty and teacher assistants can not modify the grade. However, after combining the dataset, we lack prohibition and obligation samples. To address this, we augmented the prohibition and obligation ACPs using GPT-3 Model Generator as discussed in section 5.2.

Experimental Results

Our experimental results are shown in Table 5.8. We obtained F1- score of 96% for association relation types identification. For assignment relation type,we obtained 87% F1-score. These results are promising for the most dominant relation types: association and assignment. However, we obtained 43% and 65% F1-score for prohibition and obligation, which are less visible in security requirements documents. Overall, we obtained 91% accuracy for the NGAC relation identification model.

Table 5.8: BERT's Performance for NGAC Relation Types Identification

Relation	P(%)	R(%)	F_1(%)
Association	93%	99%	96%
Assignment	88%	85%	87%
Prohibition	10%	27%	43%
Obligation	79%	55%	65%

To mitigate the low performance of prohibition and obligation samples, we employed the GPT-3 Model Generator to augment these NAC relations after the data augmentation. Table 5.9 presents the performance of the BERT models family for identifying NGAC relations. Remarkably, by the BERT model, we observed a significant improvement in the performance of prohibition and obligation by 50% and 22%, respectively.

A Discussion of the Experimental Results

To demonstrate the capabilities of our approach, we evaluate the experiments on *precision*, *recall*, and *F1-score* across the resultant of multi-label classification that represents NGAC relations types in our test data. Overall, our approach shows significantly promising performance. Our experimental results are shown in Table 5.8, which shows the performance with an overall average of 91% accuracy on the test set of ACPs sentences containing different types of NGAC relations by the BERT model. However, the low performance of the prohibition is due to a low training

Table 5.9: Comparative Analysis of the NGAC Relation Types Identification Performance.This table shows Precision (P), Recall (R) and F1 score (F_1) to compare between four models BERT, DistilBERT, RoBERTa and XLNET

	BERT			**DistilBERT**			**RoBERTa**			**XLNET**		
NGAC Relation	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
Association	97%	98%	97%	95%	95%	95%	94%	96%	95%	95%	95%	95%
Assignment	81%	92%	86%	74%	92%	82%	80%	87%	83%	74%	92%	82%
Prohibition	97%	88%	93%	82%	79%	80%	92%	79%	85%	97%	76%	85%
Obligation	89%	87%	88%	84%	79%	81%	88%	77%	82%	83%	91%	87%

sample and the nature of the access control policy specification. To illustrate that, if an access control policy specifies what is allowed for a specific person, it implicitly disallows others. Thus, security requirements documents contain association relations more than prohibition. However, we overcome the low performance by augmenting the prohibition and obligation types in the training data. To the best of our knowledge, no previous work investigated how to automatically identify the relation types between user attributes and object attributes in access control policies.

5.5 Task 3: NGAC Attributes Identification

This task aims to extract NGAC attributes in each ACP sentence. We proposed a model called Attribute Entity Recognition (NGAC-ER) to do this task. NGAC-ER is based on the BERT models to solve sequence labeling problems. We fine-tuned BERT base model and other BERT family models (RoBERTa, DistilBERT, and XLNET), to classify each token in a sequence as NGAC attributes user, user attributes, operation or action, object, and objects attributes.

Sequence Labelling Problem

A sequence labeling task is defined where a sequence of tokens is given, and a model should assign labels or classes to this token. The objective of a model is to learn the corresponding word labels in a set of labels or tags to identify and classify patterns, which can then be used to infer

labels for new sequences of tokens [18]. Well-known examples of sequence labeling tasks are NER and PoS tagging [25, 68].

NGAC Entity Recognition (NGAC-ER)

NGAC model standard based on relations between data elements to create, manage and enforce access control policies [14]. ACP specification in the NGAC model has multiple entities, including user, user attribute, object, object attribute, and action. We propose the NGAC Entity Recognition (NGAC-ER) model. This model aims to solve a sequence labeling classification problem in ACP specification. Since the BERT model has a promising performance on token classification, we fine-tuned the BERT model family to classify each token in a sequence as a user, user attribute, object, object attributes, and action. We aim to predict the following attributes *U*ser (U), *U*ser. Attributes (UA), *O*bject (O), *O*bject. Attribute (OA), and *A*ction (A) as illustrated in Algorithm 1.

The architecture of BERT for the token classification task is as follows: the input of BERT represents a text sentence. Next, BERT tokenization, which converts a sentence into tokens and then embeds every token of the text sentence. Thus, every sequence starts with a special token. ([CLS]),and ends with ([PAD]). In our model, we use BERT-based-cased; the number of layers L, the hidden size H, and the number of self- attention heads A as the default setting as follows: L= 12, H= 768, A=12, Total Parameters= 110M Sequences MAX LEN= 75 batch size= 32

Challenges

Neural architectures and NLP transformers are very sensitive to the size of training data and tend to over-fit on small datasets. The training data is the main challenge to building the NGAC Entity Recognition model (NGAC-ER). We need various sentences that have different attributes. Thus, we aim to increase the training data by augmenting our data. In this task, we generated new a thousand of different ACPs sentences in the English language using the GPT3 model as shown in section 5.2

Algorithm 1: NGAC Entity Recognition (NGAC-ER) Algorithm

Input : $S = \{s_1 \dots s_n\} \in ACPs$
Output: $NGACAttributes = \{U, UA, A, O, OA\}$
Tokenization
$\{W_i, , \dots W_i\} \leftarrow WordTokenization(s_i)$
{Sent_ID, $Word_i$,Tag_i} $\leftarrow Get.Sentence$ (s_i)
$Sent(words, tags) \leftarrow GroupBYSentID(words, Tags)$
$TokenizeSent \leftarrow Sent.Tokenization(Sent)$
input_id $\leftarrow TokenToId(Tokenized.Sent)$
Fine-Tune BERT Models
(Tr_Inputs, Val_Inputs, Tr_Tags, Val_Tags) $\leftarrow TrainTest.Split$ (input_id, tags)
NGAC-ER.Model $\leftarrow TokenClassification$ (Tr_inputs, Val_inputs, Tr_tags, Val_tags)
NGAC Attributes $\leftarrow NGAC - ER.Model$ (Test_data)

Experimental Results

To demonstrate the capabilities of our approach, we evaluate the experiments on precision, recall, and F1-score across the resultant of token classification that represents NGAC attributes. The proposed NGAC-ER shows promising results to predict each token in an ACP sentence shown in Table 5.9

Table 5.10: Comparative Analysis of the NGAC Attributes Identification Performance.This table shows Precision (P), Recall (R), and F1-score (F_1) to compare between four models BERT, DistilBERT, RoBERTa and XLNET

	BERT			**DistilBERT**			**RoBERTa**			**XLNET**		
NGAC Attributes	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
User (U)	100%	53%	70%	71%	64%	68%	74%	67%	70%	85%	73%	79%
User Attribute (UA)	93%	99%	96%	92%	90%	91%	90%	90%	90%	91%	92%	91%
Object (O)	67%	57%	62%	61%	55%	58%	58%	60%	59%	54%	55%	54%
Object Attibute (OA)	91%	87%	89%	58%	46%	51%	44%	41%	43%	79%	66%	72%
Action (A)	83%	80%	81%	81%	82%	81%	76%	80%	78%	56%	59%	57%
Other	96%	85%	89%	92%	95%	93%	92%	92%	92%	76%	76%	76%

A Discussion of the Experimental Results

Our experimental results are shown in Table 5.10, which shows the performance with an overall average of the accuracy on the test set of ACPs sentences containing different types of NGAC

attributions. However, predicting Object O and User U is not high enough because of the nature of security requirements documents. It is usually written in general without specifying a user such as John or an object such as CS Course.

NGAC-ER model significantly outperforms the module proposed in [3] by obtaining a higher accuracy in terms of *precision*, *recall*, and *F1-score* for user attributes and object attributes identification as shown in Table 5.11. Moreover, the NGAC-ER model outperforms the module in [31] that aims to identify actor, action, and resource in user stories by obtaining 87% accuracy while the NGAC-ER model obtains 93% accuracy. BERT transformer employs attention mechanisms, enabling it to focus on different segments of the input sequence during prediction. This attention mechanism allows the model to weigh the importance of different words in the input sequence, providing an understanding of the relationships between words [37]. This capability is the key factor contributing to the superior performance of the BERT transformer models over deep learning approaches in the task of extracting attributes in ACPs.

Table 5.11: Attribute Identification Performance Compared with Other Work

	Alohaly [3]			**Our work (NGAC-ER)**		
Access Control Attribute	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)
User Attribute	91%	80%	85%	93%	99%	96%
Object Attribute	75%	69%	71%	91%	87%	89%

5.6 Task 4: NGAC Model Construction

In this task, we use Neo4j to generate the NGAC graph from the entities and relations extracted using the models outlined in Sections 5.3, 5.4, and 5.5. Neo4j is a high-performance graph store that is scalable and schema-free (i.e., NoSQL) [66]. Neo4j is a fully transactional database where we can store structures in the form of graphs instead of tables [20]. It provides features such as managing the database, Cypher query language, Graph Processing Engine (GPE), and Labelled Property Graph (LPG) [66]. The LPG organizes the data as nodes, relationships, and properties.

A node is an entity with an identifiable ID and a set of properties. A relationship is a directed edge between two nodes. It also has an identifiable ID and a set of properties. A property is a key-value pair that characterizes a node or an edge. Both node and edge can have zero or more properties [20, 66].

Utilizing Neo4j, we populate the extracted NGAC entities and relations and represent them as an NGAC graph, as shown in Figure 5.2. We present an example of policies from the dataset DS6. This includes policies that define the access control privileges for sets of universities based on user attributes and object attributes.

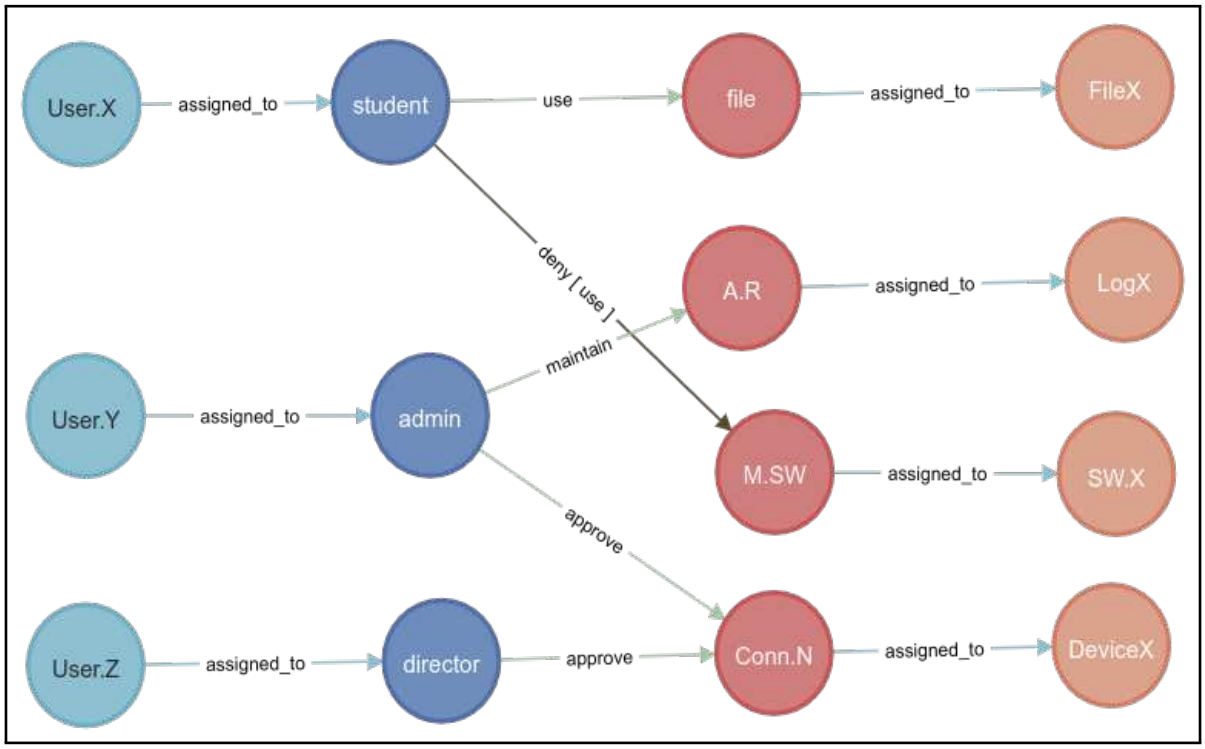

Figure 5.2: The Generated NGAC Graph Model

As shown in Table 5.12, we have a director of network services and an access control administrator who approves the devices that connect to the university network. At the same, the access control administrator maintains the access requests for the logs. Additionally, students are denied from using monitoring software. While students can only use files assigned to them. To demonstrate users accessing objects, we present the policy sentences (1-4) from the dataset DS6, shown in

Table 5.12. Sentences 5-7 assign three users (User.X, User.Y, User.Z) as a student, access control administrator, and director of network services. We added three sentences (8-11) to demonstrate objects (File.X, Log.X, SW.X, and Device.X) that are assigned to object attributes (file, access request (AR), monitoring software (M.SW), and connected network (Conn.N)).

Table 5.12: A Set of ACPs and Users and Objects

ACPs	
1	Student should use only their assigned file to access the information system.
2	Student is prohibited to use monitoring software in the university network.
3	The director of network services and access control administrator must approve the connected network devices.
4	The access control administrator will maintain for a minimum of six years logs of all access request approvals, user account creations, modifications, and deletions.
User	
5	UserX is a student.
6	UserY is an access control administrator.
7	UserZ is a director of network services.
Object	
8	File.X is an object.
9	Log.X is an object.
10	SW.X is an object.
11	Device.X is an object.

5.7 Task 5: Access Control Model Analysis

We assess the correctness of the extracted model through graph-based testing because the NGAC model is structured as a graph model. The testing method constructs a graph model representing the NGAC model and then generates paths covering each user assigned to user attribute as-

sociated with (or prohibited from) access rights to object attributes assigned to objects. These paths are then scripted as test cases to address the correctness aspects (*completeness*, *well-formedness*, *minimality*, and *consistency*). We utilize the educational policy in DS6 and healthcare policies in (DS1) to generate NGAC ACPs paths and then evaluate the correctness and compare between the two domains.

The testing procedure illustrated in Figure 5.3 is initiated by loading the paths derived from the NGAC graph for the ACP. Subsequently, a test scripter transforms these paths into Python code test cases. In each path, the test scripter retrieves the user from the initial node, the access right from an intermediary node, and the object from the concluding node. These elements are then translated into Regular Expressions.

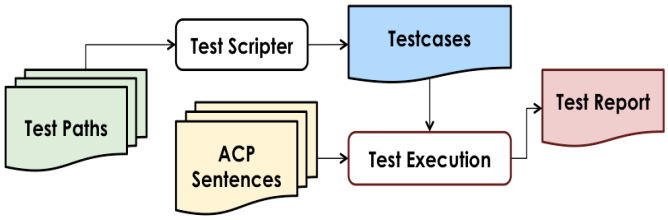

Figure 5.3: Testing Process

We use *Neo4j* Cypher path matching, which is called *MATCH* statement. Therefore, we can generate paths from every U user node to every O object node. The *MATCH* statement is written as follows:

```
MATCH paths=(u:U)-[:Assignment| Prohibition|

Association * 1..*]-(o:O)

RETURN distinct(paths) ORDER BY length(paths);
```

This *MATCH* statement generates a set of paths that covers the NGAC graph. These paths are a sequence of nodes and relations representing how a user node U reaches an object node O through $assignment$, $association$, and $prohibtion$ relations. We use these paths to script test cases that will be executed against each ACP sentence's content. For instance, when the test

cases are executed against the ACP 1 (Student should use only their assigned file to access the information system), at least one test case has to be passed successfully. This test case matches the word (student) as a user attribute node, (use) (intermediate node), and (file) as an object attribute node. If all test cases fail to match these three phrases, this ACP sentence has not been extracted completely.

The scripter will fetch the (student), (use), and (file) words and then form a regular expression as:

$$.*student.*(use).*file*.$$

The dot refers to any character; the star means zero or many. As a result, this regular expression verifies that from the beginning of the sentence, ignore a series of characters until reaching the (student) word, ignore another series of characters until reaching (use), and another string of characters until getting the word (file). Similarly, the scripter converts all paths into regular expressions. Each regular expression is then passed as an argument to a match function (*search (pattern, ACP_Sentence)*) that searches for a match over a string variable that represents an ACP sentence. The case sensitivity of letters is ignored. The result of this function is asserted to be not equal to *None*. The match function and the assertion are wrapped up in a test case defined with a unique name. The test cases generated are then encased into a class (i.e., test-suite). The following is a *Python* snippet code that shows one of the test cases resulting from the scripter when the first path from Table 5.13 is passed into it.

```
class TestSuite(unittest.TestCase):
 ACP_Sentence = ''
 def testcase01(self):
   pattern ='.*student.*use.*file*'
   found = re.search(pattern, self.ACP_Sentence + '|$', flags=re.IGNORECASE)
    self.assertNotEqual(found, None)
```

The test execution first loads the ACPs sentences. It then passes each ACP sentence to the test-suite, executes it, and reports the test result into a file. The subsequent paragraphs illustrate

Table 5.13: Paths Generated from the NGAC Graph Model

Paths: Sequence of nodes and relations
[{"name":"User.X"},{"assign":"assigned_to"},{"name":"student"},{"name":"student"},{"ars":"[use]"}, {"name":"use"},{"name":"name":"File"},{"assign":"assigned_to"},{"name":"File.X"}]
[{"name":"User.X"},{"assign":"assigned_to"},{"name":"student"},{"name":"student"},{"deny":"[deny]"}, {"name":"use"},{"name":"name":"M.SW"},{"assign":"assigned_to"},{"name":"SW.X"}]
[{"name":"User.Y"},{"assign":"assigned_to"},{"name":"admin"},{"name":"admin"},{"ars":"[maintain]"}, {"name":":access request {"name":":access request"},{"assign":"assigned_to"},{"name":"Log.X"}]
[{"name":"User.Y"},{"assign":"assigned_to"},{"name": "admin"},{"name":"admin"},{"ars":"[approve]"}, {"name":":connected network device"},{"name":" :connected network device"},{"assign":"assigned_to"}, {"name":"Device.X"}]
[{"name":"User.Z"},{"assign":"assigned_to"},{"name": "director"},{"name":"director"},{"ars":"[approve]"}, {"name":":connected network device"},{"name":":connected network device"},{"assign":"assigned_to"}, {"name":"Device.X"}]

our utilization of graph testing to validate the correctness with respect to two datasets, one in the educational domain (DS6) and the other in healthcare (DS1). From ACP sentences in both datasets, we generate 48 paths using 30 ACPs from DS6 and 86 paths using 30 ACPs from DS1.

To ensure the robustness of our approach, we meticulously assessed its correctness through the lenses of *completeness*, *well-formedness*, *minimality*, and *consistency*. Furthermore, applying *SR2ACM* to two distinct domains—educational and healthcare policies to validate the approach's effectiveness empirically. This emphasizes the thoroughness of the correctness analysis and the empirical validation using real-world data.

Completeness ensures that the generated NGAC graph covers all the ACP sentences. We checked the completeness by testing the paths generated from the NGAC graph against the ACP sentences. Table 5.14 shows the test result when this test-suite is executed against the ACP sentence (Student is prohibited to use monitoring software in the university network). It reports that the first test case passed successfully while the others failed. The test result is analyzed based on the following test criterion:

$$\text{“}for\ each\ ACP\ sentence,\ at\ least\ one\ testcase\ should\ pass\text{”}$$

Thus, this ACP sentence has been extracted as NGAC policy.

Table 5.14: Result of Test Suite Executed against ACP1

```
********** ACP ************************
Testing:  Student is prohibited to use monitioring
software in the university network.
testcase01 (ACP_TestSuite.TestSuite) ...  ok
testcase02 (ACP_TestSuite.TestSuite) ...  FAIL
testcase03 (ACP_TestSuite.TestSuite) ...  FAIL
```

Well-formedness ensures the inclusion of all NGAC entities in each ACP sentence. The ACP initiation involves the user node and concludes with the object node, where the edge connecting the user attribute and object attribute signifies either an association or prohibition relation. To verify well-formedness, distinct paths for each ACP sentence were generated, ensuring that the path initiated from the user U and concluded with the object O and that the relation between UA and OA was either association or prohibition. The generation of distinct paths was facilitated using the *"DISTINCT"* keyword in *Neo4j* Cypher. Well-formedness is then verified by confirming the presence of U, UA, ars or $deny$, OA, and O in the path using two regular expressions as:

$$.*user.*user\ attribute.*ars.*object\ attribute.*object.*$$

$$.*user.*user\ attribute.*deny.*object\ attribute.*object.*$$

The first regular expression verifies whether a path is formed as a user assigned to a user attribute that is associated with access rights to an object attribute assigned to an object. The second regular expression verifies if a path is formed as a user assigned to a user attribute that is prohibited from having access rights to an object attribute assigned to an object.

Minimality or non-redundancy refers to the fact that an access right given to a user attribute to access an object attribute should not exist for two different paths in the NGAC graph. We checked the minimality by generating two groups of paths; one group has all paths of ACPs where the path starts from user attributes UA to object attributes OA without discriminating them. The other group involves discriminating paths. We used the keyword *DISTINCT* in the *Neo4j* Cypher to

generate the distinct paths. The intersection between these two groups of paths results in redundant paths, which reflect ACP sentences' redundancy.

The paths of all ACPs are retrieved through the following *MATCH* statement:

```
MATCH allPaths=(user:U)-[:Assignment|Prohibition|
Association * 1..n] -(object:O)
RETURN allPaths ORDER BY length(allPaths);
```

The distinct ACP paths are obtained by the following *MATCH* statement that is written as follows:

```
MATCH dsPaths=(user:U)-[:Assignment|Prohibition|
Association * 1..n] -(object:O)
RETURN distinct dsPaths ORDER BY length(dsPaths);
```

Consistency refers to the NGAC ACPs that do not conflict with other ACPs. To illustrate, for a particular object attribute, access rights are given to one user attribute UA_1 and prohibited to another attribute UA_2. Thus, a user or user attribute is assigned for UA_1 and UA_2, not granting and denying a user or user attribute access to the same object. We checked the consistency by detecting the *paths-conflict* that occurs between association relation and prohibition relation, giving an access right to a user attribute to access an object in one place and denying it from accessing the same object in another. After we generate paths for ACPs, we separate the paths into two groups; one group carries association relations paths, and the other group consists of prohibition relations. We then checked the consistency by exercising that for every user attribute, matching the access rights of paths with association relations to the denied access rights of paths with prohibition relations to which the same user attribute and object attribute are assigned. We used *regular expression* for matching values of the access rights and denied access rights. To ensure the verification process performs correctly, we added two ACP sentences to each dataset that causes *paths-conflict*, *(Student is prohibited to use monitoring software in the university network)* and *(Student can use monitoring software in the university network)*. The verification process detected this *paths-conflict* successfully. Removing this *paths-conflict* and rerunning the verification process over the gener-

ated NGAC models, we found no *paths-conflict* that may lead to inconsistency between the NGAC ACPs.

The results of testing the NGAC graph generated on the *educational* policies (DS6), along with the accuracy of the number of test cases passed over the number of test cases generated from the graph. The level of *completeness* averages at 95%, with the successful passing of 46 out of 48 test cases since some policies have not been extracted. The well-formedness level averages at 100% according to our analysis, which points to 48 well-formed paths. The analysis also indicates that 98% of NGAC ACPs are non-redundant. The remaining 2% of NGAC ACPs are redundant because many policies in educational institutions have applicability across diverse contexts. For example, when a university defines conditions for an event, the associated policy is expected to be enforced, which might also be relevant to other scenarios. Moreover, based on our analysis, there is no evidence of conflict between any two policies. We did not identify any path conflicts that could result in inconsistencies in the NGAC model.

From healthcare policies (DS1), the evaluation of the NGAC graph generated reveals noteworthy findings. The *completeness* level averages 97%, with 83 out of 86 test cases passing successfully, as some policies have not yet been extracted. Our analysis indicates a *well-formedness* level averaging at 100%. Additionally, our assessment highlights that 31% of NGAC ACPs are redundant. This redundancy occurs due to repeated role-specific policies, highlighting shortcomings in policy specification. Furthermore, our analysis did not uncover any evidence of *conflicts* between policies. No path conflicts were identified that could potentially lead to inconsistencies within the NGAC model.

As a result, *SR2ACM* demonstrates its proficiency in sufficiently extracting the formal NGAC model. This assertion highlights the superior performance of our proposed models, namely *ACPs identification, NGAC relations identification, and NGAC attribute identification models* compared to machine learning methods like [47, 60, 71], as well as deep learning approaches such as [3]. Furthermore, the proposed graph testing method demonstrates its applicability and simplicity when applied to real-world access control policies, compared to the approaches presented in [13,21,56].

5.8 Summary

This chapter discusses translating natural language security requirements, including ACPs, into formal access control models, focusing on NGAC. The motivation for the automated generation of access control models arises from the labor-intensive and error-prone nature of the manual interpretation of policy statements. The chapter proposes a method for deriving the NGAC model directly from natural language security requirements, employing advanced NLP techniques such as BERT, DistilBERT, RoBERTa, and XLNET. The process involves identifying ACP statements, discerning relation types, capturing entities, generating graphical representations of access control policies, and analyzing the integrity of the derived policies.

SR2ACM is developed to automatically extract ACPs as NGAC specifications from natural language security requirement documents and represent them graphically for analysis. Our approach classifies access control statements with 93% F1-score. It also identifies association, assignments, prohibition, and obligation with 96% , 89% , 93% , and 87% F1-score, respectively. The NGAC Entity Recognition (NGAC-ER) detects user attributes with 96% F1-score and object attributes with 89% F1-score. Moreover, through the analysis of the NGAC graph, we observed encouraging outcomes affirming the correctness of the extracted ACPs from the proposed framework in terms of *completeness*, *well-formedness*, *minimality*, and *consistency* for the real-world datasets.

Chapter 6

Automated Privacy Policy Specification and Analysis

This chapter introduces our proposed approach for identifying data practices within privacy policy documents to ensure alignment with the GDPR and the CCPA. Our approach encompasses identifying entity types, data types, and purpose types, followed by a comprehensive policy analysis. In the subsequent sections, we provide a detailed and thorough explanation of our proposed approach.

6.1 Proposed Approach

We developed an automated approach named *Privacy2Practice* to extract data practices outlined in privacy policies. This NLP-based pipeline is designed to illustrate how organizations handle user data, known as *data practice*. In a privacy policy, *data practices* are the procedures an organization employs when collecting, storing, sharing, and protecting personal information or data. These practices should be outlined in the privacy policy document [10].

This approach prioritizes the key data practices mandated by the GDPR and the CCPA [29]. These practices encompass entities, data types, and purpose types, symbolizing different aspects of data collection and sharing.

- **Entities**: The entities involved include the first and third parties. The first party refers to an entity that directly collects data from users. The third party denotes external entities or organizations that receive data from the first party for various purposes.
- **Data Type**: This refers to the specific categories or data types collected or processed.
- **Purpose Type**: The purpose of the collection and sharing process is the intended reason or justification for data collecting and data sharing.

Privacy2Practice streamlines the analysis of privacy policy documents to assess their alignment with data practices required by regulations like the GDPR and the CCPA. Extracting data

practices from natural language privacy policy documents is crucial to implementing our automated approach effectively. We introduce the following NLP pipeline to achieve this, as Figure 6.1 illustrates.

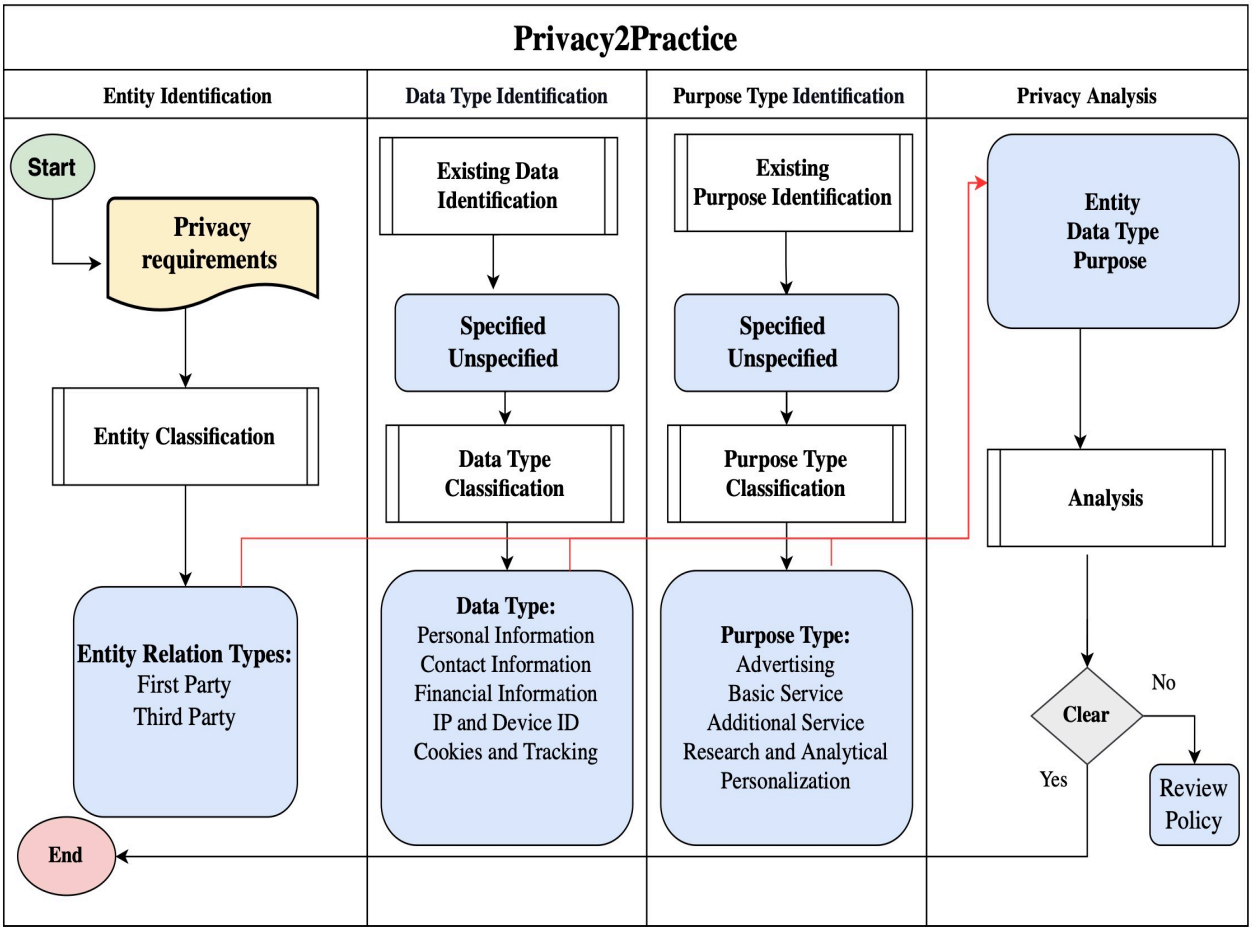

Figure 6.1: A Cross Functional Flowchart Presents Privacy2Practice

To identify data practices in privacy policy documents, we employed *Few-shot learning* [54], a specialized machine learning approach focused on making accurate predictions with limited training samples. The objective of *Few-shot learning* is not merely to recognize instances in the training set but to enable the model to "learn to learn," allowing it to generalize from a few examples to new, unseen data. In this research, we adopted *SetFit* , which is proposed by Tunstall et al. [64], as our *Few-shot learning* algorithm.

Given that *Privacy2Practice* outlines the following data practices: entity type, data type, and purpose type. Extracting data practices from privacy policy documents involves four downstream tasks to extract these practices automatically. The construction of the proposed approach (*Privacy2Practice*) encompasses four tasks:

- *Task 1: Entities Type Identification* focuses on extracting essential entities, including first party and third party entities, from the policy in natural language text. A binary classification, coupled with dependency parsing, is used to pinpoint specific data collection or sharing entities.

- *Task 2: Data Type Identification* aims to identify the presence of data within segments and classify the specific type of data mentioned.

- *Task 3: Purpose Type Identification* examines whether segments articulate any purpose behind data collection and sharing activities and categorizes the specific type of purpose mentioned.

- *Task 4: Policy Analysis* involves analyzing whether the policy covers all the identified data practices.

These tasks collectively facilitate extracting data practices from privacy policy documents, providing valuable insights into the alignment with with the GDPR and the CCPA. The respective sub-sections of this chapter provide further details regarding each task and the experimental methodology.

6.2 Datasets

Access to accurately labeled data is crucial to enhance our supervised NLP algorithms for the initial three tasks in our pipeline. After conducting an exhaustive review of existing literature, we have pinpointed a meticulously curated dataset concerning privacy policies (OOP-115) dataset [67], which we denote as Dataset 1. Moreover, to enrich our analysis for the final task (Policy

Analysis), we collected various policies from various technology companies, which we denote as Dataset 2.

6.2.1 Dataset 1

In this study, we utilize the publicly available OOP-115 dataset [67], a comprehensive resource in privacy policy. Crafted by Wilson et al. [67], this dataset comprises a diverse collection of 115 privacy policies spanning 15 sectors, including arts, shopping, business, and news. It encapsulates a wealth of information, including data practices shown in Table 6.1, and each data practice category has attributes such as data type, purpose type, and others meticulously curated by proficient annotators [67]. The final annotation scheme comprises a comprehensive breakdown into ten distinct categories of data practices, as illustrated in Table 6.1 below. Consequently, we employ this dataset as the cornerstone of our investigation, focusing on tasks aimed at identifying entities, data types, and purpose types, as elaborated in Sections 6.3, 6.4, and 6.5.

The OPP-115 dataset comprises a wide array of data practice categories as shown in Table 6.1. However, our focus was specifically on categories and attributes aligning with our research objectives. In particular, we honed in on entity categories, specifically first and third parties, and the attributes of data type and purpose type. These chosen categories were carefully extracted from the OPP-115 dataset to form the basis of our training data.

Table 6.2 offers a comprehensive view of the distribution across these categories within the OPP-115 dataset. The table details the number of samples associated with three pivotal aspects: *entities*, *data type*, and *purpose type*. For the *entities* category, which encompasses both *first party* and *third party*, the dataset contains 1790 samples. The *data type* category is represented by 801 samples, and the *purpose type* category comprises 827 samples. This distribution provides an insightful overview of the data practices we used from the OPP-115 dataset. Also, it establishes a foundational dataset essential for the training and evaluation of the *Privacy2Practice* models.

Table 6.1: Categories of Data Practices as Defined by Wilson et al. [67]

Category	Description
First Party (Collection)	Clarifies the entity who process and collect the data and the rationale behind the service provider's acquisition of user information.
Third Party (Sharing)	Elaborates on the mechanisms through which user information is shared with or gathered by third-party entities.
User Choice and Control	Outlines the choices and control options available to users regarding their information.
User Access, Edit, and Deletion	Describes whether and how users can access, modify, or delete their information.
Data Retention	Indicates the duration for which user information is retained.
Data Security	Explains the measures in place to safeguard user information.
Policy Change	Addresses how users will be notified about changes to the privacy policy.
Do Not Track	Specifies if and how Do Not Track signals for online tracking and advertising are respected.
International and Specific Audiences	Covers practices relevant to specific user groups such as children, Europeans, or California residents.
Other	Includes additional sub-labels for introductory or general text, contact information, and practices not falling under the other defined categories.

Table 6.2: Training Dataset Distribution

Data Practices	Sample Numbers
Entities (First Party- Third Party)	1790
Data Types	801
Purpose Types	827

6.2.2 Dataset 2

We collect comprehensive privacy policies from diverse companies operating within the technology sector. This dataset forms the cornerstone of our analysis.

To acquire this data, we engineered a crawler capable of extracting text from websites hosting privacy policy statements. Subsequently, we systematically parsed each document, breaking them into discrete paragraphs and statements to facilitate our analysis.

We strictly adhered to predefined criteria for selecting privacy policies throughout the data collection process. We segregated companies into two distinct groups. The first group encompasses entities known for their robust compliance with regulations such as the GDPR and the CCPA, ensuring comprehensive disclosure of their data practices within their privacy policies. Conversely, the second group comprises smaller companies that may need to provide more extensive disclosure regarding their data practices in their privacy policies. We must emphasize that we utilized this dataset in our analysis task, as discussed in Section 6.6. Table 6.3 provides a comprehensive breakdown, illustrating the number of statements collected from each company.

Table 6.3: A Set of Privacy Policy from Technology Companies

Group	Companies	Number of Sentences
Group 1	Google	239
	Microsoft	243
Group 2	CodeGeek	59
	GreystoneTech	33

6.3 Task 1: Entities Type Identification

Objective.

Identifying privacy data practices from privacy policy in natural language is crucial for organizations aiming to comprehend and regulate their data handling practices in alignment with privacy regulation, including the GDPR and the CCPA. Automating data practice identification allows organizations to bolster efficiency, minimize errors, and elevate transparency and accountability in their data processing operations [10].

Our initial focus revolves around identifying who is collecting the data and with whom it is shared, referred to as first party and third party entities, respectively. Thus, this task primarily centers on identifying the entities engaged in data collection and sharing, as delineated within privacy policies, commonly referred to as first party and third party entities.

Method.

In this task, we propose a two-level classification approach to precisely identify entities within privacy policy sentences. Firstly, we classify the entity type, aiming to differentiate between first party and third party entities, a process we term as *Entity Identification*. Once this foundational classification is established, we develop an algorithm to identify the specific entities mentioned within each sentence, a process we refer to as *Entity Detection*.

Entity Identification: The objective of this task is to categorize segments of text based on whether they refer to the entity responsible for collecting the data (referred to as the first party) or if they pertain to the external entity from which the data is sourced and subsequently obtained by the first party (referred to as the third party). To accomplish this goal, we adopt a binary text classification methodology. This entails designing a model that can effectively classify each text segment into one of two categories: referencing the first party or the third party. Given the inherent challenge posed by the limited availability of training data, we employ a *few-shot learning* approach. By utilizing this approach, we develop a binary classification model that effectively classifies text segments despite having access to only a limited dataset for training. In our approach, we leverage the *SetFit* model to build a classification model capable of accurately categorizing text segments into either first party or third party entities. The *SetFit* model has demonstrated notable efficiency and effectiveness in similar text classification tasks, making it a suitable choice for our purpose. To evaluate the performance of our entity identification process, we rely on standard evaluation metrics, including *precision*, *recall*, and *F1-score*.

Entity Detection: In this step, our focus is on defining that *first party* and *third party* entities within text segments. Our approach entails identifying *first party* entities by recognizing them as the subject or first person within a sentence, and it is followed by discerning their associated

actions. This involves analyzing the syntactic structure of the text to locate instances where the subject of the sentence corresponds to the first party mentioned. Conversely, for identifying *third party* entities, our methodology centers around pinpointing the name of organizations within the text. These organizations typically represent the third parties with whom data sharing agreements are established. To carry out this analysis, we employ a dependency parser applied to the pre-processed text data. The dependency parser enables us to analyze the grammatical structure and relationships between words in the text, providing valuable insights into how different elements within the text are connected and interact. We utilized *SpaCy* [32], which is an NLP library, to apply the dependency parser. *SpaCy* offers robust capabilities for linguistic analysis, including dependency parsing, which is elaborated upon in Chapter 2 and 4. By leveraging *SpaCy*, we can extract detailed syntactic information from the text data, facilitating the identification of the *first party* and *third party* entities within text segments.

Experimental Results.

Entity Classification: Fine-tuning a large language model for text classification requires training it on a designated dataset comprising instances pertinent to the intended task. In our scenario, the objective is to categorize text into two distinct classes: *first party* and *third party* entities. To accomplish this, we undertake an experiment wherein we fine-tuned the *SetFit* model [64] as a binary classifier using pre-defined labeled data representing *first party* and *third party* entities.

In our experiment, we fine-tuned *SetFit* model with the following hyperparameters: Train Data=1000 samples, Test=500 samples, Sentence Transformer = paraphrase-MiniLM-L3-v, num epochs = 10.

Table 6.4: The Performance of the Entity Type Identification Task

	Entity Model		
Classes	**P(%)**	**R(%)**	$\mathbf{F_1}(\%)$
First Party	100%	95%	97%
Third Party	87%	100%	93%

Based on the results of our experiment, we observed that our model demonstrated robust capability in discerning first party entities within the text, achieving an F1-score of 97%. Furthermore, it displayed proficiency in identifying third party entities, achieving an F1-score of 93%. Such high-performance metrics are indicative of the promising potential of our model.

Entity Extraction: To accomplish this, we utilize a dependency parser, leveraging the capabilities of *SpaCy*. To identify the *first party* responsible for the data collection and the *third party* with whom the data is shared, we employ the following steps using *SpaCy*.

1. Tokenization: the input sentence undergoes tokenization, breaking it into individual tokens.

2. Looping through Tokens: we iterate through each token within the tokenized sentence to conduct a detailed analysis.

3. Identifying First Party: Each token is inspected to determine if it corresponds to a first party pronoun ('I', 'we', 'us,', 'it'). Such pronouns typically denote the speaker or the first party mentioned in the context of the text. The algorithm scrutinizes the dependency relationships between tokens, aiming to ascertain whether the *first party* pronoun functions as the subject of a verb (nsubj). In linguistic terms, the nsubj dependency signifies that the token acts as the verb's subject in the sentence. The algorithm conducts further analysis in cases where the *first party* pronoun is not directly the subject of a verb but serves as the subject of an auxiliary verb (aux). It investigates whether this auxiliary verb functions as the head of a main verb. Auxiliary verbs typically precede main verbs in English sentences.

4. Identifying Third-Party: Each token in the text is meticulously inspected to determine whether it corresponds to entities tagged as organizations, which are denoted by the label "ORG," using NER. Subsequently, the text corresponding to these identified organization entities is extracted for our analysis.

Table 6.5 demonstrates an example of entity extraction from privacy policies using *SpaCy*. The "Privacy Policy" column contains sample sentences extracted from privacy policies, while

the "First Party" and "Third Party" columns display the entities identified by *SpaCy* as the first party (the entity collecting the data) and the third party (the entity with whom the data is shared), respectively. To illustrate, the privacy policy segment states, "We may collect data including but not limited to weight, steps, and height. Based on the initial setup on iOS, you enable us to link with Apple Health." In this example, *SpaCy* correctly identifies "We" as the first party collecting the data and "Apple Health" as the third party with whom the data is shared.

Table 6.5: Example of Entity Extraction using *SpaCy*

Privacy Policy	First Party	Third Party
"We may collect data including but not limited to weight, steps, and height. Based on the initial setup on iOS you enable us to link with Apple Health."	We	Apple Health

Comparison with Prior work.

In comparing our model with prior research efforts aimed at evaluating efficiency, our proposed entity identification model exhibits notable superiority over preceding methodologies, leveraging CNN and XLENT models, as detailed in [28] and [29] respectively. Notably, our model outperforms these approaches, particularly in terms of F1-score, as delineated in Table 6.6. This emphasizes the efficacy of the *SetFit* model in accurately pinpointing entity segments within text. Our results highlight the robustness of *SetFit* as a few-shot text classifier meticulously tailored for privacy policy content.

Table 6.6: Comparative analysis with previous studies regarding F1-score

	Approaches			
Entity Type	**CNN Algorithm** [28]	**XLENT Algorithm** [28]	**Polisis (CNN Algorithm** [29]	**Our Approach (SetFit)**
First Party	76%	83%	80%	97%
Third Party	78%	81%	81%	93%

6.4 Task 2: Data Type Identification

Objective.

In this task, we aim to categorize the various data instances referenced in privacy policies into specific classes, each representing a distinct type of information. To accomplish this, we concentrate on several key categories inspired by the categories in OPP-115 [67]. These categories include personal information, financial details, computer-related information such as IP and device IDs, cookies and tracking elements, and user online activities.

1. Personal Information: This category pertains to data instances containing personally identifiable information (PII) about individuals, such as names, addresses, social security numbers, or biometric data.

2. Financial Information: This category includes data instances related to financial transactions, accounts, payment methods, credit card numbers, or other sensitive financial data.

3. Computer Information: This category includes data instances associated with device identifiers, internet protocol (IP) addresses, or other unique identifiers linked to electronic devices used for accessing digital platforms.

4. Cookies and Tracking Elements: This category includes data instances related to cookies and other elements used for tracking user behavior and preferences.

5. User Online Activities: This category includes users engage with online services and websites, and how their activities are tracked and recorded by those services.

6. Unspecified: This category applies when no specific data is mentioned.

Method.

We adopt a two-level classification approach to categorize data mentioned in privacy policies systematically. Initially, we verify the existence of the data type within the segment. Subsequently, we classify the type of data present in the segment. This structured methodology ensures a comprehensive classification process, facilitating accurate identification and categorization of data within

privacy policies. In other words, two-level classification ensures that our efforts are directed solely towards relevant data segments.

1. First, in the *Existence Checking* phase, we determine whether data is present within the segment. This involves classifying the segment into specified data types and unspecified ones. We proceed to the subsequent classification level if any indication of data existence is found within the segment.

2. Second, in the *Data Type Classification* phase, we assign the segment to one of the predefined categories outlined earlier. This process entails discerning the nature of the data referenced within the segment and accurately assigning it to the appropriate category.

Experimental Results.

Our study employed the *SetFit* few-shot learning approach to tackle the task above. To determine whether a segment represents a specific data type, we leveraged the OPP-115 dataset. This dataset categorizes data types and indicates whether each type is specified or unspecified. We utilized this dataset to extract the data type attribute assigned to each segment, which we then employed as the label for the segment.

Initially *(Existence Checking phase)*, we fine-tuned *SetFit* to classify whether each segment represents a specified or unspecified data type. The performance measures using *precision*, *recall*, and *F1-score* are shown in Table 6.7.

Subsequently *(Data Type Classification phase)*, we further fine-tuned *SetFit* to categorize segments into predefined data type categories. This refinement enabled us to determine whether each segment corresponds to personal information, financial records, computer information, or tracking elements. We evaluated the performance of this classification using metrics such as *precision*, *recall*, and *F1-score* as shown in Table 6.8.

In our experiment, we fine-tuned model *SetFit* with the following hyperparameters: Train Data=600 samples, Test=200 samples, Sentence Transformer = paraphrase-MiniLM-L3v, num epochs = 10.

Drawing from our experiment's outcomes as shown in Table 6.8, we noted our model's remarkable capability in detecting existing data within text segments, achieving an F1-score of 82%. Moreover, it exhibited proficiency in discerning various types of information such as personal information, financial information, computer information, and cookie and tracking. Specifically, it attained F1-scores of 96%, 94%, and 98% respectively, for personal information, financial information, and computer information. However, when it came to identifying user online activities and cookies and tracking data, the model achieved a lower F1 score of 62% and 69%.

Comparison with Prior work.

It is important to note that prior studies have not focused on specifying the types of data handled or shared within privacy policies, nor have they explored automated methods for identifying such data types. By undertaking this task, our research makes a substantial contribution to the field by introducing an automated approach to categorize data types within privacy policies. This contribution marks a significant step forward in understanding and addressing the complexities of privacy policy analysis in an increasingly data-driven world.

Table 6.7: The Performance of the Performance of the Data Existence

	Data Type Existing		
Classes	**P(%)**	**R(%)**	F_1(%)
Specified Data Type	80%	84%	82%
Unspecified Data Type	88%	84%	86%

6.5 Task 3: Purpose Type Identification

This task aims to identify the purpose type mentioned within privacy policies into specific classes, each representing a distinct purpose type. To achieve this, we focus on the key categories that are inspired by the categories in OPP-115 [67]: advertising, basic services and features, additional services and features, analytical and research, personalization and customization, and legal requirements.

Table 6.8: The Performance of the Data Type Identification Task

	Data Type Classification		
	P(%)	**R(%)**	$F_1(\%)$
Personal Information	94%	98%	96%
Financial Information	93%	95%	94%
Computer Information	95%	100%	98%
Cookies and Tracking Elements	75%	64%	69%
User Online Activities	80%	50%	62%

1. Advertising: This class encapsulates data practices for promotional activities, market research, and consumer engagement strategies to enhance brand visibility and customer outreach.

2. Basic Services and Features: It categorizes data practices related to fundamental services and features within the segment, encompassing essential functionalities and core offerings.

3. Additional Services and Features: It categorizes data practices to identify supplementary services and features provided by the organization, which complement the primary offerings and enrich the overall user experience.

4. Analytical and Research: This category pertains to data practices focused on data analysis, research endeavors, and insights generation to understand market trends, consumer behavior, and performance metrics.

5. Personalization and Customization: This category pertains to data practices to tailor services, products, and user experiences to individual preferences and requirements, enhancing personalization and customization capabilities.

6. Legal Requirement: This category refers to the obligations imposed on businesses and organizations that collect, use, or share personal information online.

7. Unspecified: This category applies when no specific purpose is mentioned.

Method.

Through utilizing *few-shot learning* approach (*SetFit*), we aimed to develop a classification model capable of accurately assigning each purpose to its respective class, thereby facilitating a deeper understanding of the diverse objectives within the privacy policy segment. In this task, we follow the same method for *Data Type Identification Task*. We implement a two-level classification to categorize purpose mentioned within privacy policies as follows:

1. First, in the *Existence Checking* phase, we determine whether a specific purpose is present within the segment. This involves classifying the segment into specified purposes and unspecified ones. If any indication of data existence is found within the segment, we proceed to the subsequent classification level. This ensures that our classification efforts are directed solely toward segments containing the purpose of collection or sharing data.

2. Second, in the *Purpose Type Classification* phase, we assign the segment to one of the predefined categories outlined earlier. This process entails discerning the nature of the purpose referenced within the segment and accurately assigning it to the appropriate category.

In the initial phase *(Existence Checking phase)*, we fine-tuned SetFit to determine whether each segment signifies a specified or unspecified purpose type. The evaluation of this classification is presented in Table 6.9, using precision, recall, and F1-score. In the subsequent phase *(Purpose Type Classification phase)*, we further fine-tuned SetFit to classify segments into predefined purpose type categories. This enhancement allowed us to identify whether each segment relates to advertising, basic and additional services, and features, analytical and research, or personalization and customization. The performance of this classification is assessed using precision, recall, and F1-score, as detailed in Table 6.10.

In our experiment, we fine-tuned the model SetFit with the following hyperparameters: Train Data = 600 samples, Test = 200 samples, and num epochs = 10.

Experimental Results.

Based on the results of our experiment, we observed that our model demonstrated capability in discerning existing purpose in the text segment with a 90% F-1 score as shown in Table 6.9. Furthermore, it displayed proficiency in identifying advertising, basic service, additional services, and legal requirements, achieving an F1-score of 88%, 80%, and 95% 81%, respectively, as shown in Table 6.10. However, when it came to identifying analytical research and personalization purposes the model achieved a lower F1 score of 72% and 64% F1 score.

Table 6.9: The Performance of the Purpose Existence

	Purpose Existing		
Classes	**P(%)**	**R(%)**	$F_1(\%)$
Specified Purpose	91%	90%	90%
Unspecified Purpose	91%	92%	92%

Table 6.10: The Performance of the Purpose Type Identification Task

	Purpose Type Model		
Classes	**P(%)**	**R(%)**	$F_1(\%)$
Advertising	87%	89%	88%
Basic services and feature	86%	75%	80%
Additional services and feature	100%	91%	95%
Analytical and research	68%	76%	72%
Personalization and customization	73%	57%	64%
Legal requirement	81%	81%	81%

It is noteworthy that there has been no prior research dedicated to specifying the purposes for collecting or sharing data within privacy policies, and there has been no study on the automated

identification of such purpose types. Thus, this task constitutes another notable contribution to the automated privacy policy analysis.

6.6 Task 4: Privacy Policy Analysis

The provided section delves into a comprehensive analysis of privacy policies, explicitly scrutinizing the adequacy of their disclosure concerning data practices. Within the domain of privacy policies, it is paramount for organizations to communicate their data practices to users consistently. The principle of *"consistent disclosure"* underscores the critical importance of ensuring absolute clarity and transparency in the information provided within the privacy policy [2, 53]. Assessing a privacy policy document's consistency necessitates evaluating the clarity with which it discloses data practices. In this study, we leverage our previous models to determine whether the mere existence of data practices within a privacy policy indicates consistent disclosure. This approach enables us to probe deeper into the efficacy of privacy policies in transparently communicating data handling practices to users.

To assess the effectiveness of our proposed approach, we examine collections of privacy policies sourced from diverse technology companies. Our selection criteria prioritize alignment with regulatory frameworks such as the GDPR and the CCPA. Notably, the *first group* largely conforms to the stringent requirements of these regulations, encompassing major companies includes Google [2] and Microsoft [3]. In contrast, the *second group*, comprising predominantly smaller companies such as startups includes CodeGeek [4], and GreyStoneTech [5], may exhibit only partial compliance with the stipulations set forth by the GDPR and the CCPA.

We analyze the privacy policy in both groups (Group 1 and Group 2). Our analysis of the privacy policy statements from the first and second groups revealed the following observations:

---

[2]Google: https://policies.google.com/privacy?hl=en-US

[3]Microsoft: https://privacy.microsoft.com/en-us/privacystatement

[4]CodeGeek: https://www.codegeek.net/

[5]GreyStoneTech: https://www.greystonetech.com/

Group 1

The First Party: For *Google*, the primary entity responsible for collecting data is Google itself, indicating that the majority of data collection activities are carried out by the company. For *Microsoft*, the main entity responsible for data collection, indicating that the company carries out the majority of data collection activities.

The Third Party: For *Google*, the primary third party entities mentioned, with whom data is shared, are exclusively Google apps or Google services. No other third party entities are mentioned in this context. For *Microsoft*, the primary third party entities mentioned, with whom data is shared, are Microsoft apps or Microsoft services (Music, Movies, and Windows Media Player).

Data Type: For *Google*, the main data collection type is personal data, followed by cookies and tracking information, as well as online activity and computer information like IP addresses. Importantly, *Google* does not collect any financial information. Similarly, for *Microsoft*, the primary data collected is personal information and cookie and tracking data. Like *Google*, *Microsoft* also does not gather any financial information.

Purpose Type: For *Google*, the main goal of data collection is to enhance the user experience by offering additional services. Additionally, Google shares data for research and analytical purposes. For *Microsoft*, data collecting and sharing aims to provide additional services. Delivering basic services is also a key objective.

Evaluation: Based on our analysis, it is evident that Group 1 demonstrates a consistent disclosure regarding their data practices. This commitment is apparent in the following key aspects of their policies:

1. Group 1 effectively delineates the utilization of both first party and third party data within their policies. This means that users are provided with a comprehensive understanding of how their information is utilized by the platform and external entities.

2. Group 1 offers transparent insights into the collected data types. Such clarity lets users grasp the extent and nature of the data being handled.

3. Group 1 goes beyond mere acknowledgment of data sharing by providing explicit explanations for the purposes behind it. Users are informed about why their data is being collected, whether for analytics, basic and additional services, or other legitimate reasons.

In essence, Group 1 ensures that their data practices are clearly communicated. Therefore, it can be concluded that Group 1's disclosure of data practices is *consistent disclosure*.

Group 2

The First Party: For *CodeGeek*, the principal entity responsible for the collection of data is the company itself, signifying that the company assumes full accountability for this process. For *GreyStoneTeck*, the company itself is the main entity responsible for collecting the data.

The Third Party: For *CodeGeek*, the company shares data with third parties, with MailChimp being specifically disclosed as their third-party email service provider. For *GreyStoneTeck*, the company does not explicitly disclose the names of these third parties.

Data Type: For *CodeGeek*, the primary data collected is personal information, followed by cookies and tracking data and online activity records. For *GreyStoneTeck* primarily focuses on collecting personal information. What sets this company apart is its collection of financial details, including the last four digits of credit cards

Purpose Type: For *CodeGeek*, data collecting and sharing primarily aim to deliver both basic and additional services while also supporting analytical and research efforts. For *GreyStoneTech*, the main goal of data collection is to offer both basic and additional services.

Evaluation: Upon completing our analysis, it becomes evident that Group 2 exhibits inconsistency in disclosing their dprovides transparent insights into the types of data collected and shared, ineate the utilization of the third party data within their policies. Consequently, users are left without a clear understanding of with who their information is shared.

However, it is worth noting that Group 2 does provide transparent insights into the types of data collected and shared, thereby enabling users to comprehend the nature and extent of their data usage. Furthermore, Group 2 goes beyond mere acknowledgment of data sharing by elucidating

the purposes behind it, such as analytics or basic and additional services. This approach empowers users to grasp the rationale behind data collection and usage. Nevertheless, the inadequacy in effectively delineating the utilization of the third party data within their policies remains a significant shortcoming for Group 2. This gap prevents users from attaining a comprehensive understanding of their data sharing practices.

Group 2's failure to consistently disclose their data practices undermines their overall transparency. Consequently, it can be concluded that Group 2's disclosure of data practices is inconsistent and warrants improvement.

6.7 Summary

This chapter underscores the significance of safeguarding end-user's data privacy in today's digital landscape, with legislation such as the GDPR and the CCPA driving organizations to adopt transparent notice and consent systems. Privacy policies serve as crucial legal documents through which organizations communicate their data collection practices to users. However, the complexity and length of these policies often hinder users' understanding, prompting recent research efforts to leverage NLP for extracting critical information from privacy documents.

Privacy2Practice introduces an automated method for identifying data practices from privacy policy documents. This method aims to ensure consistent disclosure of data practices in alignment with the GDPR and the CCPA. The *Privacy2Practice* outlines key aspects such as data collection, sharing processes, entities involved, data types, and purposes. To achieve this, an NLP pipeline comprising four downstream tasks is employed.

Entities type identification focuses on extracting essential entities, such as first party and third party entities, from natural language text in privacy policies. Binary classification, coupled with dependency parsing, is utilized to pinpoint specific entities responsible for data collection or sharing. *Data type identification* to identify the presence of data within segments and classify the specific type of data mentioned. *Purpose type identification* examines whether segments articulate any purpose behind data-sharing activities and categorizes the specific type of purpose mentioned.

Policy analysis involves analyzing whether the policy covers all the identified data practices and determining if there is consistent disclosure.

These tasks collectively enable the extraction of data practices from privacy policy documents, offering valuable insights into regulatory compliance and facilitating the development of more robust privacy policies. *Privacy2Practice* demonstrates high performance in identifying first-party and third-party entities, achieving F1-scores of 97% and 93%, respectively. It also shows proficiency in identifying data types with an 82% F1-score and discerning the purpose of sharing with a 90% F1-score

Chapter 7

Conclusion and Future Work

This dissertation delves into the automated specification and analysis of security and privacy policies in natural language documents. It begins by exploring the evolving landscape of NLP, from dependency parsers to large language models. It investigates various automated techniques for extracting and analyzing security and privacy policies.

Security policies are usually extracted manually from security requirements documents and often contain unstructured and ambiguous information, making the process laborious, costly, and error-prone. Similarly, privacy policies, characterized by their length and intricacy, pose challenges for novices and experts to comprehend fully. This complexity drives the need to automate the extraction and analysis of security and privacy policies. Automatic extraction provides a solution, facilitating easier and more accurate policy extraction at a reduced cost in terms of time and complexity. Furthermore, automatic analysis aids developers in ensuring the correctness of the access control model, ensuring the alignment with the privacy regulations, streamlining the process, and enhancing effectiveness. Hence, this dissertation presents a novel framework incorporating two core components to address security and privacy policies specification and analysis, named *SR2ACM* and *Privacy2Practice*, intricately designed to confront these challenges adeptly.

For *security policy*, we developed the *SR2ACM* that automatically extracts ACPs from statements in natural language and converts ACPs into a formal NGAC model expressed as a graph. *SR2ACM* comprises five tasks: identifying ACP statements, discerning the types of relations within these statements, capturing the NGAC entities, generating the NGAC graph model of the access control policies, and conducting an in-depth analysis of the generated graph to evaluate the correctness of the derived policies. *SR2ACM* achieves 93% F1-score for ACPs identification and 96% F1-score for association identification, which is the predominant relation type in security requirements documents. *SR2ACM* successfully identifies user attributes with an average F1-score of 96% and object attributes with 89%. Furthermore, the analysis demonstrates that *SR2ACM* ex-

tracts ACPs correctly. The correctness of the extraction of education and healthcare policies has been ensured by validating four properties: *completeness*, *well-formedness*, *minimality*, and *consistency*. *SR2ACM* shows potential in efficiently extracting the NGAC access control models from available security requirements.

However, further investigation remains necessary. One potential avenue for future research involves exploring how *SR2ACM* can be adapted and enhanced to effectively manage changes in security requirements. Researchers can develop more dynamic and adaptable security solutions by extending this approach to incorporate features for handling updates to security requirements. Future work could focus on devising algorithms and techniques for automatically identifying and integrating changes in security policies into the existing *SR2ACM* approach. This may involve developing mechanisms for detecting modifications in security requirements and assessing their impact on existing access control policies. Additionally, research efforts could explore strategies for enhancing the scalability and efficiency of *SR2ACM* in handling dynamic security environments. This may include optimizing algorithms for processing and analyzing large volumes of security data, improving the performance of graph-based representations in representing complex access control policies, and implementing mechanisms for real-time monitoring and adaptation to evolving security threats.

For *privacy policy,* we developed *Privacy2Practice* that extracts data practices from privacy policy documents in natural language. Our proposed approach aims to ensure that privacy policy documents disclose data practices in alignment with regulations such as the GDPR and the CCPA. Leveraging a novel *few-shot learning* pipeline, this study analyzes the privacy policy documents. *Privacy2Practice* presents a method for deriving data practices directly from privacy policy documents using few-shot learning. *Privacy2Practice* prioritizes key data practices mandated by the GDPR and the CCPA, focusing on data collection and sharing processes. *Privacy2Practice* comprises four downstream tasks aimed at automatically extracting and categorizing data practices, including entity type identification, data type identification, purpose type identification, and privacy analysis. The *Privacy2Practice* showcases exceptional performance in recognizing both first

party and third party entities, achieving F1-scores of 97% and 93%, respectively. Moreover, it accurately identifies data types with an F1-score of 82% and discerns the purpose of sharing with an F1-score of 90%. Using *Privacy2Practice*, we analyze a collection of privacy policies obtained from various technology companies.

However, several promising avenues exist for future exploration in privacy policy analysis. A critical inquiry pertains to the detection of policy conflicts. An illustrative scenario involves instances where an organization asserts non-disclosure of user data to third parties in its policy yet practically compels users to engage with third party applications, thereby implicating data sharing. This discrepancy underscores the necessity for investigative efforts to devise methodologies to discern and address incongruities between stated policies and operational practices. Another significant study area involves delving into automated compliance mechanisms embedded within application behavior to ensure alignment with regulatory frameworks. This entails developing sophisticated algorithms and techniques to dynamically monitor and enforce compliance with privacy regulations as applications interact with user data. By integrating automated compliance checks directly into the behavior of software applications, organizations can proactively mitigate privacy risks and uphold regulatory standards in real-time. Moreover, another crucial direction for future research is translating privacy regulations from complex natural language documents into formal, machine-readable formats. This process involves converting the legal jargon and nuances in privacy laws into structured, standardized representations that software systems can easily interpret and implement. This, in turn, streamlines the implementation process and reduces the likelihood of misinterpretation or non-compliance, ultimately enhancing the overall effectiveness of privacy protection measures in digital environments.

In conclusion, these two components *SR2ACM* and *Privacy2Practice* present an automated framework for security and privacy policy specification and analysis, significantly contributing to advancing research and practical applications in security and privacy policy. This framework ensures understanding and evaluating policy implications, paving the way for enhanced security and privacy practices in various contexts. Finally, while automated security and privacy policy anal-

ysis offers significant advantages, addressing existing challenges requires ongoing research. By prioritizing interdisciplinary efforts, we can pave the way for a more secure and privacy-respecting digital ecosystem. Moving forward, the collaboration between academia, industry, and regulatory bodies is crucial to refining automated security and privacy policy analysis techniques.

References

[1] Abu Jabal, A., Bertino, E., Lobo, J., Law, M., Russo, A., Calo, S., Verma, D.: Polisma a framework for learning attribute-based access control policies. In: Proceedings of 25th European Symposium on Research in Computer Security. pp. 523–544. Springer, Guildford, UK (2020)

[2] Adjerid, I., Acquisti, A., Brandimarte, L., Loewenstein, G.: Sleights of privacy: Framing, disclosures, and the limits of transparency. In: Proceedings of the ninth symposium on usable privacy and security. pp. 1–11 (2013)

[3] Alohaly, M., Takabi, H., Blanco, E.: Automated extraction of attributes from natural language attribute-based access control (ABAC) policies. Cybersecurity **2**(1), 1–25 (2019)

[4] Amaral, O., Abualhaija, S., Briand, L.: Ml-based compliance verification of data processing agreements against GDPR. In: IEEE 31st International Requirements Engineering Conference (RE). pp. 53–64. IEEE, Germany (2023)

[5] Andow, B., Mahmud, S.Y., Wang, W., Whitaker, J., Enck, W., Reaves, B., Singh, K., Xie, T.: {PolicyLint}: investigating internal privacy policy contradictions on google play. In: 28th USENIX security symposium. pp. 585–602. USENIX Association, Santa Clara,CA,USA (2019)

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

[7] Antón, A.I., Earp, J.B., He, Q., Stufflebeam, W., Bolchini, D., Jensen, C.: Financial privacy policies and the need for standardization. IEEE Security & privacy **2**(2), 36–45 (2004)

[8] Bijon, K.Z., Krishnan, R., Sandhu, R.: Towards an attribute based constraints specification language. In: International Conference on Social Computing. pp. 108–113. IEEE (2013)

[9] Breaux, T.D., Antón, A.I.: Deriving semantic models from privacy policies. In: 6th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'05). pp. 67–76. IEEE (2005)

[10] Bui, D., Shin, K.G., Choi, J.M., Shin, J.: Automated extraction and presentation of data practices in privacy policies. Proceedings on Privacy Enhancing Technologies (2021)

[11] Bui, D., Yao, Y., Shin, K.G., Choi, J.M., Shin, J.: Consistency analysis of data-usage purposes in mobile apps. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security. pp. 2824–2843. ACM, Republic of Korea (2021)

[12] Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Androutsopoulos, I.: Large-scale multi-label text classification on EU legislation. arXiv preprint arXiv:1906.02192 (Jul 2019)

[13] Chen, E., Dubrovenski, V., Xu, D.: Mutation analysis of NGAC policies. In: Proceedings of the 26th ACM Symposium on Access Control Models and Technologies. p. 71–82. ACM, Spain (2021)

[14] Chiquito, A., Bodin, U., Schelén, O.: Access control model for time series databases using NGAC. In: ETFA. vol. 1, pp. 1001–1004. IEEE (2020)

[15] Cunningham, H., Mayard, D., Tablan, V.: Jape: a java annotation patterns engine. University of Sheffield (2000)

[16] Damianou, N., Dulay, N., Lupu, E., Sloman, M.: The ponder policy specification language. In: Policies for Distributed Systems and Networks: International Workshop, POLICY. pp. 18–38. Springer (2001)

[17] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv (2018)

[18] Ehrmann, M., Hamdi, A., Pontes, E.L., Romanello, M., Doucet, A.: Named entity recognition and classification on historical documents: A survey. arXiv preprint arXiv:2109.11406 (2021)

[19] Elluri, L., Joshi, K.P., Kotal, A.: Measuring semantic similarity across EU GDPR regulation and cloud privacy policies. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 3963–3978. IEEE (2020)

[20] Fernandes, D., Bernardino, J., et al.: Graph databases comparison: AllegroGraph, ArangoDB, InfiniteGraph, Neo4J, and OrientDB. In: Proceedings of DATA. pp. 373–380. ACM, Portugal (2018)

[21] Fernández, M., Mackie, I., Thuraisingham, B.: Specification and analysis of ABAC policies via the category-based metamodel. In: Proceedings of the 9th ACM conference on data and application security and privacy. pp. 173–184. ACM, Richardson, TX, USA (2019)

[22] Ferraiolo, D., Chandramouli, R., Kuhn, R., Hu, V.: Extensible access control markup language (XACML) and next generation access control NGAC. In: Proceedings of the ACM International Workshop on Attribute Based Access Control. pp. 13–24. ACM, New Orleans, (2016)

[23] Ferraiolo, D., Gavrila, S., Jansen, W.: Policy machine: Features, architecture, and specification (2015-10-27 2015)

[24] Ferraiolo, D.F., Chandramouli, R., Ahn, G.J., Gavrila, S.I.: The role control center: features and case studies. In: Proceedings of the 8th ACM symposium on Access control models and technologies. pp. 12–20. Como, Italy (2003)

[25] Fritzler, A., Logacheva, V., Kretov, M.: Few-shot classification in named entity recognition task. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing. pp. 993–1000. Limassol, Cyprus (2019)

[26] Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., Zettlemoyer, L.: AllenNLP: A deep semantic natural language processing platform (2018)

[27] Gardner, M., Grus, J., Neumann, M., Tafjord, O., Dasigi, P., Liu, N., Peters, M., Schmitz, M., Zettlemoyer, L.: AllenNLP: A deep semantic natural language processing platform. arXiv preprint arXiv:1803.07640 p. 101611 (2018)

[28] Hamdani, R.E., Mustapha, M., Amariles, D.R., Troussel, A., Meeùs, S., Krasnashchok, K.: A combined rule-based and machine learning approach for automated GDPR compliance checking. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law. pp. 40–49. São Paulo, Brazil (2021)

[29] Harkous, H., Fawaz, K., Lebret, R., Schaub, F., Shin, K.G., Aberer, K.: Polisis: Automated analysis and presentation of privacy policies using deep learning. In: 27th USENIX Security Symposium. pp. 531–548. USENIX Association, Baltimore, MD, USA (2018)

[30] He, L., Lee, K., Lewis, M., Zettlemoyer, L.: Deep semantic role labeling: What works and what's next. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. pp. 473–483. Vancouver, Canada (2017)

[31] Heaps, J., Krishnan, R., Huang, Y., Niu, J., Sandhu, R.: Access control policy generation from user stories using machine learning. In: Proceedings of Data and Applications Security and Privacy XXXV: 35th Annual IFIP WG 11.3 Conference. pp. 171–188. Springer, Calgary, Canada (2021)

[32] Honnibal, M., Montani, I.: SpaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing (2017), https://spacy.io/, accessed: 2023

[33] Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)

[34] Hu, V.C., Ferraiolo, D., Kuhn, R., Friedman, A.R., Lang, A.J., Cogdell, M.M., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., et al.: Guide to attribute based access control ABAC definition and considerations. NIST special publication **800**(162), 1–54 (2013)

[35] Jensen, C., Potts, C.: Privacy policies as decision-making tools: an evaluation of online privacy notices. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. pp. 471–478. Vienna, Austria (2004)

[36] Jordan, C.S.: Guide to Understanding Discretionary Access Control in Trusted Systems. Diane (1987)

[37] Kenton, J.D.M.W.C., Toutanova, L.K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. p. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)

[38] Larionov, D., Shelmanov, A., Chistova, E., Smirnov, I.: Semantic role labeling with pre-trained language models for known and unknown predicates. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019). pp. 619–628. Varna, Bulgaria (2019)

[39] Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: IEEE on computer vision. pp. 2980–2988 (2017)

[40] Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. ACM Computing Surveys **55**(9), 1–35 (2023)

[41] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., Tang, J.: GPT understands, too. arXiv preprint arXiv:2103.10385 (2021)

[42] Liu, Y., Cao, J., Liu, C., Ding, K., Jin, L.: Datasets for large language models: A comprehensive survey. arXiv preprint arXiv:2402.18041 (2024)

[43] Màrquez, L., Carreras, X., Litkowski, K.C., Stevenson, S.: Semantic role labeling: an introduction to the special issue. Computational linguistics **34**(2), 145–159 (2008)

[44] Meneely, A., Smith, B., Williams, L.: Appendix b: iTrust electronic health care system case study. Software and Systems Traceability **1**(3), 409–425 (2012)

[45] Mousavi Nejad, N., Scerri, S., Lehmann, J.: Knight: Mapping privacy policies to GDPR. In: Knowledge Engineering and Knowledge Management: 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings 21. pp. 258–272. Springer (2018)

[46] Narouei, M., Takabi, H.: Towards an automatic top-down role engineering approach using natural language processing techniques. In: Proceedings of the 20th ACM Symposium on Access Control Models and Technologies. pp. 157–160. ACM, New York, NY, USA (2015)

[47] Narouei, M., Takabi, H., Nielsen, R.: Automatic extraction of access control policies from natural language documents. IEEE Transactions on Dependable and Secure Computing **17**(3), 506–517 (2018)

[48] Nivre, J.: Dependency grammar and dependency parsing. MSI report **5133**(1959), 1–32 (2005)

[49] Nivre, J.: Dependency parsing. Language and Linguistics Compass **4**(3), 138–152 (2010)

[50] OpenAI: GPT3 (2018), https://openai.com/, accessed: 2024

[51] Osborn, S.: Mandatory access control and role-based access control revisited. In: Proceedings of the second ACM workshop on Role-based access control. pp. 31–40. USA (1997)

[52] Palmer, M., Gildea, D., Kingsbury, P.: The proposition bank: An annotated corpus of semantic roles. Computational linguistics **31**(1), 71–106 (2005)

[53] Pan, Y., Zinkhan, G.M.: Exploring the impact of online privacy disclosures on consumer trust. Journal of retailing **82**(4), 331–338 (2006)

[54] Parnami, A., Lee, M.: Learning from few examples: A summary of approaches to few-shot learning. arXiv preprint arXiv:2203.04291 (2022)

[55] Peng, Y., Yan, S., Lu, Z.: Transfer learning in biomedical natural language processing: an evaluation of BERT and ELMo on ten benchmarking datasets. arXiv preprint arXiv:1906.05474 pp. 58–65 (2019)

[56] Ray, I., Toahchoodee, M.: A spatio-temporal role-based access control model. In: IFIP Annual Conference on Data and Applications Security and Privacy. pp. 211–226. Springer, Berlin, Heidelberg (2007)

[57] Samarati, P., de Vimercati, S.C.: Access control: policies, models, and mechanisms. In: International School on Foundations of Security Analysis and Design. pp. 137–196 (2000)

[58] Sandhu, R., Ferraiolo, D., Kuhn, R., et al.: The NIST model for role-based access control: towards a unified standard. In: ACM workshop on role-based access control. vol. 10 (2000)

[59] Servos, D., Osborn, S.L.: Current research and open problems in attribute-based access control. ACM Computing Surveys (CSUR) **49**, 1–45 (2017)

[60] Slankas, J., Xiao, X., Williams, L., Xie, T.: Relation extraction for inferring access control rules from natural language artifacts. In: Proceedings of the 30th annual computer security applications conference. pp. 366–375. ACM, New Orleans, LA, USA (2014)

[61] Sunkle, S., Kholkar, D., Kulkarni, V.: Toward better mapping between regulations and operations of enterprises using vocabularies and semantic similarity. Complex Systems Informatics and Modeling Quarterly **5**, 39–60 (2015)

[62] Tankard, C.: What the GDPR means for businesses. Network Security **2016**(6), 5–8 (2016)

[63] de la Torre, L.: A guide to the california consumer privacy act of 2018. Available at SSRN 3275571 (2018)

[64] Tunstall, L., Reimers, N., Jo, U.E.S., Bates, L., Korat, D., Wasserblat, M., Pereg, O.: Efficient few-shot learning without prompts. arXiv preprint arXiv:2209.11055 (2022)

[65] Van De Stadt, R.: Cyberchair: A web-based groupware application to facilitate the paper reviewing process. CoRR **1206.1833**, 7 (2012), http://arxiv.org/abs/1206.1833

[66] Webber, J.: A programmatic introduction to Neo4j. In: Proceedings of the 3rd annual conference on Systems, programming, and applications: software for humanity. pp. 217–218. ACM, Arizona, USA (2012)

[67] Wilson, S., Schaub, F., Dara, A.A., Liu, F., Cherivirala, S., Leon, P.G., Andersen, M.S., Zimmeck, S., Sathyendra, K.M., Russell, N.C., et al.: The creation and analysis of a website privacy policy corpus. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. pp. 1330–1340. Berlin, Germany (2016)

[68] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the conference on empirical methods in natural language processing: system demonstrations. pp. 38–45 (2020)

[69] Wong, S.C., Gatt, A., Stamatescu, V., McDonnell, M.D.: Understanding data augmentation for classification: when to warp? In: Proceedings of the IEEE International Conference on Digital Image Computing: Techniques and Applications. pp. 1–6. IEEE, Gold Coast, Queensland (2016)

[70] Xia, Y., Zhai, S., Wang, Q., Hou, H., Wu, Z., Shen, Q.: Automated extraction of ABAC policies from natural-language documents in healthcare systems. In: Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine. pp. 1289–1296. IEEE, Las Vegas, NV, USA (2022)

[71] Xiao, X., Paradkar, A., Thummalapenta, S., Xie, T.: Automated extraction of security policies from natural-language software documents. In: Proceedings of the ACM SIGSOFT 20th

International Symposium on the Foundations of Software Engineering. pp. 1–11. ACM, Cary, NC, USA (2012)

[72] Yoo, K.M., Park, D., Kang, J., Lee, S.W., Park, W.: GPT3Mix: Leveraging large-scale language models for text augmentation. In: Findings of the Association for Computational Linguistics. pp. 2225–2239. Association for Computational Linguistics, Punta Cana, Dominican Republic (2021)