Bachelor Thesis

A STUDY ON THE PRIVACY POLICIES OF CLOUD PROVIDERS USING AN AI-BASED TOOL

Karolina Dimitriou

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

December 2024

UNIVERSITY OF CYPRUS

DEPARTMENT OF COMPUTER SCIENCE

A study on the privacy policies of cloud providers using an AI-based tool.

Karolina Dimitriou

Supervisors

Dr. George Papadopoulos

Co-supervisors

Evangelia Vanezi

Christos Mettouris

The Individual Diploma Thesis was submitted towards partially meeting the requirements for obtaining the degree of Computer Science of the Department of Computer Science of the University of Cyprus.

December 2024

Acknowledgements

I would like to express my deepest gratitude to Mr. George Papadopoulos of the Computer Science Department for the trust and supervision he provided during the preparation of this thesis. I also extend special thanks to Mrs. Evangelia Vanezi and Mr. Christos Mettouris for the valuable materials and studies they provided, which were essential for the preparation and development of my work as well as their guidance throughout my thesis.

Abstract

The term “privacy” over the years has acquired a lot of value in the modern digital space because most activities now take place on the Internet-from online shopping, to communication with people and sharing their experiences. Yet, unfortunately, there are very severe privacy concerns now that are linked with the use of the internet. The present work tries to understand it very thoroughly, especially from an angle that concerns online activities. Furthermore, it focuses on the General Data Protection Regulation (GDPR), a comprehensive regulatory framework established to solve this problem. In addition, it also introduces a sophisticated AI tool designed to facilitate GDPR compliance assessments for cloud service providers’ privacy policies.

The regulations and principles under GDPR are very complex and strict since they concern the processing of personal data, which gives citizens good control over their data. Meanwhile, these put deep obligations on the actors to whom personal data are entrusted. In this thesis, an exhaustive treatment is devoted to the thorough explanation of the complex requirements of GDPR pertaining to matters of online privacy.

In this thesis, it is also presented a custom AI tool that employs the techniques of natural language processing and performs a detailed analysis of complex legal documents such as terms of services and privacy policies. It has the capacity of extracting and identifying terms relevant to the GDPR compliance to make it much easier to review them under the established regulations.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

The reason for choosing the thesis topic “A Study on Privacy Policies of Cloud Providers using an AI-based tool” is the increasing concern for the issues regarding data privacy in the modern world. Cloud computing has become essential in the current world whereby individuals and organizations are able to store and manipulate large data easily. But with this convenience comes the difficulty of sifting through the numerous privacy policies that define the handling of data by cloud providers. This is a challenge that has the likelihood of affecting the privacy and security of individuals and organizations. Therefore, I have been encouraged to address this issue to try to find the solutions to the problem that affects everyone.

Privacy policies are formal contracts made between the users and the cloud providers and are usually regarded as legal and technical documents filled with jargon words. This opacity leads to many people and organizations having concerns on how their data is being handled thus raising important issues on trust and transparency. The inspiration to conduct this thesis comes from the need to break down these policies and make them easy to understand. By using AI technology, I equip users with tools to break down the complex language, to extract the necessary information and to enable them to make the right decisions regarding their data. In other words, this thesis is based on the efforts to close the gap between the rights of users and the obligations of cloud providers.

Additionally, my drive goes beyond just personal curiosity. I want my research to have a wider impact. My goal is to help create a digital world where people are more aware of their privacy. I want individuals to be more than just users of technology; I want them to take an active role in protecting their own information. By doing this, I hope to help people make better decisions about their data and give organizations the tools they need to follow privacy laws effectively. In the end, I'm motivated by the idea that by using AI to better understand cloud privacy policies, we can build a digital environment that is more open, responsible, and safe for everyone involved.

1.2 Concept

The foundation of this thesis lies in using Natural Language Processing (NLP) to tackle a significant challenge: categorizing text data according to the guidelines of the General Data Protection Regulation (GDPR). In an age where organizations deal with massive volumes of unstructured text, the ability to pinpoint and organize GDPR-relevant information is more important than ever.

The proposed system uses word embeddings. Words are represented as dense multidimensional vectors, which encode meaning and relationships-and hence act as bases for interpretation of text that reaches beyond the surface, catching subtle differences needed to classify data in an accurate way.

The following four important ideas bring about this concept:

- Text as Structured Data: By processing text into structured formats, the system could extract and hold the meaning and context critical to any analysis.

- Detecting Patterns: The system generates patterns and relationships in the text, which fall under predefined GDPR categories, thus doing the identification accurately

- Adaptation and Scalability: Keeping a view on the scalability requirements, the system is adaptable to the changing format of data sets and loads of text that modern-day generates.

- Minimizing Manual Effort: The whole approach is building an effective model for complete automation so that human workforce input for time-consuming activities is avoided for compliance with GDPR.

The present research describes a rather structured approach to GDPR classification, paving the way for the methodological details thereafter. The main aim of this idea is to reduce unnecessary complications and make things as efficient as possible by using technology to address the growing need for protecting people's private information.

1.3 Methodology

This thesis investigates privacy policies in the field of cloud computing, focusing primarily on the compliance of such policies with the General Data Protection Regulation (GDPR) adopted by the European Union. The ubiquitous nature of cloud computing which epitomizes traveling data across international borders makes it a critical consideration for cloud service providers to comply with the GDPR requirement to provide grounds for seeking compliance with the regulation. This research is then concerned with privacy issues in data storage and processing, the ramifications of GDPR, and the compliance regarding this regulation. This research would explore key articles of the GDPR, which outline the rights of data subjects alongside the data protection principles as well as obligations that data controllers and processors are conditioned. Cloud providers normally act either as data controllers or processors.

A literature review and case studies on GDPR compliance concerning cloud computing were carried out to give the theoretical framework for tool development. Deliberately, the research also included the construction of a dataset specifically for training the AI model. Such a dataset consisted of privacy policies that were annotated for several labels pertaining to different rules according to the GDPR.

The practical aspect was an exhaustive appraisal of notable privacy policies from cloud service providers. This phase was designed to know the commonly occurring terms and clauses in those policies, which could be linked with specific regulations in the GDPR. The learning from this analysis fed into the development of the AI tool, with the intent of making it capable of parsing privacy policies and automatically evaluating them for compliance.

The process for creating the tool for classification under GDPR included many faces:

Data Collection and Preparation:
At the beginning of the construction of the tool, a consolidated case of text documents on "GDPR" was assembled. The documents included privacy policies, regulatory guidelines, and annotated examples of GDPR-made text and it was created manually. The document underwent preprocessing steps like tokenization, normalization, and stop-word removal, making them uniform.

Annotation and Labelling:
The annotated dataset was annotated manually based on the defined categories in which sections of text could belong to the corresponding GDPR headings. The labels included key articles from GDPR, such as "Right to Access," Data Minimization," and "Accountability Principles." This step was very important in ensuring accuracy and reliability of the database for training.

Word Embedding Generation:
Word2Vec generated embeddings, capturing semantic meaning and other contextual relationships related to the text. These embeddings then enabled the tool to probe deeper beyond keywords into text as it operated on more dense vector representations, thus becoming attuned to the complexities of the GDPR language.

Model Designing and Training:
Supervised machine learning classifiers were created using Word2Vec embeddings as input features. A model was designed for the labelled dataset to learn the patterns and

relationships connecting text features to GDPR categories and hyperparameter-tuning to improve the performance of the model.

Validation and Testing:
Validation was done stringently by deploying a separate test dataset on the trained model. Accuracy metrics, precision, recall, and F1 score had been used to test performances and the findings provided evidence for gradual improvements in the architecture of the model as well as the training process.

Integration of Tools and Functionality:
The final tool was built for users in mind and could easily integrate automatically parse privacy policies, figure out the sections pertaining to the GDPR, and categorize them as such. The tool was designed with features in mind for highlighting areas of non-compliance and providing practical recommendations essential for aligning with GDPR.

Through a systematic approach to designing the tool for classifying GDPR software, one could ensure the availability of certified solutions for organizations to automate their navigation through GDPR compliance complexity.

1.4 Structure

The next chapter explores various concepts that form the foundation of this project. It covers topics such as privacy, an introduction to the General Data Protection Regulation (GDPR), an overview of the GDPR regulations applicable to cloud providers, and key Artificial Intelligence concepts relevant to the project. Additionally, it presents an analysis of similar tools and includes a comparison with the tool developed in this thesis.

In Chapter 3, the creation process of a dataset for Privacy Policies based on GDPR is explained in detail, followed by an in-depth discussion of GDPR regulations related to online cloud providers.

Chapter 4 outlines the methodology for developing the automated evaluation tool for privacy policies based on the General Data Protection Regulation in full detail. Initially, the specification and analysis of the requirements, the architecture of the software, and the selected tools are introduced. Additionally, there is an extensive explanation of the software implementation.

In Chapter 5, the collection and analysis of the results are presented, extracting some information for the evaluation of an online platform.

Finally, Chapter 6 provides an overview of the analysis of the conclusions from the study of the topic and the application of the tool, while also proposing some suggestions for future work.

Chapter 2

Background and Related Work

2.1 Background

2.1.1 In General About Privacy

The right to privacy is a basic human right that defines as one very important aspect of guarding individual autonomy, dignity, and freedom, which was formally declared in the year 1948 from the United Nations General Assembly in Paris **[1]**. This includes the individual's ability to control his or her personal information and decide when, how, and with whom it will be shared. In essence, privacy enables individuals to express their thoughts, beliefs, and feelings without fear of unwarranted intrusion or judgment. According to the article by James Rachel in 1975 entitled "Why Privacy Is Important," it is the foundational basis for trust in personal relationships, professional dealings, and societal structures **[2]**. Privacy also means protection of individuals from many acts of discrimination and harassment as well as abuse, and, in a sense, acts as an important pillar towards personal and societal health.

Technology has transformed the pure facet of privacy because it understands and challenges it. Technology, on the other hand, proves accessibility at an unrivalled point in information, communication, and convenience. Of course, the current age of the digital age brings significant threats to privacy. Personal data is produced and compiled 24/7 via smartphones, social media, Internet-of-Things devices, and data-driven services. Such data possessions are tempting coffers for cybercriminals with malicious intentions of financial gain or other forms of exploitation. Even more so, the very technologies created for our convenience can be used to monitor and violate one's privacy without one knowing through facial recognition, location tracking, and data analytics. These subjects are also reflected in the article **[3]** Beyond Surveillance: Privacy, Ethics, and Regulations in Face.

2.1.2 General Data Protection Regulation (GDPR)

General Data Protection Regulation (GDPR) is a fully applicable legal framework imposed by the European Union (EU) to protect the privacy and personal data of individuals within the Member States of the European Union and the European Economic Area. It went into effect on May 25, 2018, and thus typically governs

individuals' personal data rights and requirements for the transparent and fair handling of such data by the organizations concerned **[4]**. It consists of 99 articles and 173 recitals, covering various aspects like consent, data subject rights, and data processing obligations. It is therefore clear that the regulation has global implications for what is regarded as data protection since it applies to any entity that processes personal data about EU citizens, wherever that processing occurs. The Key Principles in the GDPR include Preconditions for explicit consent for data processing; access rights, changes, and erasure for persons; obligations to organizations concerning transparency and data-processing practices with individuals, including multinational organizations, cloud service providers, and end-users worldwide. In other words, global standards for data protection will depend on this regulation's applicability to cover or govern data across sectors such as cloud service providers.

2.1.3 Cloud Providers and GDPR Compliance

Cloud computing has become a backbone of modern digital infrastructure, providing scalable storage and processing power to organizations and individuals worldwide. Cloud service providers play a pivotal role in storing and processing vast amounts of personal data, often acting as data controllers or processors as defined by the General Data Protection Regulation (GDPR). This section explores the implications of GDPR for cloud providers, the challenges they face, and the role of privacy policies in ensuring compliance.

2.1.3.1 Introduction to Cloud Services and Data Processing

Essentially, cloud services allow organizations to accommodate themselves towards the internet for the storage, processing, and sharing of data with no major installations being made within their premises. These features are the very reasons why these services become so much innovative and valuable as such into:
their scalability, their efficiency, and of course, their cost-effectiveness. This important feature, however, of using resources dependent on shared data, data storage, and so on does raise serious issues of data privacy and, of course, security, as discussed in the article "**[5]** Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." Cloud providers may be entrusted with high-

sensitivity personal data since it becomes subjected to the regulations of the European General Data Protection Regulation (GDPR) for any data pertaining to an EU resident.

2.1.3.2 GDPR Obligations for Cloud Providers

Among other things, by way of regulation, the cloud providers should guarantee that personal data is protected by undertaking specific obligations such as suitable technical and organizational measures, getting the user's clear and unambiguous consent, and ensuring that users can enjoy data access, rectification, and erasure. Other than that, there are more stringent stipulations on the transfer of data, which become applicable when an entity operates from outside the EU.

Such obligations hence create the necessity of having robust protection measures such as encryption, pseudonymization, and regular security assessments. Providers also need to contractually bind the third-party processors to adhere to GDPR requirements through legally binding agreements. **[6]**

2.1.3.3 Challenges in GDPR Compliance

Cloud providers experience enormous difficulties in complying with the GDPR legislation when there is a need to transfer data across borders. Such data transfers require migration under legal avenues such as Standard Contractual Clauses (SCCs) or adequacy decisions, which are usually hard and time-consuming. This is further compounded when data processing is outsourced to third parties **[7]**.

It is on a similar level that the dynamic and continuously changing face of cyber threats and technologies requires all organizations to constantly develop, upgrade, and/or change to remain compliant. Evading these issues will incur costly fines and have adverse effects on reputation as well as loss of user trust.

2.1.3.4 Importance of Privacy Policies in Cloud Providers

Privacy policies are the clearest portals of entry for cloud providers and users, with respect to how they collect, process, and secure personal data. Without doubt,

compliance of privacy policies within the ambit of the GDPR is critical for the building of trust with users while ensuring transparency.

Structured such that it captures legal obligations and requirements, the policy encourages accountability and credibility. Cloud providers must continually adapt from time to time the privacy policy in line with the much concise and clear requirements by the GDPR on communication because of changes in data processing and changes in regulations and compliance requirements.

2.1.4 Artificial Intelligence and Its Applications in Legal Text Analysis

AI has entered the doors of many industries; almost all have benefited from it in varying measures: either processing things quickly and effectively; eliminating redundancy; enhancing human decision-making; or improving data analysis insights. All these features make AI the front-runner in analyzing legal documents and interpreting huge chunks of legal materials at incredible speed and accuracy. This section thus aims to present the basics of AI with respect to my thesis and demonstrate how the specific uses of these technologies can be put to analyze privacy policies for compliance with the General Data Protection Regulation (GDPR).

2.1.4.1 Introduction to Artificial Intelligence (AI)

Artificial Intelligence is when a machine mimics human intelligence by simulating thinking and learning by programming them to do so, as does a human being. So, it's a broad scope of technologies that, through machine learning, natural language processing (NLP), and automation-wherever the task can be performed and otherwise need a cognitive element of a human being-have AI-empowered machines do the job. Here, it relativizes itself towards legal text-reading. AI reads, processes, and interprets that typically complex legal language, distances itself from it, and helps in that decision by leveraging or streamlining an evaluation of privacy policies regarding GDPR compliance.

2.1.4.2 Natural Language Processing (NLP)

Natural Language Processing is that discipline of artificial intelligence which deals with making the machine interact with the human language. In other words, interpreting the

meaning of language generated by a person from input to output in the machine. Let machines see the interpretation and possible generated ways of text-analysis techniques in automated analytical tools such as text classifications, sentiment analysis, and named-entity recognitions (NER), respectively. The above will enable AI models to parse a legal text, observe and catch all the key terms and conditions, and assess whether the given sentence follows rules of the GDPR or not. With that, the interpretation of legal language will give territorial efficiency and make analyses better with less time and energy spent on each evaluation.

2.1.4.3 Machine Learning and Text Classification

Machine Learning (ML) is a component of AI, concerned in training, modelling, and predicting data patterns through algorithms based on learned experience. For instance, ML can be practically trained on privacy policies that say whether it contains compliant or non-compliant particular sections. In the light of text classification, this refers to the part that analyses the privacy policy that learns clauses, terminologies that either pass or fail to meet the standards set by GDPR. By its automated classification, the process of reviewing privacy policies is considered accelerated and much more accurate due to the lowering of human error and subjectivity.

2.1.4.4 Word Embeddings

Word embeddings are probably the first thing that machines learn in understanding and processing text data: words are seen as the always-dense-long-continuous vectors in some high-dimensional space. The treatment of words as discrete entities is typical of bag-of-words or term frequency-inverse document frequency (TF-IDF) methods. However, word embeddings include the relationship of words within a broader meaning concerning that word.

The heart of the word embeddings is captured in terms of similarity between words. For example, in a high-dimensional vector space, "king" is much closer to "queen" or "doctor" to "nurse" than to "bicycle", which brings them much farther apart, because they are related meaning-wise. The general concept of embedding is that they have

trained on large corpora with many texts, in which co-occurrences of words in many different contexts must define how they are going to be represented in the vector space.

There are many famous and reputed models which can create word embeddings, such as Word2Vec, GloVe, and FastText. However, Word2Vec [22] is very popular out of all these because of its simplicity and effectiveness: predicting a word from its surrounding context or vice versa using either Continuous Bag-of-Words (CBOW) or Skip-Gram algorithms is a different way to reach the objective. It has been useful in different tasks in natural language processing, such as text classification, sentiment analysis, and information retrieval, by providing meaningful numeric representations of textual data for machines to analyze and process.

The embeddings provide the means for understanding the fine line between a classification decision and the similarity that word embeddings can uncover for legal texts and regulation-understandable terms. They minimalistically allow terms to be related in fine differences, making it possible for effective classification in the context of GDPR classification.

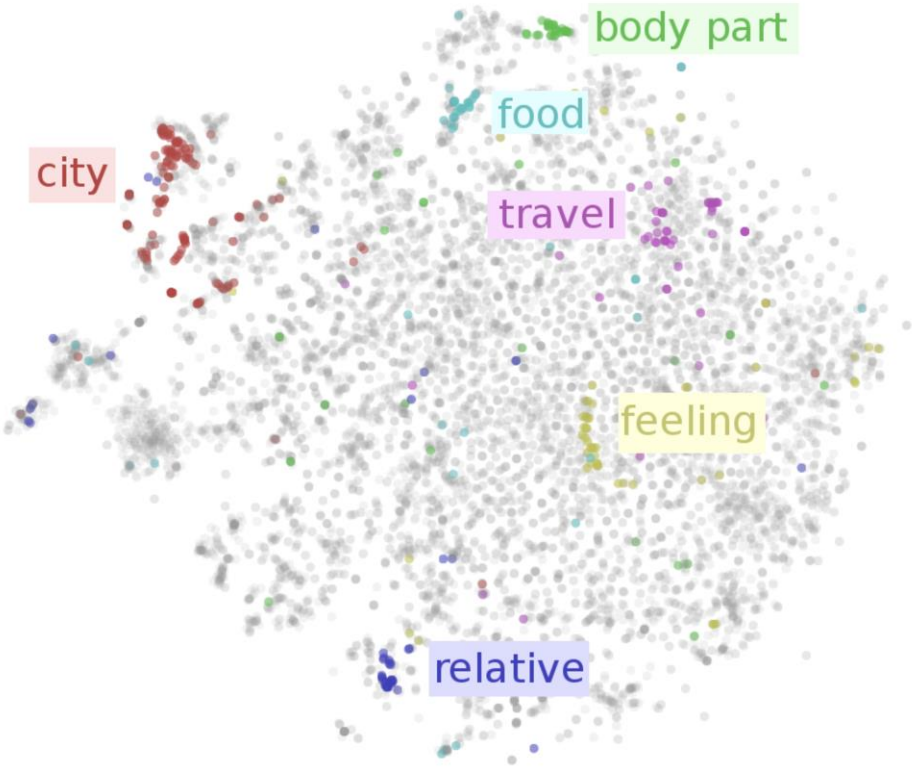


Image 2.1: A high dimensional space of embeddings with clusters of similar words.

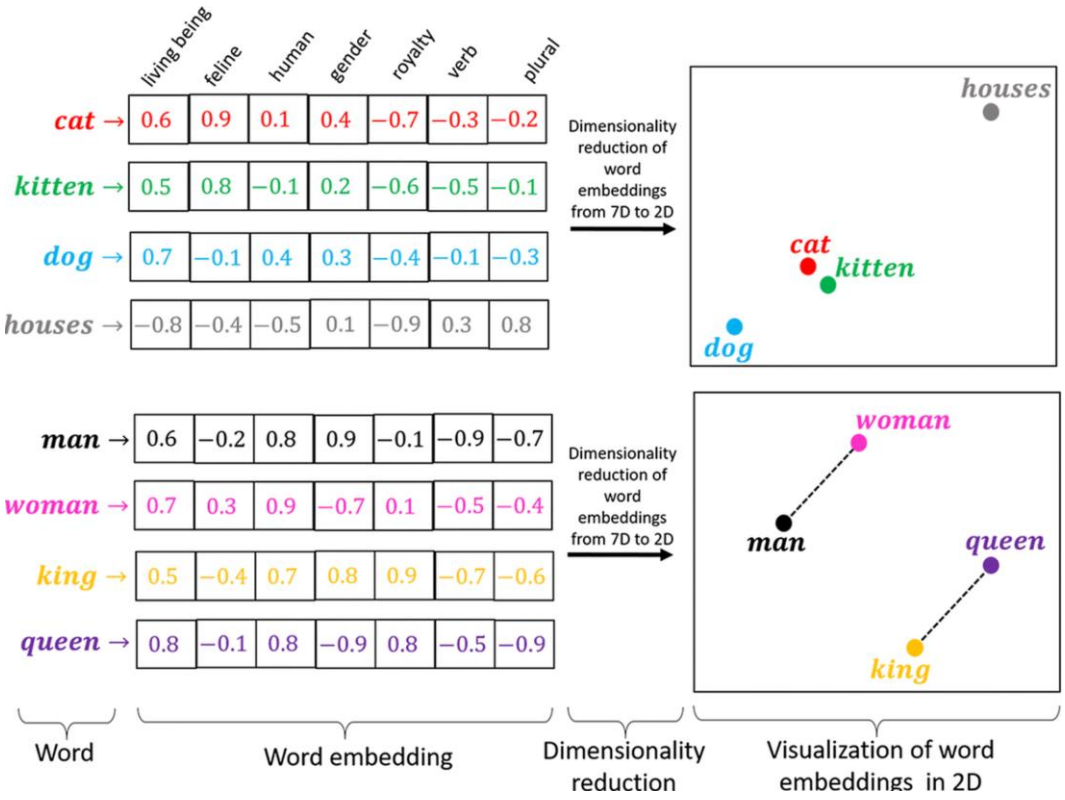


Image 2.2: A basic demonstrational example of how word embeddings work.

2.1.4.5 AI in Legal Text Analysis

AI has made great strides in adopting the legal field. One of the most significant applications is in processing and analyzing huge volumes of legal documents. This AI-powered tool helps extract important points from long pieces of text, find out whether certain regulations apply, for example, the GDPR, and much more. This tool is very helpful for legal practitioners, as it provides all-inclusive summaries of documents with coral areas of concern and attention to compliance collation. In such cases, AI helps organizations meet high standards regarding compliance and reduces the risk of ignoring vital details by ensuring that privacy policies are harmonized with regulations.

2.1.4.6 Supervised vs. Unsupervised Learning

Machine Learning can basically be classified into supervised and unsupervised learning. It is training models that use supervisor information, meaning that the data for input is labelled with the corresponding output. For me, supervised learning comes into play as it would involve the training of models to classify sections of privacy policies based on

whether they are considered GDPR compliant or not. This will use a set of privacy policies that were manually reviewed by a reviewer and labelled as compliant or non-compliant, which allows the model to learn what features indicate compliance or non-compliance.

The second category as I minus unsupervised learning deals with data that is not labelled and aims at segregating the data for possible hidden patterns or inherent structure in the data. Although exploratory data analytics are extremely effective using the unsupervised learning model, supervised learning fits more into my thesis because it gives the specific classifications needed for evaluating compliance with policies concerning GDPR.

2.1.4.7 AI Ethics in Legal Contexts

Using AI in legal contexts comes with ethical concerns, especially around data privacy, bias, and accountability. When analyzing privacy policies, AI systems must avoid creating biases or violating the privacy they aim to protect. Ethical AI requires transparent training, regular checks to reduce bias, and following data protection laws like GDPR. It's also important to have systems in place to fix mistakes, ensuring AI supports legal compliance rather than harming it.

2.2 Similar Tools

There are numerous AI-powered tools designed to address compliance challenges, showing just how important and necessary these solutions have become. I selected five tools that stood out either because they closely align with the concept of my thesis, offer unique features that complement my approach, or present interesting variations worth highlighting. These examples illustrate the growing need for innovative compliance tools and the different ways AI is being used to meet this demand.

2.2.1 Automated GDPR Compliance Checking

This tool **[8]** provides an automated way to assess privacy policy documents for GDPR compliance. By leveraging machine learning and natural language processing, it

identifies whether specific sections of a policy adhere to GDPR requirements. Like my thesis tool, this solution uses AI to analyze privacy policies, but it is more generic and adaptable to any privacy policy format. The main difference lies in the scope: while my tool focuses on cloud providers' privacy policies, this tool has a broader application across multiple sectors and contexts.

2.2.2 AI Privacy Toolkit by IBM

The AI Privacy Toolkit by IBM **[9]** is a robust framework for evaluating the privacy risks of AI models, offering tools for fairness, explainability, and compliance assessment. While it shares the compliance focus with my tool, the toolkit is geared towards AI systems rather than textual privacy policies. The difference is evident in its purpose—ensuring that AI applications themselves align with privacy standards rather than evaluating privacy policy documents.

2.2.3 Centraleyes

Centraleyes **[10]** is a comprehensive risk management platform that uses AI to help organizations assess and monitor compliance risks. It covers various compliance frameworks, including GDPR, ISO 27001, and more, making it more versatile than my tool. Both tools leverage AI for compliance-related tasks, but Centraleyes provides broader risk management services, whereas my thesis tool focuses specifically on GDPR compliance for privacy policies in the cloud domain.

2.2.4 Compliance.ai

Compliance.ai **[11]** is a regulatory change management platform that automates the monitoring and management of regulatory updates. Using AI, it maps changes to internal policies and procedures, helping organizations stay current with compliance obligations. Like my tool, it uses AI to streamline compliance processes, but it differs in focus. Compliance.ai addresses dynamic regulatory updates across various industries, while my tool is a specialized solution targeting GDPR compliance in static privacy policy documents.

2.2.5 AuditBoard

AuditBoard **[12]** is a cloud-based platform designed for enterprise compliance management, including audit management, risk assessment, and regulatory compliance. Its AI-driven capabilities help streamline workflows and track compliance efforts effectively. Like my tool, AuditBoard incorporates automation for compliance tasks, but it primarily focuses on enterprise-level compliance management rather than analyzing privacy policies for GDPR adherence.

Chapter 3

Research Methodology and Analysis

3.1 Introduction

This chapter provides an understanding of methodology and approach for investigating the privacy policies of cloud providers according to the General Data Protection Regulation (GDPR). The research has as its objective the identification of specific relevant regulations that apply to cloud providers and analysis of how these are

represented in the privacy policies of the cloud providers. It also describes the procedure used to create a labelled dataset susceptible to training a machine learning model to identify GDPR compliance in privacy policies.

3.2 Research Methodology

3.2.1 Approach to Identify Relevant Regulations

This research contribution has been initiated by sifting through the yet-to-be dug provisions of the GDPR relevant to cloud service providers. Aggregating different aspects of rights and obligations in the principles of data protection, GDPR houses articles not applicable in identical manners to cloud service providers. Above all, since such companies generate and transfer immense data volumes, the last major part of the legal regulation, namely data protection, privacy, and security, defines the norms expected within this cloud service industry.

99 articles together of GDPR spell out most widely operational changes in the personal data protection rights from an individual to an organization's responsibility for sending or processing data with one another. The articles, important for the operations of cloud service providers, include those on processing and security of data, availability, and rights data subject have regarding the data collected about him or her by the organization.

Cloud services are typically categorized into:

- Infrastructure-as-a-Service (IaaS), where cloud providers offer computing infrastructure, storage, and networking.
- Platform-as-a-Service (PaaS), where cloud providers offer software development platforms.
- Software-as-a-Service (SaaS), where cloud providers deliver software applications hosted in the cloud.

All these shared modes deal with different types of processing of personal data-inside these types involve the collecting, storing, moving, or deleting data, among which would have to be aligned with the requirements of the GDPR.

Given the varied nature of cloud services, a subset of GDPR articles was identified for further analysis. These regulations were chosen based on how directly they impact the operational activities of cloud service providers. The process of selecting the relevant articles involved a close examination of the primary GDPR regulations and their alignment with common practices in cloud computing, such as:

- Data Processing: Cloud providers are involved in the collection, storage, and processing of personal data. GDPR outlines strict guidelines regarding how data must be processed and for what purposes.

- Security: Cloud providers must implement appropriate technical and organizational measures to protect personal data from data breaches and unauthorized access.

- Transparency: Cloud providers are required to be transparent about their data processing activities, including providing clear privacy notices to their users and ensuring that their practices comply with GDPR.

- Cross-Border Data Transfers: Since cloud providers often operate globally, they must ensure that any transfers of personal data outside the EU/EEA comply with GDPR's rules on international data transfers.

The selected regulations directly address these operational concerns. Section 3.3, Detailed Analysis of GDPR Regulations, provides an in-depth examination of the GDPR regulations that specifically apply to cloud providers. It explains how and why these regulations are relevant, along with a detailed discussion of what each regulation entails.

3.2.2 Dataset Description

Collecting the dataset for this research was exhaustive and intricate in its practical applicability by itself in line with GDPR classification. The sources of such data include privacy policies and documentation of actual data use by major cloud service providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud. These

companies' policies provide comprehensive scope as well as concrete examples for implementing GDPR compliance.

Process of creating the dataset:

Selecting the Sources:
Initially, I obtained privacy policies and terms of service directly from their official websites. The focus was on documents which concerned GDPR obligations, for instance, actions taken with personal data, rights of users on that data, or protection measures of that data.

Carefully Reading Them One by One:
Each document is thoroughly read for relevant sections. I flagged any parts that referred to data processing, consent, or security. These flagged parts were then matched to the respective GDPR article. This was a very patient exercise requiring very strong understanding of the GDPR guidelines.

Annotating the Material:
After identifying relevant sections, I tagged them according to relevant GDPR categories. A snippet on how users can request deletion of their data is tagged "Erasure", while that which deals with ensuring data accuracy is noted "Rectification." This ensured relevant snippets were meaningfully tagged.

Normalizing the Text:
All was normalized by changing the formats of the text and the terminology used therein. For instance, normalization made all versions of varying phrases in different providers to allow for easy whole analysis of the data. This reduced the amount of noise in the dataset.

Adding More Examples to the Dataset:
To enhance the size and the diversity of the dataset that would be used for the model in training, some imaginary examples have been included. Theoretical cases which did not derive from actual policy were thoughtfully constituted to seem as close as possible to

realistic GDPR scenarios. For instance, certain bits were invented to cater to some of the less common GDPR requirements to ensure that the model had a wide diversity in training. However, from the tone and the type of actual policies, they all turned out to be highly valuable but incoherent.

Examples:

1, "We process data lawfully. You can have it erased at any time.", 1, 1, 0, 0, 0, 0, 0, 0

This snippet mentions lawful processing and the right to erasure, so both categories are marked with a 1.

2, "Our system ensures you may request and download all personal information.", 0, 0, 0, 0, 0, 1, 0, 0

This one is about data portability, which is reflected by the 1 in the "Portability" column.

Each column corresponds to a specific GDPR category, with 1 meaning the snippet is relevant to that category and 0 meaning it’s not. This clear labelling system made it straightforward to feed the data into the classification model.

By blending real-world examples with carefully crafted imaginary ones, I ensured the dataset was both comprehensive and high quality. The manual effort involved in this process guaranteed a level of precision and context-awareness that automated methods alone could not achieve.

3.3 Detailed Analysis of GDPR Regulations

This segment discusses an in-depth analysis of GDPR regulations selected for this research, highlighting the importance it has on cloud providers. Each regulation is elaborated upon with the focus on applicability as well as ways in which it is usually covered in cloud providers' privacy policies.

3.3.1 Article 5: Lawfulness, fairness and transparency

This article **[13]** establishes seven principles that guide data processing activities and ensure the protection of personal data. These principles must be adhered to by all organizations, including cloud providers, when processing personal data. The seven principles are:

1. Lawfulness, Fairness, and Transparency:
    - Lawfulness means that personal data must be processed in a lawful manner, which can only happen if there is a valid legal basis for processing (e.g., consent, performance of a contract, legal obligation, etc.).
    - Fairness means that data should be processed in a way that is fair to the data subject. This includes providing individuals with clear and accurate information about how their data will be used.
    - Transparency requires that data subjects are informed about how their personal data will be used, and organizations must make this information easily accessible and understandable.
2. Purpose Limitation:
    - Personal data should only be collected for specified, legitimate purposes and not processed further in a way that is incompatible with those purposes.
3. Data Minimization:
    - Personal data should be adequate, relevant, and limited to what is necessary for the purposes for which it is processed.
4. Accuracy:
    - Personal data should be accurate and, where necessary, kept up to date. Every reasonable step must be taken to ensure that inaccurate data is erased or rectified without delay.
5. Storage Limitation:

- Personal data should be kept in a form which permits identification of data subjects for no longer than necessary for the purposes for which the data is processed.

6. Integrity and Confidentiality (Security):
    - Personal data should be processed in a manner that ensures appropriate security, including protection against unauthorized or unlawful processing, and against accidental loss, destruction, or damage.
7. Accountability:
    - The data controller is responsible for, and must be able to demonstrate compliance with, the above principles.

Cloud providers play a central role in processing and storing personal data on behalf of their users. This makes Article 5 critical to ensuring that cloud providers comply with GDPR and respect users' data protection rights.

Lawfulness, Fairness, and Transparency: Cloud providers must ensure that users are informed about how their data will be used and that they have a legal basis for processing data.

Purpose Limitation: Cloud providers must not use personal data for purposes outside the scope of the service the user has agreed to. For example, if a user stores files on a cloud service, the provider cannot use this data for targeted advertising without obtaining the user's explicit consent.

Data Minimization: Cloud providers must only collect the necessary data to perform their services and avoid excessive data collection that could increase risk.

Accuracy: Cloud providers need to allow users to update their personal information and ensure the data they hold is accurate.

Storage Limitation: Cloud providers should ensure that personal data is stored only for as long as necessary. Data not needed for service delivery should be deleted or anonymized.

Integrity and Confidentiality (Security): Security is a major concern for cloud providers, who must implement robust security measures to protect user data from breaches and unauthorized access.

Accountability: Cloud providers must demonstrate compliance with these principles by keeping detailed records and undergoing audits to ensure that all aspects of GDPR are being adhered to.

3.3.2 Article 6: Lawfulness of Processing

Article 6 **[14]** sets out the conditions under which the processing of personal data is lawful. For processing to be compliant with GDPR, it must meet at least one of the following six lawful bases:

1. Consent:
    - The data subject has given their explicit consent to the processing of their personal data for one or more specific purposes.
2. Contractual Necessity:
    - Processing is necessary for the performance of a contract to which the data subject is a party, or to take steps at the request of the data subject prior to entering a contract.
3. Legal Obligation:
    - Processing is necessary for compliance with a legal obligation to which the controller is subject.
4. Vital Interests:
    - Processing is necessary to protect the vital interests of the data subject or of another natural person.
5. Public Task:

- Processing is necessary for the performance of a task carried out in the public interest or in the exercise of official authority vested in the controller.

6. Legitimate Interests:
    - Processing is necessary for the purposes of the legitimate interests pursued by the data controller or by a third party, except where such interests are overridden by the interests or fundamental rights and freedoms of the data subject which require protection of personal data, where the data subject is a child.

Cloud providers need to assess which lawful basis applies to their data processing activities. This assessment must be made before collecting or processing personal data. Below are some scenarios relevant to cloud providers:

Consent: If a cloud provider collects sensitive personal data (e.g., health data, political opinions) or wants to use data for marketing purposes, they must obtain explicit consent from users.

Contractual Necessity: Most cloud service providers process personal data based on this lawful basis. When a user subscribes to a service, the provider is legally allowed to process personal data (e.g., account creation, billing, service provisioning).

Legal Obligation: Cloud providers may need to store data for regulatory compliance or respond to law enforcement requests. For instance, they may need to store data for financial auditing or comply with government data retention requirements.

Legitimate Interests: Cloud providers may rely on this basis for processing personal data for activities such as fraud detection, security measures (e.g., monitoring for cyber threats), or enhancing user experience. However, the provider must conduct a balancing test to ensure that their interests do not override the privacy rights of users.

3.3.3 Article 7: Conditions for consent

Article 7 **[15]** elaborates on how consent should be acquired from data subjects (individuals) for processing their personal data. It specifies that consent must be:

1. Freely Given
    - Consent must be given voluntarily, and the individual must have a genuine choice. Users should not be forced or pressured into giving consent. If users are given no real option (e.g., if a service is denied unless consent is provided), then the consent cannot be considered valid.
2. Specific
    - Consent must be specific to a particular processing activity. A broad, blanket consent for all possible data processing is not sufficient. Users must be informed about exactly what their data will be used for, and they should consent to specific purposes, rather than an undefined or general use.
3. Informed
    - The individual must be provided with all the necessary information to make an informed decision. This includes details about the data processing purposes, the types of personal data involved, and any third parties who might receive or process the data. The information must be clear and accessible.
4. Unambiguous
    - Consent must be given through a clear affirmative action, such as checking a box, clicking an "I agree" button, or other similar actions. It must be obvious that the data subject intended to give consent. Silence, pre-ticked boxes, or inactivity cannot constitute consent.
5. Easily Withdrawn
    - Consent must be as easy to withdraw as it is to give. Users must be able to revoke their consent at any time without difficulty, and they must be informed of this right at the time consent is obtained. When consent is withdrawn, the data processing must cease unless there is another lawful basis for processing the data.

6. Documented
    - The data controller must be able to demonstrate that consent has been obtained, and it must keep records of when and how consent was obtained, as well as the information provided to the data subject at the time of consent.

Article 7 is essential for cloud providers, particularly when processing personal data based on user consent. Cloud providers must ensure that consent is freely given, specific, informed, and unambiguous, and that it can be easily withdrawn at any time. They must also maintain detailed records of consent to demonstrate compliance with GDPR. This article directly impacts how cloud providers manage user privacy and transparency, and non-compliance with these provisions could result in significant legal and financial consequences for the provider. By adhering to the requirements of **Article 7**, cloud providers can foster trust with users and ensure that personal data is handled responsibly and ethically.

3.3.4 Articles 15: Right of Access by the Data Subject

Article 15 **[16]** of the GDPR gives individuals the right to:

1. Access the Personal Data: Individuals have the right to obtain confirmation from cloud providers on whether personal data concerning them is being processed. For cloud providers, this could involve providing users with access to their stored data, the data processed, and the associated metadata (e.g., logs of data access or transfer).
2. Information on Processing Activities: Individuals can request information on how their personal data is being processed, for what purposes, and how long it will be stored. This directly affects cloud providers, who must detail these practices in their privacy policies.
3. Right to Receive a Copy of Personal Data: Upon request, cloud providers must provide users with a copy of their personal data in a commonly used format. This is particularly important for cloud providers offering storage services or data management solutions. The data subject must be able to access their data, which could be stored in a variety of cloud-based systems.

4. Data Portability: If the data subject requests, they can also request to receive their data in a structured, commonly used, and machine-readable format, which they can then transfer to another provider. This has implications for cloud providers offering data storage and management services.
5. Third-party Data Sharing: Cloud providers must disclose if and to whom the personal data has been shared, including any third parties or international transfers. If cloud providers share data with other organizations or partners (such as sub-processors), they must inform the data subject about these transfers.

In conclusion, **Article 15** of the GDPR is indeed applicable to cloud providers, as it is integral to ensuring transparency and accountability regarding how personal data is processed, stored, and accessed. Cloud providers must be prepared to comply with these requests and integrate them into their data management and privacy policies.

3.3.5 Article 16: Right to Rectification

Article 16 **[17]** grants individuals the right to have their personal data rectified without undue delay if it is inaccurate or incomplete. Specifically:

1. **Right to Rectify Inaccurate Data**: If a cloud provider is storing personal data that is incorrect or outdated, the data subject has the right to ask the provider to correct it. For example, if a user's contact details or preferences are incorrect in a cloud service, they can request that these details be updated.
2. **Right to Complete Incomplete Data**: If a cloud provider holds incomplete personal data, and the data subject feels that the data needs to be completed (e.g., missing details in a profile or data set), the data subject has the right to request that the cloud provider complete that data. This could be particularly relevant for services where users input personal information across various features (e.g., user profiles, billing information, or content sharing).
3. **No Undue Delay**: Cloud providers are obligated to act on rectification requests promptly, typically within a month (unless the request is complex or there is a valid reason for delay, as per the GDPR's guidelines). This is critical for cloud services as it reflects their responsibility in maintaining accurate and up-to-date data for their users.

4. **Notifications to Other Parties**: If a cloud provider has shared the rectified data with third parties (e.g., partners, sub-processors, or other services), they must inform these parties of the rectification, where feasible. This ensures that all entities handling personal data are using accurate information.
5. **No Charge**: The data subject does not need to pay to request the rectification of inaccurate or incomplete data, unless the request is clearly unfounded or excessive. Cloud providers must ensure that their policies reflect that data subjects can make corrections free of charge.

Article 16 applies directly to cloud providers by requiring them to maintain the accuracy and completeness of personal data. Cloud providers must have processes in place to manage and rectify personal data errors or omissions, and they must act on requests for correction in a timely manner to comply with GDPR. This article ensures that users' personal data remains reliable and up to date in the cloud environment.

3.3.6 Articles 17: Right to Erasure

Article 17 **[18]** provides data subjects with the right to request the erasure of their personal data in certain situations. The key provisions include:

1. Right to Request Erasure: A data subject can request the deletion of their personal data when:
    - The personal data is no longer necessary for the purposes for which it was collected or processed.
    - The data subject withdraws their consent (where processing was based on consent), and there is no other legal basis for processing.
    - The data subject objects to the processing (based on legitimate interests, for example), and there are no overriding legitimate grounds for the processing.
    - The personal data was unlawfully processed.
    - The personal data must be erased for compliance with a legal obligation.
2. Obligations of the Data Controller: When the right to erasure is exercised, the data controller (in this case, the cloud provider) must erase the personal data without undue delay, unless one of the following exceptions applies:

- Freedom of Expression: Erasure is not required when it conflicts with the right to freedom of expression and information.
- Legal Obligations: Data must be retained for compliance with legal obligations, such as tax or accounting regulations.
- Public Interest in Health or Research: Data may need to be retained for reasons related to public health, scientific research, or statistical purposes.
- Establishment, Exercise, or Defense of Legal Claims: Data may be retained if necessary for the establishment, exercise, or defense of legal claims.

3. Notification of Erasure: If the data has been disclosed to third parties, cloud providers must inform those third parties about the erasure request, where possible. Cloud providers should take reasonable steps to notify other recipients that the data has been erased, particularly when data has been shared with sub-processors or other services.
4. Timeframe for Erasure: Cloud providers must act on the erasure request without undue delay, and in most cases, they must fulfil the request within one month. If the request is complex or numerous, the provider may extend this period by two additional months, but they must inform the data subject within the initial month.

Article 17 is directly applicable to cloud providers, as it gives data subjects the right to request the erasure of their personal data. Cloud providers must have robust mechanisms in place to handle erasure requests, ensure compliance with data retention policies, and protect user rights in relation to their personal data. By implementing the necessary technical and organizational measures, cloud providers can ensure they meet the requirements of **Article 17** and uphold GDPR compliance.

3.3.7 Article 18: Right to Restriction of Processing

Article 18 **[19]** provides data subjects with the right to restrict the processing of their personal data under the following conditions:

1. **Conditions for Restriction**: A data subject can request the restriction of processing when:
    - **Accuracy of the Data**: The data subject contests the accuracy of their personal data and the controller (the cloud provider, in this case) must verify the accuracy of the data. During the verification period, the processing of the data should be restricted.
    - **Illegality of Processing**: The processing is unlawful, but the data subject opposes the erasure of their personal data and instead requests that its processing be restricted.
    - **Objection to Processing**: The data subject has objected to the processing of their data (based on legitimate interests or public interest grounds), and the controller is verifying whether the legitimate grounds for processing override those of the data subject.
    - **Data no longer needed for Processing**: The personal data is no longer necessary for the purposes of processing, but the data subject requires the data to be retained for the establishment, exercise, or defense of legal claims.
2. **Implications of Restriction**: When processing is restricted, the data can still be stored but cannot be used for processing unless one of the following conditions apply:
    - The data subject gives consent.
    - Processing is necessary for the establishment, exercise, or defense of legal claims.
    - Processing is necessary for the protection of the rights of another person or for important public interest reasons.
3. **Notification of Restriction**: Once the restriction is applied, the cloud provider must inform the data subject about the restriction and ensure that they have been notified if the restriction is lifted.
4. **Communication with Third Parties**: If the personal data is shared with third parties, the cloud provider must notify them about the restriction, unless it proves impossible or requires disproportionate effort.
5. **Timeframe for Action**: If a data subject requests restriction of processing, the cloud provider must act without undue delay and, in most cases, within one

month of receiving the request. This period can be extended by two additional months if the request is complex.

Article 18 is directly relevant to cloud providers, as it empowers data subjects to request the restriction of the processing of their personal data in certain situations. Cloud providers must be equipped with systems to handle such requests and ensure that they temporarily suspend processing activities when required. This article underscores the need for cloud providers to implement robust data access, management, and compliance frameworks that allow for flexible control over how personal data is processed and stored. Non-compliance with **Article 18** can result in significant legal and reputational risks for cloud providers, making it crucial for them to develop effective procedures for managing restriction requests.

3.3.8 Article 20: Right to Data Portability

Article 20 **[20]** provides data subjects with the following rights:

1. **Right to Obtain Data**:
    - A data subject has the right to receive their personal data from the data controller (the cloud provider, in this case) in a **structured, commonly used, and machine-readable format**. This format is typically a format that can be easily transferred to another system or service, such as CSV, JSON, or XML.
    - The data must be provided to the data subject **without undue delay**, and at the latest, within **one month** of the request being made.
2. **Right to Transfer Data**:
    - A data subject can request that their personal data be transmitted directly from one data controller (such as one cloud provider) to another, where technically feasible. This allows individuals to move their personal data from one provider to another without needing to manually download and upload the data.
3. **Scope of Application**:
    - This right applies only to data that the individual has provided to the data controller (e.g., user-generated content or personal details) and the data

that is processed based on the individual's **consent** or **contractual necessity** (such as a cloud service agreement).

Article 20 of the GDPR is particularly important for cloud providers, as it enables individuals to move their personal data freely between services. Cloud providers must ensure they have the technical capability to facilitate data portability in a secure and efficient manner. By providing data in a machine-readable format and ensuring it can be transferred directly to another service, providers meet the regulatory requirements and offer users greater control over their personal data. Compliance with this article promotes transparency, trust, and strengthens data subject rights, which are central to the GDPR.

3.3.9 Article 21: Right to Object

Article 21 **[21]** outlines the conditions under which a data subject has the right to object to the processing of their personal data and the obligations of the data controller (e.g., cloud providers) when an objection is raised. There are two primary types of objections covered under this article:

1. **Right to Object to Processing Based on Legitimate Interest**:
    - Data subjects have the right to object to the processing of their personal data when the processing is based on **legitimate interests** pursued by the data controller or a third party. In such cases, the cloud provider must cease processing unless they can demonstrate compelling legitimate grounds for the processing that override the interests or rights of the data subject.
    - For example, a cloud provider might process user data for the purpose of improving their services (legitimate interest), but if a user objects to this processing, the provider must stop unless it can demonstrate a compelling reason to continue (e.g., legal obligations or security needs).
2. **Right to Object to Direct Marketing**:
    - Data subjects have an absolute right to object to the processing of their personal data for direct marketing purposes. If the processing is for direct marketing, including profiling related to marketing, the cloud provider

must stop processing the data for this purpose immediately, without needing to justify the processing.

- For instance, if a cloud provider sends promotional emails or uses personal data for targeted advertising, a user can object to this processing, and the provider must cease the marketing activities.

3. **Special Conditions for Scientific, Historical, or Statistical Research**:
    - A data subject can object to the processing of their data for scientific, historical, or statistical research purposes unless the processing is necessary for the performance of a task carried out for reasons of public interest or official authority.

Article 21 of the GDPR is vital for protecting individuals' rights to control how their personal data is used. Cloud providers must ensure that they can manage objections to data processing, especially for direct marketing and legitimate interests. By respecting users' objections, cloud providers can avoid legal issues, foster trust, and improve their compliance with GDPR. However, this right also presents operational challenges, as cloud providers must implement systems to handle objections and ensure data is not processed against the wishes of the user unless a compelling reason exists to continue processing.

3.4 Dataset and Manual Labelling Process

The dataset consists of privacy policies from leading cloud service providers. Each policy was manually reviewed, and key sections were identified that address GDPR compliance. The labelling process involved tagging specific articles from the GDPR based on the content of the privacy policies. For example, sections discussing data processing were labelled with Article 6 (lawfulness of processing), while sections about user rights were labelled with Article 15 (right of access).
The labels used in this process include the key rights outlined in the GDPR **[4]**: the right to be informed, the right of access, the right to rectification, the right to erasure, the right to restrict processing, the right to data portability, and the right to object. These labels were essential to structure the dataset and enable the training of the machine learning model. By providing clear associations between specific GDPR articles and

sections of the privacy policies, the labelling process ensures the model can accurately identify GDPR compliance in new privacy policies. This manual approach, though time-intensive, forms the foundation for creating a reliable and robust tool for compliance evaluation.

Label	Description
Lawfulness	Related to lawful processing of data.
Erasure	Pertains to the right to have personal data erased (right to be forgotten).
Access	Covers the right of data subjects to access their personal data.
Object	Refers to the right to object to data processing.
Rectification	Concerns the right to have inaccurate personal data corrected.
Portability	Involves the right to data portability, allowing data subjects to receive their data in a structured format.
Transparency	Covers the requirements for clear and transparent communication about data handling practices.

Table 3.1: All the categories of the dataset.

```
SnippetID,Text,Lawfulness,Erasure,Access,Object,Rectification,Portability,SecurityBreach,Transparency
1,"We process data lawfully. You can have it erased at any time.",1,1,0,0,0,0,0,0
2,"Our system ensures you may request and download all personal information.",0,0,0,0,0,1,0,0
3,"We keep your data secure. If anything goes wrong, we alert you of a breach immediately.",0,0,0,0,0,0,1,0
4,"Users can rectify their information and also object to marketing emails.",0,0,0,1,1,0,0,0
5,"Data is handled with transparency. You can request access if needed.",0,0,1,0,0,0,0,1
6,"Our site fully adheres to lawful data principles and maintains accountability.",1,0,0,0,0,0,0,1
7,"Request the deletion (erasure) or access to your account info anytime.",0,1,1,0,0,0,0,0
8,"We provide secure processing to prevent breaches. You can rectify mistakes if found.",0,0,0,0,1,0,1,0
9,"Portability is guaranteed. You can object to certain forms of data usage too.",0,0,0,1,0,1,0,0
10,"Lawfully collected data, plus you can erase or rectify as per GDPR standards.",1,1,0,0,1,0,0,0
11,"We handle personal info with care and abide by lawfulness at all times.",1,0,0,0,0,0,0,0
12,"Right to object is respected. We encourage transparency about our processing.",0,0,0,1,0,0,0,1
13,"Security checks are conducted regularly. We also ensure you can rectify errors.",0,0,0,0,1,0,1,0
14,"We keep minimal data. Users can erase their profiles if they choose.",0,1,0,0,0,0,0,0
15,"Lawful basis is provided, and data is portable upon request.",1,0,0,0,0,1,0,0
16,"Please note you can object to certain analytics or ask for rectification.",0,0,0,1,1,0,0,0
17,"All data is processed under lawfulness. Security is tested against breaches.",1,0,0,0,0,0,1,0
18,"We offer transparency. Erase any personal information with a single request.",0,1,0,0,0,0,0,1
19,"Download your full data to exercise portability or rectify mistakes at will.",0,0,0,0,1,1,0,0
20,"Object to marketing messages or request secure handling to avoid breaches.",0,0,0,1,0,0,1,0
21,"We enforce strong breach prevention. Lawful processing is our top priority.",1,0,0,0,0,0,1,0
22,"Request account access or have data erased. We remain transparent on all operations.",0,1,1,0,0,0,0,1
23,"Please note data portability is supported, plus you can object anytime.",0,0,0,1,0,1,0,0
24,"Our lawfulness policy includes rectification for incorrect data as needed.",1,0,0,0,1,0,0,0
25,"Users can have transparency into how info is used. Erasure is also allowed.",0,1,0,0,0,0,0,1
26,"We follow lawful standards. Access your personal details or rectify them whenever.",1,0,1,0,1,0,0,0
27,"You can object to certain data usage. We ensure data is portable upon demand.",0,0,0,1,0,1,0,0
28,"Security measures are robust, so data breaches are minimized. Rectify errors as well.",0,0,0,0,0,1,0,1,0
29,"We handle everything lawfully. Transparency is crucial for building user trust.",1,0,0,0,0,0,0,1
30,"Erase your data or object to processing. We keep security checks active.",0,1,0,1,0,0,1,0
31,"Request rectification at any time, or ask for portable copies of your info.",0,0,0,0,1,1,0,0
32,"Our site is transparent about how data is used. Erasure is always an option.",0,1,0,0,0,0,0,1
33,"We maintain lawful conditions. You can access your personal records by request.",1,0,1,0,0,0,0,0
```

Image 3.1 Sample of the dataset.

3.5 Analysis of Cloud Provider Privacy Policies

3.5.1 Comparison of Cloud Provider Privacy Policies

A comparison of privacy policies from major cloud providers, such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud, reveals several common practices. All providers emphasize transparency in their data processing practices, detailing the types of data collected, how it is used, and how long it is retained. However, there are some variations in the level of detail provided, particularly regarding cross-border data transfers and the specific technical measures taken to protect personal data.

CLOUD PROVIDER	Information Collection	Use of Personal Information	Sharing of Information	Data Security Measures	User Rights
AWS	Collects user data through interactions and automatic means.	Used for service delivery, improvement, and marketing.	Shares with affiliates and service providers.	Implements encryption and compliance programs.	Users can access, correct, and delete data.
ALIBABA	Gathers data through user interactions and automatic collection.	Used for communication, account management, and marketing.	Shares with affiliates and third parties for services.	Employs security measures like encryption.	Users have rights to access and manage their data.
AZURE	Collects service data during cloud service usage.	Used for delivering services, improving performance, and support.	Does not sell but may share with consent or legal obligation.	Utilizes encryption for data protection.	Users can access, delete, or export their data.
ORACLE	Collects personal data from various sources including user interactions.	Used for communication, account management, and marketing purposes.	Shares internally and with third parties as needed.	Implements security measures to protect personal information.	Users can access, correct, or delete their data.
GOOGLE CLOUD	Collects personal information during service provision.	Used for service delivery, improvement, and security purposes.	Does not sell but shares under specific conditions.	Employs strong encryption and security protocols.	Users have rights to manage their data under GDPR.

Table 3.2 Summary and comparison of famous cloud providers' privacy policies

3.5.2 Insights and Trends in Cloud Provider GDPR Compliance

The analysis reveals that most cloud providers have made substantial efforts to comply with GDPR, particularly in terms of transparency and user rights. However, challenges remain in areas such as cross-border data transfers, where additional safeguards are necessary, and ensuring that consent is always obtained in a compliant manner.

3.5.3 Challenges in Evaluating Compliance

Evaluating compliance with GDPR regulations for cloud providers comes with several challenges. One major difficulty is the complexity of the GDPR itself. Some terms and requirements, such as "appropriate safeguards" or "legitimate interests," are not clearly defined, leaving room for different interpretations. Privacy policies often add to the challenge as they are lengthy and written in complicated language, making it hard to understand the exact data processing practices. Additionally, there is no standard format for these policies, so each cloud provider uses its own structure and terminology, which makes it harder to compare and analyze them.

Another challenge is the technical side of evaluating compliance. Many privacy policies do not fully explain all their data processing activities, making it difficult to identify issues like sharing data with third parties. Advanced tools, such as natural language processing (NLP), are needed to analyze these policies because they are written in free text without a consistent format. Building a dataset to train these tools is also a time-consuming task since privacy policies need to be manually labelled with the correct GDPR articles. Mistakes or inconsistencies in labelling can reduce the accuracy of the tools and make it harder to apply them to a wide range of policies. These challenges show that evaluating GDPR compliance is not a straightforward process and requires a mix of legal understanding and technical methods.

3.6 Conclusion

This chapter provided an in-depth analysis of the GDPR regulations that apply to cloud providers and examined how these regulations are reflected in their privacy policies. The research methodology, including the dataset and labelling process, was explained in detail. Insights from the analysis of privacy policies revealed trends in GDPR compliance, highlighting both areas of strength and areas requiring further improvement. The next chapter will focus on the training and evaluation of the model designed to identify GDPR compliance in privacy policies.

Chapter 4

Implementation

4.1 Introduction

In this chapter, I describe the development of an AI-based tool designed to automate the process of reviewing cloud providers' privacy policies for GDPR compliance. The increasing complexity and volume of data privacy regulations necessitate efficient tools that aid in ensuring compliance, thereby reducing potential legal risks for businesses. My tool leverages natural language processing (NLP) techniques to analyse text and extract relevant compliance-related information.

The first stage of the project involved building a custom dataset directly designed for the situation; in its most basic form, as the core of the data, the privacy policies and terms of service documents created by the most renowned cloud providers such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud; and these were supplemented by fictitious examples depicting imaginary realistic scenarios of the GDPR, thereby diversifying and enriching the dataset. The individual text snippets were

then reviewed and classified manually according to the above-mentioned categories by referring to the GDPR, such as Lawfulness, Erasure, Access, and Security Breach.

The datasets were processed using Word2Vec embeddings, which turn words into compact vector representations where the meanings and contexts of the words are captured. These embeddings were fed as input features of a supervised machine-learning classifier that was trained for labelling text snippets with the previously established GDPR labels. The training included rigorous testing of various architectures of models, hyperparameters, as well as preprocessing techniques for better accuracy in classification.

The instrument was severely assessed using various parameters, including accuracy, precision, recall, and F1 score. To achieve an unbiased evaluation, validation has been done in a portion of the dataset that was separated during experimentation. Testing has also been done for edge cases and ambiguous text to quantify the robustness and reliability of the instrument in real-world scenarios.

4.2 Tools and Libraries used

In-depth consideration was paid on the tools and technologies during the successful completion of this project. They were selected due to their high competency in handling complex text processing, machine learning jobs, and ensuring efficient model development. The tools and libraries used in conducting this research have been summed below:

Programming Language: Python

Python is the programming language that was used in the project having a vast library ecosystem as well as great adoption in the areas of Natural Language Processing and machine learning. Ease and readability of Python made it very easy to develop, try, and perfect the model implementation.

Integrated Development Environment: PYCHARM

PyCharm served as the main environment for coding and debugging. Its robust features, tailored specifically for Python development, significantly streamlined the process of constructing and testing classifiers. PyCharm's seamless integration with Anaconda allowed for efficient environment management, ensuring compatibility with required libraries and dependencies. The powerful debugging tools and intuitive interface enhanced the development workflow, while built-in version control features like Git integration simplified project management and collaboration.

Key Libraries and Frameworks:

Gensim: It was Word2Vec embedding that became a significant part of Gensim for understanding the semantic and contextual meanings of words in the dataset. These embeddings would help in making the accurate text classification of transforming the text into machine-readable numeric data.

TensorFlow: The classification model employed TensorFlow, an open-source machine learning framework, for building and training it. Its flexibility and scalability allowed for the design of a solid model capable of handling the complexities inherent in GDPR texts classification.

Scikit-learn (sklearn): It is the library that does most of the preprocessing, model evaluation, and statistical analysis. Its toolbox for precision, recall, and F1 score was above any other in assessing how much the model performs.

Pandas: They were all-purpose data manipulation, organization, and general dataset background preparation tools, allowing cleaning, structuring, and labelling of the text snippets without much hassle.

Matplotlib and Seaborn: These libraries were used for visualization and analysis of results on the data. The plots and graphs thus produced do demonstrate the performance of the model over time and datatrend in terms of the performance of the model on such datasets.

NumPy: NumPy was essential in performing numerical operations and handling multidimensional arrays, especially during embedding generation and model training.

4.3 Implementation

4.3.1 System Architecture

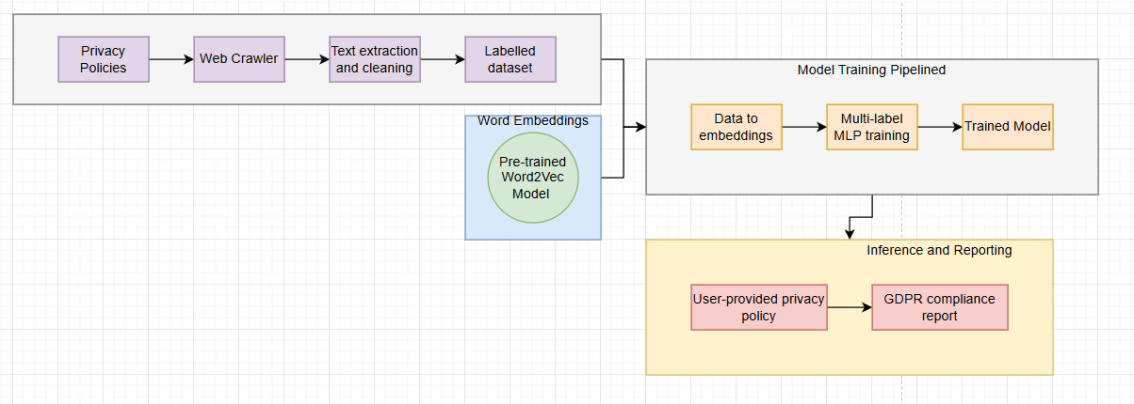


Image 4.1 System's Architecture

Data Ingestion & Preparation:

- Privacy Policies: The system begins by sourcing privacy policies from the internet.
- Web Crawler: A tool that automatically collects privacy policies from various websites.
- Text Extraction & Cleaning: Extracts and cleans the text from the collected privacy policies.
- Labeled Dataset: The cleaned text is labeled to create a dataset for training.

Model Training Pipeline:

- Data to Embeddings: Converts the labeled dataset into word embeddings using a pretrained Word2Vec model.
- Multi-Label MLP Training: Trains a multi-label Multi-Layer Perceptron (MLP) model using the word embeddings.

Inference & Reporting:

- Trained Model: Uses the trained MLP model to analyze user-provided privacy policies.
- User-provided Privacy Policy: Takes the input privacy policy provided by the user.
- GDPR Compliance Report: Outputs a report detailing the GDPR compliance of the user-provided privacy policy.

4.3.2 Word2Vec Embeddings

Converting words to vectors is the first step in the procedure of building the model. Words that are semantically similar need to be close in the vector space. For this specific process, I used a pre-trained Word2Vec model on data from Google News [22].

To test the embeddings, the following simple script was written:

```
from gensim.models import KeyedVectors

model_path = 'GoogleNews-vectors-negative300.bin'
model = KeyedVectors.load_word2vec_format(model_path, binary=True)

# Example usage
print(model.most_similar('lawyer'))  # Get similar words
print(model['contract'])  # Access vector for a word
```

Image 4.2 Testing the embeddings

This script loads the Google News embeddings in the memory and prints the most similar word to the word lawyer and the embedding of the word “contract”.

The most similar words to the word “lawyer” were the following:

[('attorney', 0.8204510807991028), ('lawyers', 0.7513089776039124), ('laywer', 0.7442193627357483), ('solicitor', 0.6864126920700073), ('attorneys', 0.6720287799835205), ('Lawyer', 0.6663058996200562), ('counsel',

0.6496455669403076), ('litigator', 0.6400834321975708), ('barrister', 0.6279128789901733), ('attorny', 0.6219167709350586)]

The confidence score of each word is also shown.

Converting words to vectors using Word2Vec was the first step in building the model. Now that these embeddings were created, I use them to train the model that predicts the GDPR compliance of every given text.

4.3.3 Constructing the Problem

A neural network classifier will work for multiple labels with many output neurons, and these neurons would be for different labels or categories. In the case of GDPR rights, there is one output neuron for "Lawfulness," the other for "Erasure," and so on. Each neuron will generate a score-from 0 to 1-that denotes the model's confidence in referring to that text with a particular label. Unlike the single-output (multi-class) scenario, a multi-label approach allows multiple categories to be activated simultaneously.

Under the hood, the function defines a sigmoid activation on each output neuron. This function forms the learned representation (logits) for each label into a sort of probability-like score. For example, if the neuron for "Lawfulness" yields 0.8, that around-and-above-this point means that the model is about 80 percent sure that it refers to a lawful basis for data processing. These outputs are not forced to sum to 1: each label is treated independently; hence, it may be possible for the text to be classified under "Lawfulness," "Right to Erasure," and "Security" all at once.

To conclude predictions, typically a threshold value (often 0.5) is applied: any label for which the confidence score is above that threshold is deemed "present" and everything below it "absent." Therefore, snippets can end up because of lacking multiple labels where confidence scores exceed that threshold. Since probabilities are handled independently for each label, the model inherently supports overlapping categories and complex policy statements reflecting the multidimensional nature of legal requirements.

4.3.4 The classifier

In this project, the multi-label classifier is a multi-layer perceptron (MLP) neural network that attempts to classify text fragments with multiple GDPR annotations. The reason for opting for such an approach is that an overlap most likely exists in the way a specific GDPR compliance category can possibly cover a specific snippet. For example, the snippet may relate to aspects of "Lawfulness" and "Erasure" in a single go. The detailed components and working of the classifier are as follows:

Neural Network Architecture

Input Layer:

The input feature part of the input layer is formed by word embeddings. Word embeddings are such dense numerical representations of the semantic and contextual relationships of words making them suitable to be run over the neural network. The number of neurons in the input layer is defined by the embedding dimension, that is, corresponding to the size of the Word2Vec vectors.

```
10  # ------------------------------
11  # 1. Load the Pretrained Word2Vec Model
12  # ------------------------------
13  print("Loading Word2Vec embeddings...")
14  w2v_model_path = "GoogleNews-vectors-negative300.bin"  # Adjust path as needed
15  w2v_model = KeyedVectors.load_word2vec_format(w2v_model_path, binary=True)
16  embedding_dim = w2v_model.vector_size
17  print(f"Word2Vec loaded. Embedding dimension = {embedding_dim}")
18
```

Image 4.3 Loading Word2Vec

```
# ------------------------------
# 4. Text -> Embeddings
# ------------------------------
1 usage  kdimit04
def sentence_to_embedding(sentence, w2v_model):
    words = sentence.split()
    word_vectors = [w2v_model[word] for word in words if word in w2v_model]
    if word_vectors:
        return np.mean(word_vectors, axis=0)
    else:
        return np.zeros(embedding_dim)


print("Generating embeddings for text...")
X = np.array([sentence_to_embedding(text, w2v_model) for text in tqdm(texts)])
```

Image 4.4 Generating embeddings

Hidden Layers:

The initial hidden layer is of 128 neurons and has ReLU (Rectified Linear Unit) as the activation function. An introduction of non-linearity has been made through this method; thus, refraining the model from merely inheriting complicated functions from input data.

A dropout layer having 0.3 drop probability during training is added to avoid overfitting and randomly deactivating 30% of the neurons.

The second hidden layer now has also 64 neurons with a ReLU activation function, as it helps to further refine the feature extraction by extracting more specific patterns.

Another dropout layer with 0.3 dropout rate is also applied for regularization.

```
# ------------------------------------
# 6. Build a Multi-Label MLP Classifier
# ------------------------------------
model = models.Sequential()
model.add(layers.Dense( units: 128, activation="relu", input_shape=(embedding_dim,)))
model.add(layers.Dropout(0.3))
model.add(layers.Dense( units: 64, activation="relu"))
model.add(layers.Dropout(0.3))
```

Image 4.5 Build Multi-Label MLP classifier

Output Layer:

The output layer contains one neuron for each GDPR category. Since this is a multi-label classification problem, the activation function used is sigmoid, which outputs a probability score for each category independently. A score closer to 1 indicates that the snippet likely belongs to that category, while a score closer to 0 suggests it does not.

```
85  # Output layer size = number of categories, activation = 'sigmoid' for multi-label
86  model.add(layers.Dense(Y.shape[1], activation="sigmoid"))
```

Image 4.6 Output Layer

At this point, the model is compiled using Adam optimizer, probably the most famous optimization algorithm, which adapts the learning rate to be smaller at the last stages of training to obtain a better convergence. With respect to the loss function, binary cross entropy is used, which is a great fit for an independent evaluation of each category for multi-label problems. Accuracy is used as the baseline metric to measure the performance; however, precision, recall, and F1 score can be calculated as well.

```
88  model.compile(
89      optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
90      loss="binary_crossentropy",  # multi-label => binary crossentropy
91      metrics=["accuracy"],
92  )
```

Image 4.7 Model's Compilation

Training Phase:

Training set model is compiled with this data into a training and validation set.
This is accomplished using:

1. The input (word embeddings) is fed to the network.
2. The network adjusts weights and biases by backpropagation to minimize the binary cross-entropy loss.
3. Use validation data to monitor mod
4. el performance and prevent overfitting through early stopping or hyper parameter fine-tuning.

```
print("Training multi-label classifier...")
history = model.fit(
    X_train,
    Y_train,
    validation_split=0.2,
    epochs=200,  # Adjust as needed
    batch_size=32,
    verbose=1,
)
```

Image 4.8 Training the classifier

Essential parameters:

1. Epochs: The model is trained for 200 epochs indicating that the network will see whole data 200 times. This can be modified based on the convergence.
2. Batch Size: The size of each train step for running the model will have 32 samples, which provides a trade-off between better model performance and computational efficiency to achieve it.

The trained model outputs a probability score for each category for a given input snippet. For example, if a text snippet pertains to both “Lawfulness” and “Transparency,” the model assigns high probability scores to these categories while keeping scores for unrelated categories low.

This neural network design ensures that the classifier can handle the complexity of GDPR text classification efficiently, providing a robust tool for automating compliance tasks.

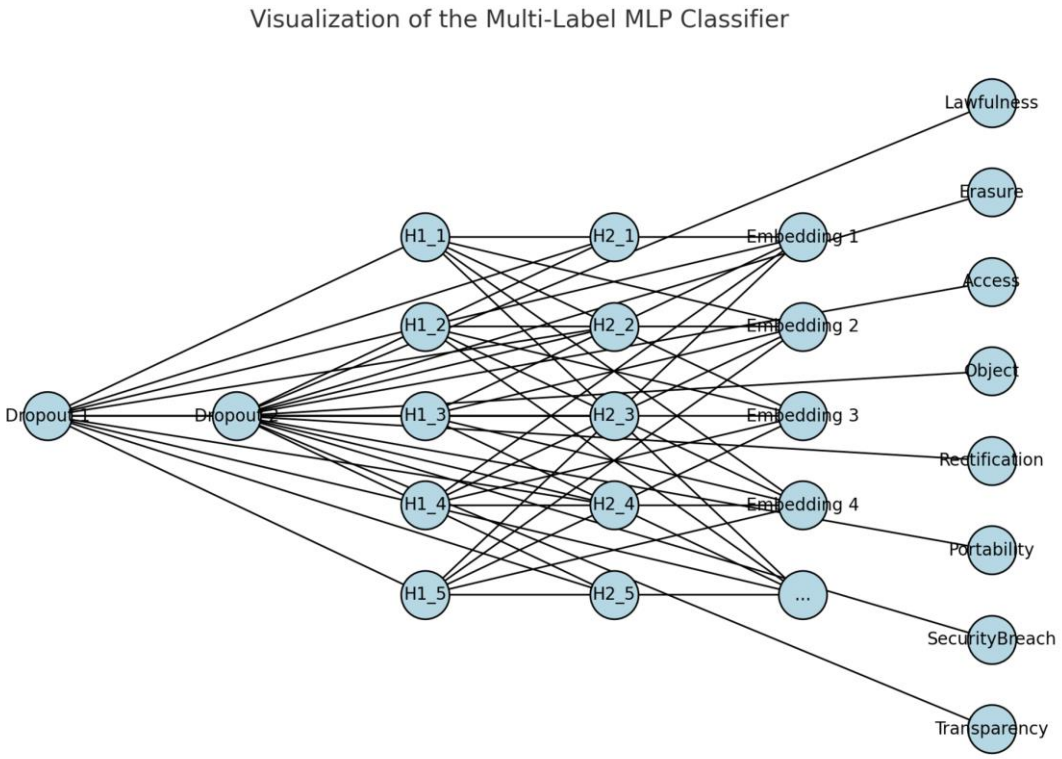


Image 4.9 A visualization of the model used.

4.3.5 The training script

The script begins by loading a pretrained Word2Vec model, which is essential for converting text into dense vector representations.

- Model Path: The path to the GoogleNews-vectors-negative300.bin file is provided, which contains embeddings trained on a large corpus of Google News articles.
- Embedding Dimension: The embeddings have a dimensionality of 300, meaning each word is represented as a 300-dimensional vector. This is a crucial input for later stages.

A custom dataset is loaded from a CSV file named the_dataset2.csv. The file contains text snippets and their corresponding one-hot encoded labels for multiple GDPR categories.

- Columns: Key columns include Text (the text snippets) and categorical labels such as "Lawfulness," "Erasure," and "Transparency."

- Output: The dataset is loaded into a pandas DataFrame for ease of manipulation.

Image 4.10 Loading the dataset

The sentence_to_embedding function generates embeddings for each text snippet:

- Tokenization: Each sentence is split into words.
- Vector Conversion: Words are mapped to their Word2Vec vectors if they exist in the vocabulary.
- Averaging: The average of all word vectors in a sentence is computed to create a single embedding representing the entire sentence.
- Fallback: If no words in the sentence are in the Word2Vec model, a zero vector is used.

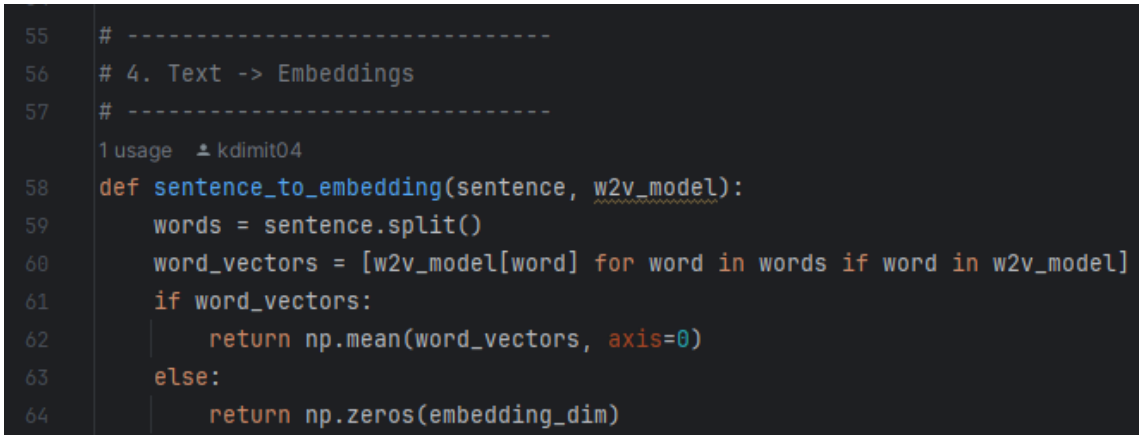


Image 4.11 sentence-to-embedding function

The data is split into training and testing sets using an 80/20 ratio:

- **Training Data**: Used for fitting the model.
- **Testing Data**: Held out for final evaluation to assess the model's performance on unseen data.

```
70  # ------------------------------
71  # 5. Train / Validation / Test Split
72  # ------------------------------
73  X_train, X_test, Y_train, Y_test = train_test_split(
74      *arrays: X, Y, test_size=0.2, random_state=42
75  )
```

Image 4.12 Training and testing

The neural network is designed using TensorFlow/Keras:

- **Input Layer**: Accepts the 300-dimensional word embeddings.
- **Hidden Layers**:
    - Layer 1: 128 neurons with ReLU activation for learning complex patterns.
    - Dropout: 30% dropout rate to prevent overfitting.
    - Layer 2: 64 neurons with ReLU activation for further feature extraction.
    - Dropout: Another 30% dropout rate for regularization.
- **Output Layer**: Contains one neuron per GDPR category with a sigmoid activation function. Each neuron independently predicts the probability of the text belonging to that category.
- **Compilation**: Uses the Adam optimizer with a learning rate of 0.001 and binary cross-entropy as the loss function (suitable for multi-label problems).

The model is trained using the training data:

- **Epochs**: Set to 200, meaning the dataset is passed through the model 200 times.
- **Batch Size**: 32 samples per training step for computational efficiency.
- **Validation Split**: 20% of the training data is used for validation to monitor performance during training.

For the model evaluation:

- **Predictions**: The model outputs probabilities for each category for the test data.
- **Thresholding**: Probabilities are converted to binary labels (1 or 0) using a threshold of 0.5.
- **Subset Accuracy**: Measures the percentage of samples for which all labels are correctly predicted.
- **Sample Predictions**: Prints a few examples of true vs. predicted labels to analyze the model's behavior.

```
111  # Calculate a simple subset accuracy: how many predictions match exactly
112  subset_accuracy = np.mean(
113      [1 if np.array_equal(Y_pred[i], Y_test[i]) else 0 for i in range(len(Y_test))]
114  )
115
116  print(f"Subset Accuracy (exact match of all labels): {subset_accuracy:.2f}")
```

Image 4.13 Subset accuracy

```
134  # Print a few example predictions
135  for i in range(min(5, len(X_test))):
136      true_labels = [label_columns[idx] for idx, val in enumerate(Y_test[i]) if val == 1]
137      pred_labels = [label_columns[idx] for idx, val in enumerate(Y_pred[i]) if val == 1]
138
139      print(f"\n=== Sample {i} ===")
140      print(f"Text: {texts[i]}")
141      print(f"True: {true_labels}")
142      print(f"Pred: {pred_labels}")
```

Image 4.14 Prediction outputs

Model: Saved as gdpr_multi_label_model.h5 for future use.

```
144  # ---------------------------------
145  # 8. Save the Model & Label Info
146  # ---------------------------------
147  model.save("gdpr_multi_label_model.h5")
148  print("Model saved as gdpr_multi_label_model.h5")
149
150  joblib.dump(label_columns, filename: "gdpr_label_columns.pkl")
151  print("Label columns saved as gdpr_label_columns.pkl")
```

Image 4.15 Model saving

The following is a snippet of the training log prints:

```
Training multi-label classifier...
Epoch 1/200
2024-12-27 18:07:41.658936: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin
optimizer for device_type GPU is enabled.
2024-12-27 18:07:41.800440: E
tensorflow/core/grappler/optimizers/meta_optimizer.cc:954] model_pruner failed:
INVALID_ARGUMENT: Graph does not contain terminal node
AssignAddVariableOp_10.
7/7 [==============================] - ETA: 0s - loss: 0.6913 - accuracy:
0.11982024-12-27 18:07:42.874793: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin
optimizer for device_type GPU is enabled.
7/7 [==============================] - 2s 102ms/step - loss: 0.6913 -
accuracy: 0.1198 - val_loss: 0.6794 - val_accuracy: 0.1636
Epoch 2/200
7/7 [==============================] - 0s 19ms/step - loss: 0.6769 -
accuracy: 0.1198 - val_loss: 0.6647 - val_accuracy: 0.2000
Epoch 3/200
7/7 [==============================] - 0s 17ms/step - loss: 0.6649 -
accuracy: 0.2166 - val_loss: 0.6544 - val_accuracy: 0.1636
Epoch 4/200
7/7 [==============================] - 0s 17ms/step - loss: 0.6518 -
accuracy: 0.1751 - val_loss: 0.6452 - val_accuracy: 0.2000
Epoch 5/200
7/7 [==============================] - 0s 15ms/step - loss: 0.6445 -
accuracy: 0.1982 - val_loss: 0.6341 - val_accuracy: 0.2364
Epoch 6/200
7/7 [==============================] - 0s 15ms/step - loss: 0.6256 -
accuracy: 0.2258 - val_loss: 0.6208 - val_accuracy: 0.2545
```

Table 4.1 Training log snippet part 1

The end of the training script is the following:

```
Epoch 199/200
7/7 [==============================] - 0s 15ms/step - loss: 0.0218 -
accuracy: 0.4009 - val_loss: 0.2487 - val_accuracy: 0.2909
Epoch 200/200
7/7 [==============================] - 0s 15ms/step - loss: 0.0259 -
accuracy: 0.3871 - val_loss: 0.2563 - val_accuracy: 0.2909
Evaluating on test set...
```

2024-12-27 18:08:04.206490: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114] Plugin optimizer for device_type GPU is enabled.
3/3 [==============================] - 0s 16ms/step
Subset Accuracy (exact match of all labels): 0.45
Majority Accuracy (>= 50% labels correct): 0.97

=== Sample 0 ===
Text: We process data lawfully. You can have it erased at any time.
True: ['Lawfulness', 'Erasure', 'Access', 'Object', 'Rectification', 'Portability', 'SecurityBreach', 'Transparency']
Pred: ['Lawfulness', 'Erasure', 'Access', 'Object', 'Rectification', 'Portability', 'SecurityBreach', 'Transparency']

=== Sample 1 ===
Text: Our system ensures you may request and download all personal information.
True: ['Erasure', 'Transparency']
Pred: ['Erasure']

=== Sample 2 ===
Text: We keep your data secure. If anything goes wrong, we alert you of a breach immediately.
True: ['Portability', 'Transparency']
Pred: ['Transparency']

=== Sample 3 ===
Text: Users can rectify their information and also object to marketing emails.
True: ['Rectification', 'Transparency']
Pred: ['Rectification', 'Transparency']

=== Sample 4 ===
Text: Data is handled with transparency. You can request access if needed.
True: ['SecurityBreach', 'Transparency']
Pred: ['SecurityBreach', 'Transparency']
Model saved as gdpr_multi_label_model.h5
Label columns saved as gdpr_label_columns.pkl

Table 4.2 Training log snippet part 2

As shown above, I printed two different accuracy metrics. The first one is the “exact match of all labels” accuracy which is 45% and the second one is the “majority accuracy” which means that we need more that 50% of correct labels to consider it correct which is 97%.

4.3.6 Inference Performance

I run some sample inference to test the model on the fly. The script, inference.py, can be found in Appendix A.

We have the following:

Sample 0:

- **Text**: "We process data lawfully. You can have it erased at any time."
- **True Labels**: [‘Lawfulness’, ‘Erasure’].
- **Predicted Labels**: [‘Lawfulness’, ‘Erasure’].
- The model perfectly predicted all labels for this sample, showing its capability to handle straightforward examples.

Sample 1:

- **Text**: "Our system ensures you may request and download all personal information."
- **True Labels**: [‘Access’, ‘Portability’]
- **Predicted Labels**: [‘Portability’]
- The model missed the “Access” label, indicating it might struggle with subtle cues in complex sentences.

Sample 2:

- **Text**: "We keep your data secure. If anything goes wrong, we alert you of a breach immediately."
- **True Labels**: [‘Security’, ‘Transparency’, ‘SecurityBreach’]
- **Predicted Labels**: [‘SecurityBreach’]
- Here, the model successfully identified "SecurityBreach" but missed ‘Security” and “Transparency”. This suggests that it may need more examples to recognize relationships between similar or overlapping categories.

Sample 3:

- **Text**: "Users can rectify their information and also object to marketing emails."
- **True Labels**: [‘Rectification’, ‘Object’, ‘Transparency’]

- **Predicted Labels**: [‘Rectification’, ‘Object’, ‘Transparency’]
- This is another correct prediction, indicating the model’s ability to recognize direct, clear statements about user rights.

Sample 4:

- **Text**: "Data is handled with transparency. You can request access if needed."
- **True Labels**: [‘Access’, ‘Transparency’]
- **Predicted Labels**: [‘Access’, ‘Transparency’]
- Another perfect prediction, showing that the model is consistent in detecting clear mentions of key GDPR categories.

The model performs well in predicting the majority of categories for most examples, as seen in the high majority accuracy (97%).
It is especially effective in straightforward cases where the language directly maps to the GDPR categories.

The model struggles with ambiguous or less explicit text, as seen in Samples 1 and 2. This suggests that additional training data with diverse sentence structures and contextual nuances could improve performance.
A 45% subset accuracy reflects the challenge of exact matches in multi-label classification, especially with overlapping and nuanced categories like GDPR.

4.3.7 Privacy Policy checker script on Cloud Providers

Here I created a tool in which the user can give a URL of a privacy policy from cloud providers, and the system decides if the policy is GDPR compatible or not.

The Graphical User Interface is clear and simple, as shown below:

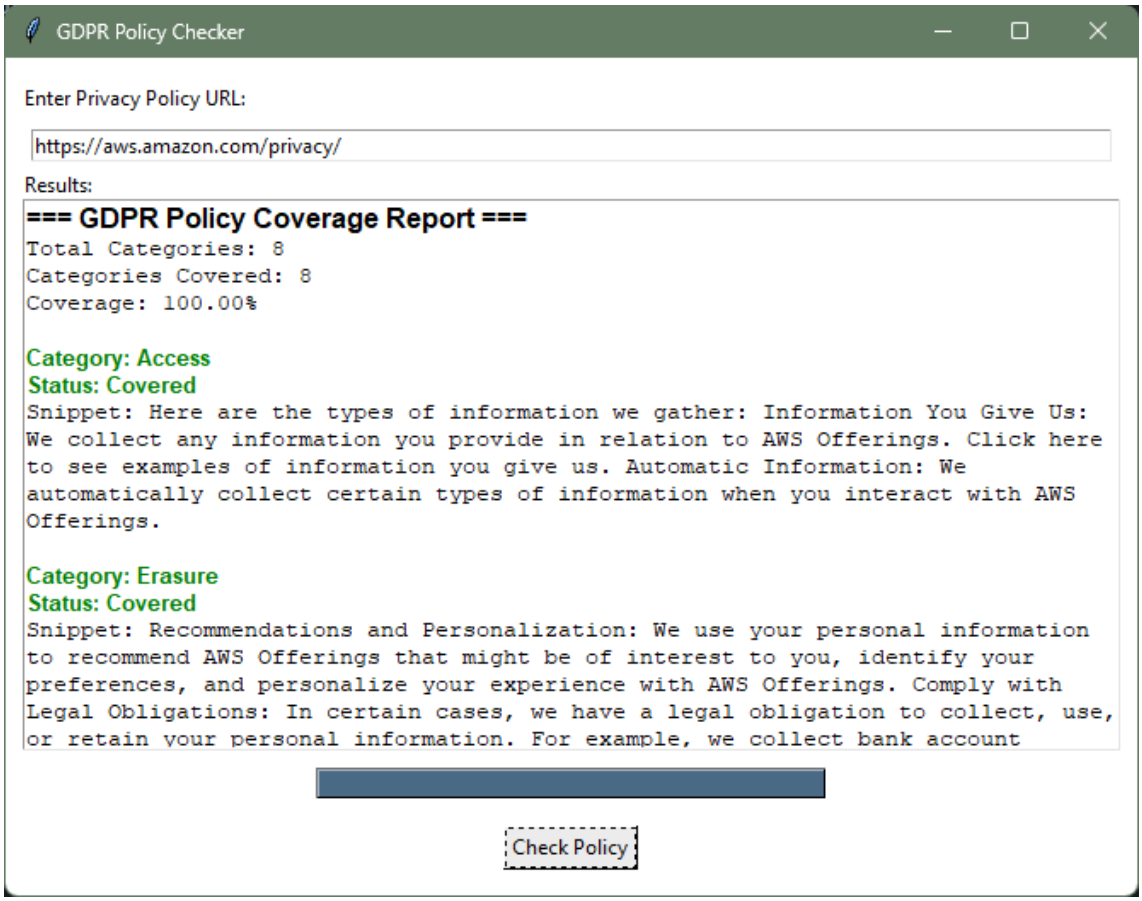


Image 4.16 Graphical User Interface

The user inserts a URL of a privacy policy, in the particular example I used AWS privacy policy. Then you click the button at the bottom of the window, “Check Policy” and you wait a few seconds until the appropriate files of the model load and process the policy. After that, the tool prints out the GDPR categories that were covered in that policy, along with a compliance score. Moreover, for each covered category there are printed snippets for justification. By scrolling down, the user can see all the categories and snippets analytically.

The script works as described in the below steps.

First, the script takes a link (URL) to a privacy policy from the user.
It fetches the web page content and extracts its main text using requests and BeautifulSoup.

```
1 usage  kdimit04
def fetch_policy_text(url, max_len=2_000_000):
    """
    Fetches the webpage content from the given URL and attempts to extract main text.
    Returns a cleaned string of text.
    """
    try:
        headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)"}
        response = requests.get(url, headers=headers, timeout=10)
        response.raise_for_status()
    except requests.exceptions.RequestException as e:
        print(f"Error fetching URL: {e}")
        return ""
```

Image 4.17 Fetching privacy policy text

Then, it splits the text into sentences and groups them in pairs (2-sentence chunks).

```
1 usage  kdimit04
def split_into_chunks(text, chunk_size=2):
    """
    Splits the text into sentences, then groups them into chunks of 'chunk_size' sentences.
    Returns a list of chunk strings.
    """
    sentences = nltk.tokenize.sent_tokenize(text)
    chunks = []
    for i in range(0, len(sentences), chunk_size):
        chunk = " ".join(sentences[i : i + chunk_size])
        chunk = chunk.strip()
        if chunk:
            chunks.append(chunk)
    return chunks
```

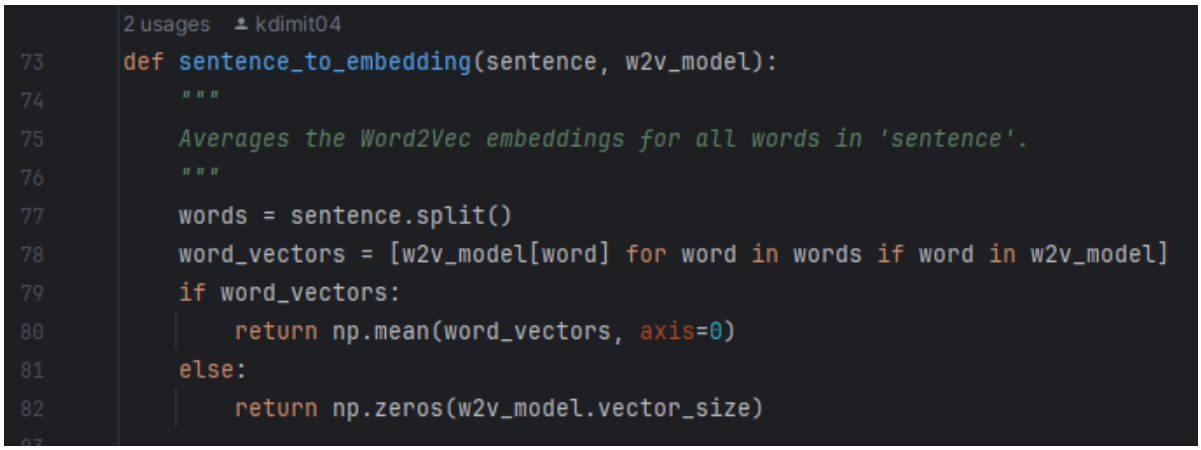
Image 4.18 Splitting text into chunks

It loads the trained multi-label GDPR model and Word2Vec embeddings.

```
1 usage  kdimit04
def load_model_and_embeddings():
    """
    Loads the trained multi-label GDPR model, label columns, and Word2Vec embeddings.
    Adjust file paths as needed.
    """
    # Load the model
    model = tf.keras.models.load_model("gdpr_multi_label_model.h5")

    # Load label columns
    label_columns = joblib.load("gdpr_label_columns.pkl")

    # Load Word2Vec
    w2v_model_path = "GoogleNews-vectors-negative300.bin"  # Adjust if needed
    w2v_model = KeyedVectors.load_word2vec_format(w2v_model_path, binary=True)

    return model, label_columns, w2v_model
```

Image 4.19 Loading the trained model and embeddings

It infers which GDPR categories are referenced in each chunk.



```
2 usages  kdimit04
def sentence_to_embedding(sentence, w2v_model):
    """
    Averages the Word2Vec embeddings for all words in 'sentence'.
    """
    words = sentence.split()
    word_vectors = [w2v_model[word] for word in words if word in w2v_model]
    if word_vectors:
        return np.mean(word_vectors, axis=0)
    else:
        return np.zeros(w2v_model.vector_size)
```

Image 4.20 Identifying possible GDPR categories in the chunks

The script tracks which categories were found in at least one chunk.

```python
def predict_chunk(chunk, model, w2v_model, label_columns, threshold=0.7):
    """
    Converts a chunk into embedding, runs it through the model,
    and returns the set of predicted labels above 'threshold'.
    """
    embedding = sentence_to_embedding(chunk, w2v_model).reshape(1, -1)
    probs = model.predict(embedding)[0]  # shape: (num_labels,)
    predicted_indices = np.where(probs >= threshold)[0]
    predicted_labels = {label_columns[i] for i in predicted_indices}
    return predicted_labels
```

Image 4.21 Tracks found GDPR categories in the chunks

Lastly, it prints the final percentage of GDPR categories that appear to be addressed, along with a list of which ones were found and also prints the relevant sentence that triggered the category.

```python

    print("\n=== GDPR Policy Coverage Report ===")
    print(f"URL: {url}")
    print(f"Total GDPR categories: {total_categories}")
    print(f"Covered categories ({covered_count}): {sorted(found_categories.keys())}")
    print(f"Final coverage: {coverage_percentage:.2f}%\n")

    # Print the snippet where each category was first detected
    print("Where each category was first detected:\n")
    for cat in sorted(found_categories.keys()):
        snippet = found_categories[cat]
        print(f"Category: {cat}")
        print(f"Snippet: {snippet}\n")
```

Image 4.22 Printing the coverage percentage and snippets

Chapter 5

Evaluation and Comparison of Results

5.1 Evaluation of the Model

I created an evaluation script (evaluate_model.py) that loads the trained multi-label model and dataset, performs inference on the test set, and computes a variety of multi-label metrics, including:

- Subset Accuracy
- Hamming Loss
- Precision, Recall, and F1-Score (per label and macro-averaged)
- Mean Average Precision (mAP)

The full code of the script evaluate_model.py can be found in Appendix A
The output of this script is the following:

```
Subset Accuracy (exact match): 0.45
Hamming Loss: 0.1123

Classification Report (per label, includes macro avg):
              precision    recall  f1-score   support

   Lawfulness       0.82      0.74      0.78       31
      Erasure       0.83      0.93      0.88       27
       Access       0.91      0.95      0.93       22
```

```
       Object       1.00      0.90      0.95        30
 Rectification       0.86      0.97      0.91        32
  Portability        0.80      0.77      0.79        31
SecurityBreach       0.90      0.84      0.87        31
  Transparency       0.81      0.90      0.85        29

    micro avg        0.86      0.87      0.87       233
    macro avg        0.87      0.88      0.87       233
 weighted avg        0.87      0.87      0.87       233
  samples avg        0.89      0.85      0.83       233

Mean Average Precision (mAP): 0.9452
  AP for Lawfulness: 0.9234
  AP for Erasure: 0.9369
  AP for Access: 0.9784
  AP for Object: 0.9950
  AP for Rectification: 0.9650
  AP for Portability: 0.9036
  AP for SecurityBreach: 0.9370
  AP for Transparency: 0.9226

Sample Predictions:
Text: We process data lawfully. You can have it erased at any time.
 True: ['Lawfulness', 'Erasure', 'Access', 'Object', 'Rectification', 'Portability',
'SecurityBreach', 'Transparency']
 Pred: ['Lawfulness', 'Erasure', 'Access', 'Object', 'Rectification', 'Portability',
'SecurityBreach', 'Transparency']

Text: Our system ensures you may request and download all personal information.
 True: ['Erasure', 'Transparency']
 Pred: ['Erasure']

Text: We keep your data secure. If anything goes wrong, we alert you of a breach
immediately.
 True: ['Portability', 'Transparency']
 Pred: ['Transparency']
```

Subset Accuracy (Exact Match):

The model achieves a 0.45 subset accuracy, meaning that it perfectly matches all true labels for around 45% of the test samples. Subset accuracy is very strict for multi-label classification: a single misclassified label (missing or extra) causes the entire sample to be counted as incorrect. Consequently, a score of 0.45 can still indicate reasonably strong performance, given that any error in one out of eight labels invalidates a match.

Hamming Loss:

With a 0.1123 Hamming loss, only about 11% of label predictions across all samples are incorrect (i.e., predicting 1 instead of 0 or vice versa). Since Hamming loss averages the label-wise errors, a relatively low value indicates the model is quite accurate on a per-label basis.

Precision, Recall, and F1-Score (Per Label & Macro):

The classification report shows each GDPR category's precision, recall, and F1-score, with values mostly in the 0.80–0.95+ range, which are quite high.

The macro average F1 is 0.87, indicating overall balanced performance across labels, and the model handles each category without letting any single label dominate or suffer too severely.

Mean Average Precision (mAP):

A high mAP of 0.9452 underscores that when considering the model's probability outputs (not just the final 0.5 threshold), it ranks the correct labels high in most cases. Label-specific AP values, such as 0.9950 for "Object", reflect that the model is extremely reliable at distinguishing text referencing that category.

Sample Predictions:

The first sample shows a complete match (all eight labels correct), highlighting the model's ability to recognize multiple overlapping GDPR rights in a single snippet.

The other two samples illustrate situations where the model misses or excludes one label (e.g., "Portability") but still captures the rest. Such partial mismatches explain why subset accuracy is moderate, even though the per-label metrics remain strong.

Overall, these results suggest the classifier is performing very well on individual GDPR labels—reflected by high per-label precision, recall, and mAP—while still having a moderate subset accuracy, which is typical in multi-label settings where exact label-set matches are challenging.

5.2 Evaluation of Cloud Providers Privacy Policies

The goal of this thesis is to create a tool that can check if the privacy policies of cloud providers follow the GDPR guidelines. To test how well the tool works, I used a range of privacy policies from different cloud providers and compared the results. The script gdpr_policy_checker.py was created as a tool to check if cloud providers privacy policies are compliant with GDPR regulations. Besides that, it can work as a testing tool for the model to review its performance and accuracy.

5.2.1 Testing the Tool on Different Privacy Policies

I tested the tool on several cloud providers' privacy policies. The tool was designed to check for compliance with different GDPR categories, like "Access," "Erasure," "Portability," and others. For each privacy policy, the tool returned that it was 100% compliant with GDPR.

When I test it with the Amazon privacy policy URL (https://aws.amazon.com/privacy/) I get the following output:

```
=== GDPR Policy Coverage Report ===
URL: https://aws.amazon.com/privacy/
Total GDPR categories: 8
Covered categories (8): ['Access', 'Erasure', 'Lawfulness', 'Object', 'Portability',
'Rectification', 'SecurityBreach', 'Transparency']
Final coverage: 100.00%

Where each category was first detected:

Category: Access
Snippet: This Privacy Notice does not apply to the “content” processed, stored, or
hosted by our customers using AWS Offerings in connection with an AWS account.
See the agreement governing your access to your AWS account and the AWS Data
Privacy FAQ for more information about how we handle content and how o...

Category: Erasure
Snippet: Comply with Legal Obligations: In certain cases, we have a legal obligation
to collect, use, or retain your personal information. For example, we collect bank
account information from AWS Marketplace sellers for identity verification.
```

Category: Lawfulness
Snippet: This Privacy Notice does not apply to the “content” processed, stored, or hosted by our customers using AWS Offerings in connection with an AWS account. See the agreement governing your access to your AWS account and the AWS Data Privacy FAQ for more information about how we handle content and how o...

Category: Object
Snippet: Automatic Information: We automatically collect certain types of information when you interact with AWS Offerings. Click here to see examples of information we collect automatically.

Category: Portability
Snippet: This Privacy Notice does not apply to the “content” processed, stored, or hosted by our customers using AWS Offerings in connection with an AWS account. See the agreement governing your access to your AWS account and the AWS Data Privacy FAQ for more information about how we handle content and how o...

Category: Rectification
Snippet: This Privacy Notice does not apply to the “content” processed, stored, or hosted by our customers using AWS Offerings in connection with an AWS account. See the agreement governing your access to your AWS account and the AWS Data Privacy FAQ for more information about how we handle content and how o...

Category: SecurityBreach
Snippet: Measure, Support, and Improve AWS Offerings: We use your personal information to measure use of, analyze performance of, fix errors in, provide support for, improve, and develop AWS Offerings. Recommendations and Personalization: We use your personal information to recommend AWS Offerings that might...

Category: Transparency
Snippet: This Privacy Notice does not apply to the “content” processed, stored, or hosted by our customers using AWS Offerings in connection with an AWS account. See the agreement governing your access to your AWS account and the AWS Data Privacy FAQ for more information about how we handle content and how o...

which means that every category was found, which makes the policy 100% GDPR compatible.

The same thing is noticed in Google’s policy which is expected.

=== GDPR Policy Coverage Report ===
URL: https://policies.google.com/privacy?hl=en-US
Total GDPR categories: 8
Covered categories (8): ['Access', 'Erasure', 'Lawfulness', 'Object', 'Portability', 'Rectification', 'SecurityBreach', 'Transparency']

Final coverage: 100.00%

Where each category was first detected:

Category: Access
Snippet: If European Union or United Kingdom data protection law applies to the processing of your information, you can review the European requirements section below to learn more about your rights and Google's compliance with these laws. Privacy Checkup Looking to change your privacy settings? Take the Privacy Checkup Effective September 16, 2024 | Archived versions | Download PDF Contents Introduction Information Google collects Why Google collects data Your privacy controls Sharing your information Keeping your information secure Exporting & deleting your information Retaining your information Compliance & cooperation with regulators European requirements About this policy Related privacy practices We build a range of services that help millions of people daily to explore and interact with the world in new ways.

Category: Erasure
Snippet: Our services include: Google apps, sites, and devices, like Search, YouTube, and Google Home Platforms like the Chrome browser and Android operating system Products that are integrated into third-party apps and sites, like ads, analytics, and embedded Google Maps You can use our services in a variety of ways to manage your privacy. For example, you can sign up for a Google Account if you want to create and manage content like emails and photos, or see more relevant search results. And you can use many Google services when you're signed out or without creating an account at all, like searching on Google or watching YouTube videos.

Category: Lawfulness
Snippet: Privacy & Terms Sign in Overview Privacy Policy Terms of Service Technologies FAQ Privacy & Terms Overview Privacy Policy Data transfer frameworks Key terms Partners Updates Terms of Service Technologies FAQ Privacy & Terms Privacy & Terms Overview Privacy Policy Terms of Service Technologies FAQ Google Account Privacy Policy Introduction Information Google collects Why Google collects data Your privacy controls Sharing your information Keeping your information secure Exporting & deleting your information Retaining your information Compliance & cooperation with regulators European requirements About this policy Related privacy practices Data transfer frameworks Key terms Partners Updates Google Privacy Policy When you use our services, you're trusting us with your information. We understand this is a big responsibility and work hard to protect your information and put you in control. This Privacy Policy is meant to help you understand what information we collect, why we collect it, and how you can update, manage, export, and delete your information.

Category: Object
Snippet: Our services include: Google apps, sites, and devices, like Search, YouTube, and Google Home Platforms like the Chrome browser and Android operating system Products that are integrated into third-party apps and sites, like ads, analytics, and embedded Google Maps You can use our services in a variety of ways to manage

your privacy. For example, you can sign up for a Google Account if you want to create and manage content like emails and photos, or see more relevant search results. And you can use many Google services when you're signed out or without creating an account at all, like searching on Google or watching YouTube videos.

Category: Portability
Snippet: You can also choose to browse the web in a private mode, like Chrome Incognito mode, which helps keep your browsing private from other people who use your device. And across our services, you can adjust your privacy settings to control whether we collect some types of data and how we use it. To help explain things as clearly as possible, we've added examples, explanatory videos, and definitions for key terms .

Category: Rectification
Snippet: You can also choose to browse the web in a private mode, like Chrome Incognito mode, which helps keep your browsing private from other people who use your device. And across our services, you can adjust your privacy settings to control whether we collect some types of data and how we use it. To help explain things as clearly as possible, we've added examples, explanatory videos, and definitions for key terms .

Category: SecurityBreach
Snippet: The information we collect includes unique identifiers , browser type and settings, device type and settings, operating system, mobile network information including carrier name and phone number, and application version number. We also collect information about the interaction of your apps, browsers, and devices with our services, including IP address , crash reports, system activity, and the date, time, and referrer URL of your request. We collect this information when a Google service on your device contacts our servers — for example, when you install an app from the Play Store or when a service checks for automatic updates.

Category: Transparency
Snippet: If European Union or United Kingdom data protection law applies to the processing of your information, you can review the European requirements section below to learn more about your rights and Google's compliance with these laws. Privacy Checkup Looking to change your privacy settings? Take the Privacy Checkup Effective September 16, 2024 | Archived versions | Download PDF Contents Introduction Information Google collects Why Google collects data Your privacy controls Sharing your information Keeping your information secure Exporting & deleting your information Retaining your information Compliance & cooperation with regulators European requirements About this policy Related privacy practices We build a range of services that help millions of people daily to explore and interact with the world in new ways.

I also tested it with less known cloud providers that mostly operates in areas outside the European Union. DigitalOcean cloud provider is an example.

=== GDPR Policy Coverage Report ===
URL: https://www.digitalocean.com/legal/privacy-policy
Total GDPR categories: 8
Covered categories (8): ['Access', 'Erasure', 'Lawfulness', 'Object', 'Portability', 'Rectification', 'SecurityBreach', 'Transparency']
Final coverage: 100.00%

Where each category was first detected:

Category: Access
Snippet: If you are looking for California-specific information, check out our CCPA Privacy Notice as well, which is incorporated into this Privacy Policy. We may collect and receive information about users of our Services from various sources, including: (i) information you provide through your user account on the Services if you register for the Services; (ii) your use of the Services; and (iii) from third-party websites, services, and partners. We recommend that you read this Privacy Policy in full, including the Additional Disclosures referenced at the bottom of this document, to ensure you are fully informed.

Category: Erasure
Snippet: If you are looking for California-specific information, check out our CCPA Privacy Notice as well, which is incorporated into this Privacy Policy. We may collect and receive information about users of our Services from various sources, including: (i) information you provide through your user account on the Services if you register for the Services; (ii) your use of the Services; and (iii) from third-party websites, services, and partners. We recommend that you read this Privacy Policy in full, including the Additional Disclosures referenced at the bottom of this document, to ensure you are fully informed.

Category: Lawfulness
Snippet: Blog Docs Get Support Contact Sales DigitalOcean Products Featured Products Droplets Scalable virtual machines Kubernetes Scalable virtual machines AI / ML Build and scale AI models Cloudways Managed cloud hosting App Platform Get apps to market faster Compute Droplets Kubernetes CPU-Optimized Droplets Functions App Platform AI / ML GPU Droplets 1-Click Models GenAI Platform Bare Metal GPUs Backups & Snapshots Backups Snapshots SnapShooter Networking Virtual Private Cloud (VPC) Cloud Firewalls Load Balancers DNS DDoS Protection Managed Databases MongoDB Kafka MySQL PostgreSQL Caching OpenSearch Storage Spaces Object Storage Volume Block Storage Developer Tools API CLI Support Plans Monitoring Uptime Identity Access Management Cloud Website Hosting Cloudways See all products Solutions AI and Machine Learning Develop, train, and deploy AI apps GPUs GenAI Platform 1-Click Models Blockchain Infrastructure for decentralized apps Blogs, Forums and Content Websites Lightning-fast, reliable CMS hosting Wordpress Ghost Mastodon Data Analytics Real-time data processing at scale Data Streaming AdTech & Martech Developer Tools DevOps and CI/CD solutions CI/CD Prototyping Digital Marketing Agencies Power your clientsâ websites and campaigns Freelancer IT Consulting Ecommerce Build beautiful online storefronts Dropshipping WooCommerce Magento Game Development Low-latency multiplayer servers Minecraft Hosting IOT Connect

to the power of the cloud ISVs Streamlined ISV application development Secure Web Hosting Powerful protection from DDoS and more Private VPN Startup Cloud Hosting Scalable, cost-effective infrastructure Small Business Video Streaming High-bandwidth, low-latency delivery Web and Mobile Apps Simple cross-platform app hosting cPanel Docker Next.js Node.js Website Hosting Fast page loads and reliable site uptime VPS Hosting Virtual Machines Questions? Talk to an expert See all solutions Developers Our Community Community Home DevOps and development guides CSS-Tricks All things web design The Wave Content to level up your business. Resources Tutorials Questions and Answers Marketplace Tools Write for DOnations Cloud Chats Customer Stories DigitalOcean Blog Pricing Calculator Get Involved Hatch Startup Program Open Source Sponsorships Hacktoberfest Deploy 2025 DO Impact Nonprofits Wavemakers Program Documentation Quickstart Compute Storage Managed Databases Containers Billing API Reference Partners DigitalOcean Partner Programs Become a Partner Partner Services Program ISV Partner Program Marketplace Hatch Partner Program Connect with a Partner Partner Programs Resources Customer Stories DigitalOcean Onboarding Series Training for Agencies and Freelancers Price Estimate Calculator Featured Partner Articles Cloud cost optimization best practices Read more How to choose a cloud provider Read more DigitalOcean vs. AWS Lightsail: Which Cloud Platform is Right for You?

Category: Object
Snippet: IV. Communication Information. If you contact us directly, we may receive additional information about you such as your name, email address, phone number, the contents of the message and/or attachments you may send us, and any other information you may choose to provide.

Category: Portability
Snippet: If you are looking for California-specific information, check out our CCPA Privacy Notice as well, which is incorporated into this Privacy Policy. We may collect and receive information about users of our Services from various sources, including: (i) information you provide through your user account on the Services if you register for the Services; (ii) your use of the Services; and (iii) from third-party websites, services, and partners. We recommend that you read this Privacy Policy in full, including the Additional Disclosures referenced at the bottom of this document, to ensure you are fully informed.

Category: Rectification
Snippet: If you are looking for California-specific information, check out our CCPA Privacy Notice as well, which is incorporated into this Privacy Policy. We may collect and receive information about users of our Services from various sources, including: (i) information you provide through your user account on the Services if you register for the Services; (ii) your use of the Services; and (iii) from third-party websites, services, and partners. We recommend that you read this Privacy Policy in full, including the Additional Disclosures referenced at the bottom of this document, to ensure you are fully informed.

Category: SecurityBreach
Snippet: If you are looking for California-specific information, check out our CCPA Privacy Notice as well, which is incorporated into this Privacy Policy. We may

collect and receive information about users of our Services from various sources, including: (i) information you provide through your user account on the Services if you register for the Services; (ii) your use of the Services; and (iii) from third-party websites, services, and partners. We recommend that you read this Privacy Policy in full, including the Additional Disclosures referenced at the bottom of this document, to ensure you are fully informed.

Category: Transparency
Snippet: If you are looking for California-specific information, check out our CCPA Privacy Notice as well, which is incorporated into this Privacy Policy. We may collect and receive information about users of our from various sources, including: (i) information you provide through your user account on the Services if you register for the Services; (ii) your use of the Services; and (iii) from third-party websites, services, and partners. We recommend that you read this Privacy Policy in full, including the Additional Disclosures referenced at the bottom of this document, to ensure you are fully informed.

As the GDPR regulation was forced to be applied in the European Union since 2018, it is not unexpected that famous cloud providers are 100% compliant with GDPR. One of the challenges I faced during the testing was that most cloud providers have already updated their policies to meet GDPR standards. This made it difficult to find examples where a policy is clearly non-compliant with GDPR. Because of this, the results of my tests may not fully reflect the tool's ability to identify non-compliance. While the tool showed 100% compliance for all the tested policies, this does not necessarily mean the tool is perfect. It may be that many cloud providers have already made their privacy policies fully compliant with GDPR. It is also possible that the tool has limitations in identifying complex or subtle compliance issues, especially in areas that are not explicitly covered in the policies or are open to interpretation.

5.3 Conclusion

Both the evaluation of the cloud providers' privacy policies and the model’s performance reveal strong capabilities in identifying GDPR compliance, but also suggest areas for further refinement. The tool's high accuracy in labelling cloud provider policies as fully compliant may be a result of its strict thresholds, which might not account for subtleties in policy text. On the other hand, the model’s evaluation metrics

show strong precision, recall, and mAP scores, indicating solid performance across the different GDPR categories.

While the model performs well on individual labels and achieves decent subset accuracy, there is room to improve in terms of ensuring more complete matches in multi-label predictions. Further tuning of the tool's thresholds and model adjustments could help address these challenges and refine the tool's effectiveness in real-world applications.

Chapter 6

Conclusions and Future Projects

6.1 Introduction

This would be the concluding chapter, which synthesizes the main contributions of the thesis in terms of developing an AI-based GDPR compliance tool and assessing the successes and challenges encountered while doing so. This chapter sums up the research findings, examines implications for practice, and outlines future avenues for exploration and improvement based on objectives that include building a model for analysing privacy policies for GDPR compliance.

6.2 Conclusions

6.2.1 Effectiveness of the GDPR Compliance Tool

The objective of this study was to develop and implement a tool that measures compliance with the GDPR within the privacy policies of online cloud providers. The multi-label classification model demonstrated strong performance in detecting relevant GDPR categories, such as the Right to Erasure, Right of Access, and Data Portability. While the overall subset accuracy is strict, the model's high precision and recall across individual categories indicate its effectiveness in flagging relevant GDPR components. These results underscore the potential of NLP-based techniques to assist businesses and data protection officers in navigating complex privacy documents, although there remains room for improvement in fully matching all labels per document.

6.2.2 Methodological Achievements

The methodology of the project included various activities: data collection by means of a customized crawler, data cleaning and preprocessing, model training with Word2Vec embeddings, and integration into a multi-label classification architecture. Unstructured text handling, dataset balancing, and hyperparameter tuning issues were solved in every phase. Iterative refining of the data pipeline and the classification model has been instrumental in achieving the performance levels discussed above. Furthermore, the development of a pipeline for inference that can parse real-world privacy policies proves the applicability of the approach in practice.

6.2.3 Practical Implications

From a practical standpoint, the GDPR Compliance Tool provides a scalable means to review privacy policies, reducing manual effort and enabling more frequent checks. By highlighting specific GDPR categories within text snippets, it can serve as a preliminary diagnostic for privacy teams, business stakeholders, and legal consultants. The modular design also makes it adaptable for organizations with different compliance needs,

potentially shortening the cycle time between drafting policies and verifying their compliance status.

6.2.4 Limitations and Challenges

Despite encouraging outcomes, several limitations remain. First, the dataset's representativeness was constrained by the availability and formatting of publicly accessible privacy policies. The model may over-predict certain categories or underperform on policy texts containing unique or untested phrasings. Additionally, legal compliance is nuanced, and automated tools can only provide indications rather than definitive legal judgments. Lastly, the system's reliance on domain-specific language models highlights a need for continuous updates to maintain accuracy as regulations evolve.

6.3 Future Projects

6.3.1 Enhancing Model Accuracy

Model accuracy improvements Advances to the tool's predictive capabilities constitute a promising subject of research in the future. The integration of advanced NLP architectures such as transformer-based models trained on specialized corpora of the legal domain could lead to even better detection accuracy. A more extensive and diverse training data corpus, composed of multilingual texts and semi-supervised learning techniques, may also yield a more powerful performance.

6.3.2 Broadening the Scope

Its applicability goes beyond the present GDPR-centric structure. Future designs could modify existing models and datasets to suit laws such as CCPA or LGPD for such a consolidated compliance checking tool to work across different jurisdictions. This characteristic would greatly contribute to its relevance in a global compliance setting.

6.3.3 Real-time Compliance Monitoring

A further direction concerns real-time scanning and monitoring capabilities. The tool could give practically real-time alerts on compliance by continuously analyzing updates to any privacy policy or crawling a particular group of websites. This proactive reaction takes care of quickly informing companies and their legal counsels regarding possible misalignments with the latest legislation.

6.3.4 Integration with Legal Frameworks

Joint ventures with legal people will enable the fine-tuning of this tool output into more binding evaluations. Collaborations with law firms or regulatory bodies may even result in permission to use authoritative guidelines and rule-based checks that would complement the model's predictions. Such integrations may pave the way for a consistently more holistic compliance advisory solution by embedding a legal rationale behind the flagged issues.

6.3.5 Commercialization Potential

Finally, there is scope to convert this prototype into a marketable product. Future endeavours might explore subscription-based or API-driven business models, conduct market analyses for privacy compliance software, and investigate licensing or partnership opportunities. Successful commercialization would require not only technical refinements but also strategic considerations around branding, user experience, and regulatory cooperation.

In sum, the current study lays a strong foundation for AI-driven GDPR compliance checks. Through targeted enhancements, broader regulatory coverage, real-time capabilities, and strategic partnerships, the tool could evolve into a comprehensive and valuable solution in the sphere of digital privacy governance.

Bibliography

[1]. United Nations, "Universal Declaration of Human Rights," 1948, https://www.un.org/en/about-us/universal-declaration-of-human-rights.

[2]. Diamond, Jared, "The Worst Mistake in the History of the Human Race," JSTOR, 1987, https://www.jstor.org/stable/2265077.

[3]. Wang, X., Wu, Y. C., Zhou, M., & Fu, H., "Beyond Surveillance: Privacy, Ethics, and Regulations in Face Recognition Technology," Frontiers in Big Data, 2024 https://pmc.ncbi.nlm.nih.gov/articles/PMC11256005/

[4]. GDPR.eu, "What is GDPR?", 2024, https://gdpr.eu/what-is-gdpr/.

[5]. G. M. O'Hara, "A legal perspective on the General Data Protection Regulation," *Elsevier*, 2008, https://www.sciencedirect.com/science/article/abs/pii/S0167739X08001957.

[6]. H. S. van der Sloot, "The General Data Protection Regulation: A Legal Overview," *European Data Protection Law Review*, Lexxion, 2016, https://edpl.lexxion.eu/article/edpl/2016/3/4.

[7]. BigID, "Navigating Cross-Border Data Transfers," 2024, https://bigid.com/blog/navigating-cross-border-data transfers/?utm_source=chatgpt.com.

[8]. SmartLaw Hub, "Automated GDPR Compliance Checking," GitHub, 2024, https://github.com/smartlawhub/Automated-GDPR-Compliance-Checking

[9]. IBM, "AI Privacy Toolkit," GitHub, 2024, https://github.com/IBM/ai-privacy-toolkit

[10]. Centraleyes, 2024, https://www.centraleyes.com/

[11]. "Compliance.ai: Simplifying Regulatory Compliance Management", 2024, https://www.compliance.ai/.

[12]. Audit, Compliance, & Risk Management Software | AuditBoard

[13]. Art. 5 GDPR – Principles relating to processing of personal data - General Data Protection Regulation (GDPR)

[14]. Art. 6 GDPR – Lawfulness of processing - General Data Protection Regulation (GDPR)

[15]. Art. 7 GDPR – Conditions for consent - General Data Protection Regulation (GDPR)

[16]. Art. 15 GDPR – Right of access by the data subject - General Data Protection Regulation (GDPR)

[17]. Art. 16 GDPR – Right to rectification - General Data Protection Regulation (GDPR)

[18]. Art. 17 GDPR – Right to erasure (‘right to be forgotten’) - General Data Protection Regulation (GDPR)

[19]. Art. 18 GDPR – Right to restriction of processing - General Data Protection Regulation (GDPR)

[20]. Art. 20 GDPR – Right to data portability - General Data Protection Regulation (GDPR)

[21]. Art. 21 GDPR – Right to object - General Data Protection Regulation (GDPR)

[22]. GoogleNews-vectors-negative300 (word2vec)

APPENDIX A

Model Training Script (train_model4.py)

```
import numpy as np
import pandas as pd
from tqdm import tqdm
from sklearn.model_selection import train_test_split
import tensorflow as tf
from keras import layers, models
from gensim.models import KeyedVectors
import joblib

# ------------------------------
# 1. Load the Pretrained Word2Vec Model
# ------------------------------
print("Loading Word2Vec embeddings...")
w2v_model_path = "GoogleNews-vectors-negative300.bin"  # Adjust path
as needed
w2v_model = KeyedVectors.load_word2vec_format(w2v_model_path,
binary=True)
embedding_dim = w2v_model.vector_size
print(f"Word2Vec loaded. Embedding dimension = {embedding_dim}")


# ------------------------------
# 2. Load Your One-Hot Dataset
# ------------------------------
def load_one_hot_dataset(filepath="the_dataset2.csv"):
    """
    Expects a CSV with columns like:

SnippetID,Text,Lawfulness,Erasure,Access,Object,Rectification,Portabil
ity,SecurityBreach,Transparency
    """
    df = pd.read_csv(filepath)
    return df


print("Loading dataset with one-hot labels...")
data = load_one_hot_dataset("the_dataset2.csv")
print(f"Loaded {len(data)} rows.")

# Define the label columns explicitly
label_columns = [
    "Lawfulness",
    "Erasure",
    "Access",
    "Object",
    "Rectification",
    "Portability",
    "SecurityBreach",
    "Transparency",
]

# ------------------------------
```

```
# 3. Extract Text and Labels
# ------------------------------
texts = data["Text"].tolist()
Y = data[label_columns].values  # shape: (num_samples, 8)


# ------------------------------
# 4. Text -> Embeddings
# ------------------------------
def sentence_to_embedding(sentence, w2v_model):
    words = sentence.split()
    word_vectors = [w2v_model[word] for word in words if word in
w2v_model]
    if word_vectors:
        return np.mean(word_vectors, axis=0)
    else:
        return np.zeros(embedding_dim)


print("Generating embeddings for text...")
X = np.array([sentence_to_embedding(text, w2v_model) for text in
tqdm(texts)])

# ------------------------------
# 5. Train / Validation / Test Split
# ------------------------------
X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, random_state=42
)

# --------------------------------------
# 6. Build a Multi-Label MLP Classifier
# --------------------------------------
model = models.Sequential()
model.add(layers.Dense(128, activation="relu",
input_shape=(embedding_dim,)))
model.add(layers.Dropout(0.3))
model.add(layers.Dense(64, activation="relu"))
model.add(layers.Dropout(0.3))
# Output layer size = number of categories, activation = 'sigmoid' for
multi-label
model.add(layers.Dense(Y.shape[1], activation="sigmoid"))

model.compile(
    optimizer=tf.keras.optimizers.Adam(learning_rate=0.001),
    loss="binary_crossentropy",  # multi-label => binary crossentropy
    metrics=["accuracy"],
)

print("Training multi-label classifier...")
history = model.fit(
    X_train,
    Y_train,
    validation_split=0.2,
    epochs=200,  # Adjust as needed
    batch_size=32,
    verbose=1,
)

# ---------------------------------
```

```
# 7. Evaluate the Model
# ----------------------------------
print("Evaluating on test set...")
Y_pred_probs = model.predict(X_test)  # shape: (num_samples,
num_categories)
Y_pred = (Y_pred_probs >= 0.5).astype(int)

# Calculate a simple subset accuracy: how many predictions match
exactly
subset_accuracy = np.mean(
    [1 if np.array_equal(Y_pred[i], Y_test[i]) else 0 for i in
range(len(Y_test))]
)

print(f"Subset Accuracy (exact match of all labels):
{subset_accuracy:.2f}")


# Calculate the accuracy of having more or equal than 50% of labels
correct
def calculate_majority_accuracy(true_labels, predicted_labels):
    correct_counts = [
        sum(1 for t, p in zip(true, pred) if t == p)
        for true, pred in zip(true_labels, predicted_labels)
    ]
    majority_correct = sum(
        1 for count in correct_counts if count >= len(true_labels[0])
/ 2
    )
    return majority_correct / len(true_labels)


majority_accuracy = calculate_majority_accuracy(Y_test, Y_pred)
print(f"Majority Accuracy (>= 50% labels correct):
{majority_accuracy:.2f}")

# Print a few example predictions
for i in range(min(5, len(X_test))):
    true_labels = [label_columns[idx] for idx, val in
enumerate(Y_test[i]) if val == 1]
    pred_labels = [label_columns[idx] for idx, val in
enumerate(Y_pred[i]) if val == 1]

    print(f"\n=== Sample {i} ===")
    print(f"Text: {texts[i]}")
    print(f"True: {true_labels}")
    print(f"Pred: {pred_labels}")

# ----------------------------------
# 8. Save the Model & Label Info
# ----------------------------------
model.save("gdpr_multi_label_model.h5")
print("Model saved as gdpr_multi_label_model.h5")

joblib.dump(label_columns, "gdpr_label_columns.pkl")
print("Label columns saved as gdpr_label_columns.pkl")
```

Evaluation script (evaluate_model.py)

```
import os
import numpy as np
import pandas as pd
from tqdm import tqdm
from sklearn.model_selection import train_test_split
from sklearn.metrics import (
    accuracy_score,
    hamming_loss,
    classification_report,
    average_precision_score,
)
import tensorflow as tf
from gensim.models import KeyedVectors
import joblib

# ------------------------------
# 1. Load Pretrained Word2Vec
# ------------------------------
print("Loading Word2Vec embeddings...")
w2v_model_path = "GoogleNews-vectors-negative300.bin"  # Update path
if needed
w2v_model = KeyedVectors.load_word2vec_format(w2v_model_path,
binary=True)
embedding_dim = w2v_model.vector_size
print(f"Word2Vec loaded (dim={embedding_dim}).")


# ------------------------------
# 2. Load One-Hot Dataset
# ------------------------------
def load_dataset(filepath="the_dataset2.csv"):
    """
    Same dataset used during training:
    Columns: SnippetID,Text,Lawfulness,Erasure,Access,Object,...
    """
    df = pd.read_csv(filepath)
    return df


print("Loading dataset for evaluation...")
data = load_dataset("the_dataset2.csv")
label_columns = [
    "Lawfulness",
    "Erasure",
    "Access",
    "Object",
    "Rectification",
    "Portability",
    "SecurityBreach",
    "Transparency",
]
Y = data[label_columns].values  # shape: (num_samples, num_labels)
texts = data["Text"].tolist()


# ------------------------------
# 3. Convert Text -> Embeddings
# ------------------------------
```

```
def sentence_to_embedding(sentence, w2v_model):
    words = sentence.split()
    word_vectors = [w2v_model[word] for word in words if word in
w2v_model]
    if word_vectors:
        return np.mean(word_vectors, axis=0)
    else:
        return np.zeros(embedding_dim)


print("Generating embeddings for the full dataset...")
X = np.array([sentence_to_embedding(text, w2v_model) for text in
tqdm(texts)])

# --------------------------------
# 4. Recreate Train/Test Split
# --------------------------------
# IMPORTANT: Must use the same random_state=42 and test_size=0.2 as in
training
X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, random_state=42
)

print(f"Data split: Train={len(X_train)}, Test={len(X_test)}")

# --------------------------------
# 5. Load the Model & Label Info
# --------------------------------
print("Loading trained multi-label model...")
model = tf.keras.models.load_model("gdpr_multi_label_model.h5")

# If you saved label columns, you can load them, but here we already
have them:
# label_columns = joblib.load("gdpr_label_columns.pkl")

# --------------------------------
# 6. Inference on Test Set
# --------------------------------
print("Running inference on test set...")
Y_pred_probs = model.predict(X_test)  # shape: (num_samples,
num_labels)
Y_pred = (Y_pred_probs >= 0.5).astype(int)

# --------------------------------
# 7. Compute Evaluation Metrics
# --------------------------------


### (A) Subset Accuracy
def subset_accuracy(y_true, y_pred):
    return np.mean(
        [1 if np.array_equal(y_pred[i], y_true[i]) else 0 for i in
range(len(y_true))]
    )


subset_acc = subset_accuracy(Y_test, Y_pred)
print(f"\nSubset Accuracy (exact match): {subset_acc:.2f}")

### (B) Hamming Loss
```

```
# Average fraction of wrong labels (0/1) across all classes and
samples
hl = hamming_loss(Y_test, Y_pred)
print(f"Hamming Loss: {hl:.4f}")

### (C) Precision, Recall, F1 per label & Macro Averages

report = classification_report(
    Y_test, Y_pred, target_names=label_columns, zero_division=0
)
print("\nClassification Report (per label, includes macro avg):")
print(report)

### (D) Mean Average Precision (mAP)
# average_precision_score in sklearn can compute AP per label, then we
average.
# 'average=None' -> returns array of AP per label. Then we take the
mean.
label_aps = []
for i in range(Y_test.shape[1]):
    ap = average_precision_score(Y_test[:, i], Y_pred_probs[:, i])
    label_aps.append(ap)

mAP = np.mean(label_aps)
print("Mean Average Precision (mAP): {:.4f}".format(mAP))

# Print label-wise AP if desired
for label, ap in zip(label_columns, label_aps):
    print(f"  AP for {label}: {ap:.4f}")

# ------------------------------
# 8. Print Sample Predictions
# ------------------------------
print("\nSample Predictions:")
for i in range(3):  # just show 3 examples
    true_labels = [label_columns[j] for j in np.where(Y_test[i] ==
1)[0]]
    pred_labels = [label_columns[j] for j in np.where(Y_pred[i] ==
1)[0]]
    print(f"Text: {data.iloc[i]['Text']}")
    print(f" True: {true_labels}")
    print(f" Pred: {pred_labels}\n")
```

Inference script (inference.py)

```
import numpy as np
import joblib
import tensorflow as tf
from gensim.models import KeyedVectors

# 1. Load the Word2Vec model
print("Loading Word2Vec embeddings...")
w2v_model_path = "GoogleNews-vectors-negative300.bin"  # Adjust if
needed
w2v_model = KeyedVectors.load_word2vec_format(w2v_model_path,
binary=True)
embedding_dim = w2v_model.vector_size
print(f"Word2Vec loaded. Embedding dimension = {embedding_dim}")
```

```
# 2. Load the trained multi-label model and label columns
print("Loading multi-label model...")
model = tf.keras.models.load_model("gdpr_multi_label_model.h5")

print("Loading label columns...")
label_columns = joblib.load(
    "gdpr_label_columns.pkl"
)  # e.g., ["Lawfulness", "Erasure", ...]
num_labels = len(label_columns)
print("Label columns:", label_columns)


def sentence_to_embedding(sentence, w2v_model):
    """Convert a sentence into an averaged Word2Vec embedding."""
    words = sentence.split()
    word_vectors = [w2v_model[word] for word in words if word in
w2v_model]
    if word_vectors:
        return np.mean(word_vectors, axis=0)
    else:
        return np.zeros(embedding_dim)


def predict_gdpr_labels(text, threshold=0.5):
    """
    Given input text:
      1) Convert to embedding
      2) Pass through model
      3) Return label probabilities + predicted labels above
'threshold'
    """
    embedding = sentence_to_embedding(text, w2v_model).reshape(1, -1)
    probs = model.predict(embedding)[0]  # shape: (num_labels,)

    # Which labels are predicted above threshold?
    predicted_indices = np.where(probs >= threshold)[0]
    predicted_labels = [label_columns[i] for i in predicted_indices]

    return probs, predicted_labels


if __name__ == "__main__":
    test_text = (
        "Portability is guaranteed. You can object to certain forms of
data usage too"
    )
    print("\n=== Test Inference ===\n")
    print("Text:", test_text)

    # Run inference
    threshold = 0.5
    probabilities, labels = predict_gdpr_labels(test_text,
threshold=threshold)

    # Print each label's probability
    print("\nProbabilities for each label (in %):")
    for label, p in zip(label_columns, probabilities):
        print(f"  {label}: {p*100:.2f}%")

    # Print predicted labels above threshold
```

```
    print(f"\nPredicted labels (threshold={threshold}):")
    if labels:
        for lbl in labels:
            print("  -", lbl)
    else:
        print("  None")
```

Script to use the overall tool to check GDPR Compliance (gdpr_policy_checker.py)

```
import tkinter as tk
from tkinter import ttk, messagebox
import threading
import requests
from bs4 import BeautifulSoup
import re
import nltk
import numpy as np
import joblib
import tensorflow as tf
from gensim.models import KeyedVectors

def fetch_policy_text(url, max_len=2_000_000):
    """
    Fetches the webpage content from the given URL and attempts to
extract main text.
    Returns a cleaned string of text.
    """
    try:
        headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64;
x64)"}
        response = requests.get(url, headers=headers, timeout=10)
        response.raise_for_status()
    except requests.exceptions.RequestException as e:
        return f"Error fetching URL: {e}"

    soup = BeautifulSoup(response.text, "html.parser")

    body = soup.find("body")
    if not body:
        return ""

    text = body.get_text(separator=" ")
    text = text[:max_len]
    text = re.sub(r"\s+", " ", text)
    return text.strip()


def split_into_chunks(text, chunk_size=3):
    """
    Splits the text into sentences, then groups them into chunks of
'chunk_size' sentences.
    Returns a list of chunk strings.
    """
    sentences = nltk.tokenize.sent_tokenize(text)
    chunks = []
    for i in range(0, len(sentences), chunk_size):
        chunk = " ".join(sentences[i : i + chunk_size])
        chunk = chunk.strip()
        if chunk:
```

```
            chunks.append(chunk)
    return chunks


def sentence_to_embedding(sentence, w2v_model):
    """
    Averages the Word2Vec embeddings for all words in 'sentence'.
    """
    words = sentence.split()
    word_vectors = [w2v_model[word] for word in words if word in
w2v_model]
    if word_vectors:
        return np.mean(word_vectors, axis=0)
    else:
        return np.zeros(w2v_model.vector_size)


def run_inference(url, results_text_widget, progress_var):
    """
    The main logic: loads the model, processes the URL, and updates
the UI with results.
    Runs in a separate thread to avoid blocking the main GUI.
    """
    try:
        # Update progress to show "loading"
        progress_var.set(20)

        # Load model
        model =
tf.keras.models.load_model("gdpr_multi_label_model.h5")

        # Load label columns
        label_columns = joblib.load("gdpr_label_columns.pkl")

        # Load Word2Vec
        w2v_model_path = "GoogleNews-vectors-negative300.bin"
        w2v_model = KeyedVectors.load_word2vec_format(w2v_model_path,
binary=True)

        # Update progress
        progress_var.set(40)

        # Fetch policy text
        policy_text = fetch_policy_text(url)
        if not policy_text or "Error fetching" in policy_text:
            results_text_widget.insert(
                tk.END, f"Failed to extract text from
policy.\n\n{policy_text}\n"
            )
            progress_var.set(0)
            return

        chunks = split_into_chunks(policy_text, chunk_size=3)

        # Update progress
        progress_var.set(60)

        found_categories = {}
        threshold = 0.7  # More stringent threshold
```

```
        for chunk in chunks:
            if len(chunk.split()) < 20:
                continue
            embedding = sentence_to_embedding(chunk,
w2v_model).reshape(1, -1)
            probs = model.predict(embedding)[0]
            pred_indices = np.where(probs >= threshold)[0]

            for idx in pred_indices:
                cat = label_columns[idx]
                # Store the first snippet that triggered this category
                if cat not in found_categories:
                    found_categories[cat] = chunk

        # Calculate coverage
        total_categories = len(label_columns)
        covered_count = len(found_categories)
        coverage_percentage = (covered_count / total_categories) * 100

        # Update progress to near-completion
        progress_var.set(80)

        # Build and display the report
        build_report(results_text_widget, found_categories,
label_columns, coverage_percentage)

        # Finish progress
        progress_var.set(100)

    except Exception as e:
        results_text_widget.insert(tk.END, f"An error occurred:\n{e}")
    finally:
        pass  # Could reset progress here or leave at 100


def build_report(results_text_widget, found_categories, label_columns,
coverage_percentage):
    """
    Builds and formats the report to be inserted into the text widget.
    This version formats categories with color coding and a clean
structure.
    """
    results_text_widget.delete("1.0", tk.END)  # Clear previous
results

    # Summary
    results_text_widget.insert(tk.END, "=== GDPR Policy Coverage
Report ===\n", "bold")
    results_text_widget.insert(tk.END, f"Total Categories:
{len(label_columns)}\n")
    results_text_widget.insert(tk.END, f"Categories Covered:
{len(found_categories)}\n")
    results_text_widget.insert(tk.END, f"Coverage:
{coverage_percentage:.2f}%\n\n")

    # Detailed category info
    for cat in sorted(found_categories.keys()):
        snippet = found_categories[cat]
        status = "Covered" if found_categories[cat] else "Not Covered"
        color_tag = "covered" if status == "Covered" else "error"
```

```
        # Insert category
        results_text_widget.insert(tk.END, f"Category: {cat}\n",
color_tag)
        results_text_widget.insert(tk.END, f"Status: {status}\n",
color_tag)
        results_text_widget.insert(tk.END, f"Snippet: {snippet}\n\n")


def start_inference(url_entry, results_text_widget, progress_var):
    """
    Called when user clicks 'Check Policy' button.
    Spawns a thread to run inference so the GUI remains responsive.
    """
    url = url_entry.get().strip()
    if not url:
        messagebox.showwarning("No URL", "Please enter a valid URL.")
        return

    # Clear previous results
    results_text_widget.delete("1.0", tk.END)

    # Start background thread
    thread = threading.Thread(
        target=run_inference, args=(url, results_text_widget,
progress_var), daemon=True
    )
    thread.start()


def create_gui():
    root = tk.Tk()
    root.title("GDPR Policy Checker")

    style = ttk.Style(root)
    style.theme_use("default")
    style.configure(".", foreground="black", background="white")
    style.configure("TLabel", foreground="black", background="white")
    style.configure("TButton", foreground="black", background="white")
    style.configure("TEntry", foreground="black",
fieldbackground="white")

    main_frame = ttk.Frame(root, padding="10 10 10 10")
    main_frame.grid(row=0, column=0, sticky=(tk.N, tk.S, tk.E, tk.W))

    url_label = ttk.Label(
        main_frame, text="Enter Privacy Policy URL:",
foreground="black"
    )
    url_label.grid(row=0, column=0, sticky=tk.W, pady=5)

    url_entry = ttk.Entry(main_frame, width=60)
    url_entry.grid(row=1, column=0, sticky=(tk.W, tk.E), padx=5,
pady=5)

    results_label = ttk.Label(main_frame, text="Results:",
foreground="black")
    results_label.grid(row=2, column=0, sticky=tk.W)

    results_text = tk.Text(
```

```
        main_frame, width=80, height=20, wrap="word", bg="white",
fg="black"
    )
    results_text.grid(row=3, column=0, sticky=(tk.W, tk.E))

    # Progress Bar
    progress_var = tk.IntVar()
    progress_bar = ttk.Progressbar(
        main_frame,
        orient=tk.HORIZONTAL,
        length=300,
        mode="determinate",
        variable=progress_var,
    )
    progress_bar.grid(row=4, column=0, pady=10)

    # Button
    check_button = ttk.Button(
        main_frame,
        text="Check Policy",
        command=lambda: start_inference(url_entry, results_text,
progress_var),
    )
    check_button.grid(row=5, column=0, pady=5)

    # Make the columns/rows expand with the window
    root.columnconfigure(0, weight=1)
    root.rowconfigure(0, weight=1)
    main_frame.columnconfigure(0, weight=1)
    main_frame.rowconfigure(3, weight=1)

    # Tag configurations for colored text
    results_text.tag_configure("covered", foreground="green",
font=("Arial", 10, "bold"))
    results_text.tag_configure("error", foreground="red",
font=("Arial", 10))
    results_text.tag_configure("bold", font=("Arial", 12, "bold"))

    return root


if __name__ == "__main__":
    app = create_gui()
    app.mainloop()
```