

LLM Based Privacy Policy Analysis

Internat Real-Time Lab at Columbia University

Instructors:

Prof. Henning Schulzrinne

Prof. Gaston Ormazabal

Students:

Jasna Budhathoki

Andrew Chang

Dhruv Limbani

Siddharth Karmarkar

Rose Beck

Taner Sonmez

Mihir Trivedi

Tom Xu

Tips

Just like regular
PowerPoints, click
the right arrow
here to go to the
next slide.



Project Overview

If you are particularly interested in any specific part, you can also click into it.



- 1) Objectives
- 2) Literature review
- 3) Data Preparation
 - 3.1) Collection & Scrapping
 - 3.2) Chunking
 - 3.3) Labeling
 - 3.4) Train test split
 - 3.5) Vectorization
- 4) AI Experiments
 - 4.1) Zero-Shot
 - 4.2) Static Few-Shot
 - 4.3) RAG
- 5) Summary Across All Experiments
- 6) Next steps

Objectives and Deliverables

Policy Analysis

Privacy policies and Terms of Service are often long, complex, and written in legal jargon that most users skim or ignore. As a result, users unknowingly consent to data practices that may compromise their privacy - including cross-border data transfers, targeted profiling, and the use of their data to train AI/ML models. To address this issue, privacy policies will be analyzed with the goal of increasing transparency, providing users with accessible insights, and enabling more informed decisions regarding their digital footprint.

Project Goal

The goal of this project is to leverage Large Language Models (LLMs) to automatically analyze and summarize privacy policies, with a focus on detecting the presence of specific high-risk clauses. In particular, policies will be flagged if they:

- Transfer user data outside the US/EU/UK
- Share data with advertisers for customer profiling
- Use user data to train or enhance AI/ML models

Expected Deliverables

Browser Extension: A user-friendly browser extension powered with LLM that scans privacy notices in real-time, highlights critical risks and alerts users if the policy includes any of the above high-risk clauses.

Privacy Policy Database: A curated and searchable collection of analyzed privacy policies with risk labels for reference and comparison.

Project Overview

If you are particularly interested in any specific part, you can also click into it.



- 1) Objectives
- 2) Literature review
- 3) Data Preparation
 - 3.1) Collection & Scrapping
 - 3.2) Chunking
 - 3.3) Labeling
 - 3.4) Train test split
 - 3.5) Vectorization
- 4) AI Experiments
 - 4.1) Zero-Shot
 - 4.2) Static Few-Shot
 - 4.3) RAG
- 5) Summary Across All Experiments
- 6) Next steps

2) Literature review

If you wish to skip this part, you can click the back arrow here to go back.



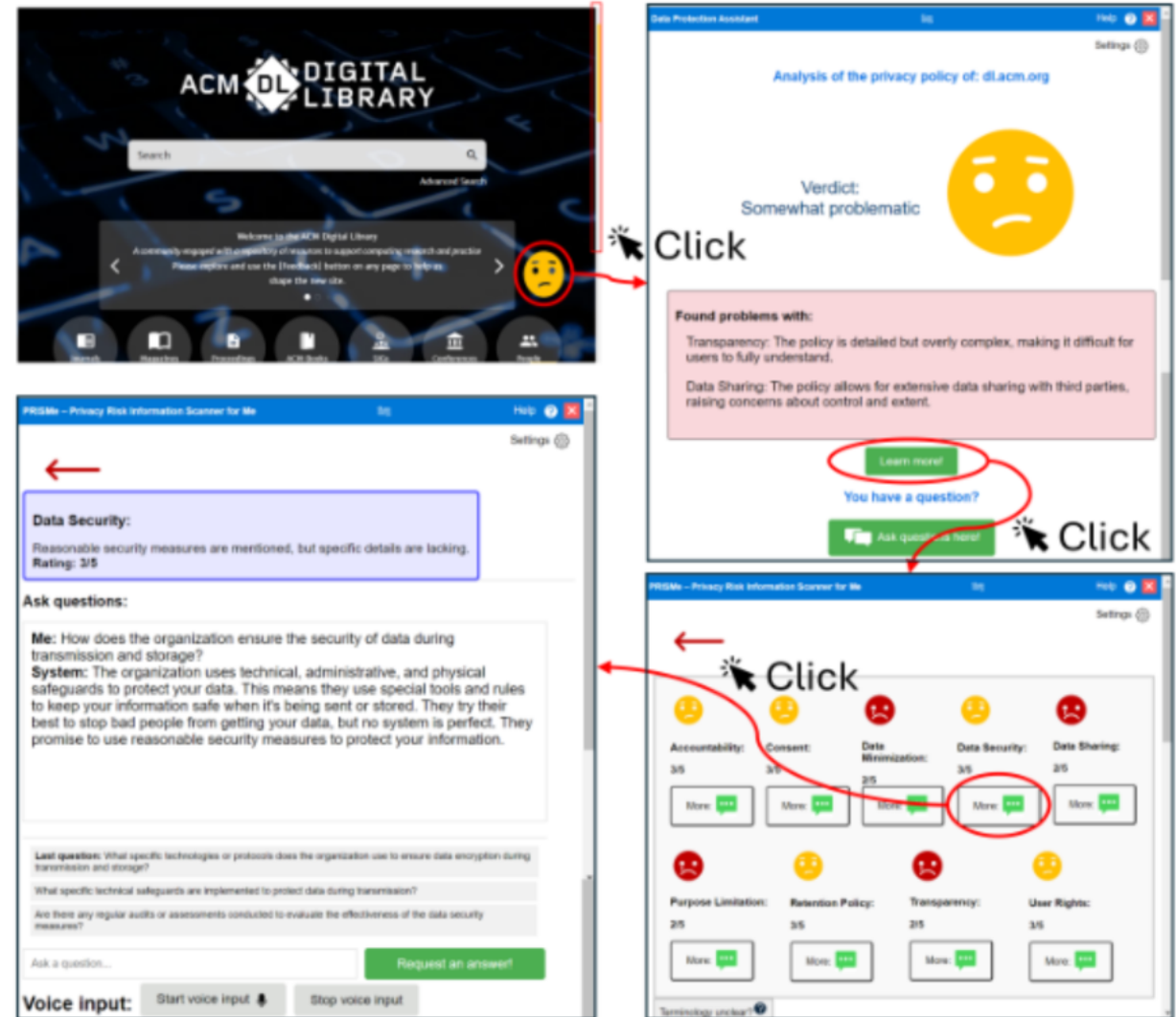
STEP:

Literature Review

PRISMe: A Novel LLM-Powered Tool for Interactive Privacy Policy Assessment, Jan 2025

(Vincent Freiburger, Arthur Fleig, Erik Buchmann, Center for Scalable Data Analytics and Artificial Intelligence, Germany)

- **Overview:** GPT-4o Based Chrome Extension, Visual Cues(colored scrollbars, smiley icons), Interactive Chat Interface
- **Approach:** Real-Time Policy Scraping(with *Pyppeteer*), Storing Analyzed Policies(*SQLite*), Dynamic(Context-Aware) Evaluation with GPT-4o zero-shot COT prompting
- **Results & Methodology:** Mixed-Method User Study (N=22)
 - 3 Scenarios -> SUS Score 88.9% -> Interview
 - Improved Privacy Awareness etc.
- **Key Insights for Our Project:** Interactive Chat, Tailored Explanations Based On Website



Large language models: a new approach for privacy policy analysis at scale

David Rodriguez(1) · Ian Yang(2) · Jose M. Del Alamo(1) · Norman Sadeh(2)

[1] ETSI Telecomunicación, Universidad Politécnica de Madrid, Madrid, Spain,

[2] School of Computer Science, Carnegie Mellon University, Forbes Ave, Pittsburgh, PA 15213, USA

Objective:

- LLMs for categorizing data practice disclosures in privacy policies including Data Collection and Sharing Practices and International Data Transfer
- International Data Transfer Disclosures: Cross-Border Data Transfer, Data Storage Locations, Third-Party
- MAPP Dataset, OPP-115 Dataset, APP-350 Dataset, IT-100 Dataset

Methodology

- Prompt Design: Data Segment, Task Segment, Two-shot examples
- Parameter Tuning: temperature = 0 and top_p = 1 (Optimal)
- Dataset Utilization: OPP-115 dataset (115 privacy policies) for validation, Stratified Sampling
- Evaluation Metrics: Accuracy, Precision, Recall, F1-score

Result

- ChatGPT outperformed traditional statistical and symbolic NLP models with F1 scores exceeding 93%

Personal data type	Experi- mental set	Control set
Computer information	27	22
Contact information	28	29
Cookies and tracking elements	29	25
Demographic data	21	22
Financial	23	17
Generic personal information	28	31
Health, genetic, or biometric data	7	7
IP address and device IDs	31	30
Location	23	23
Other	27	26
Personal identifier	9	7
Political, religious, or philosophical belief	1	1
Social media data	14	11
Unspecified	30	27
User online activities	31	30

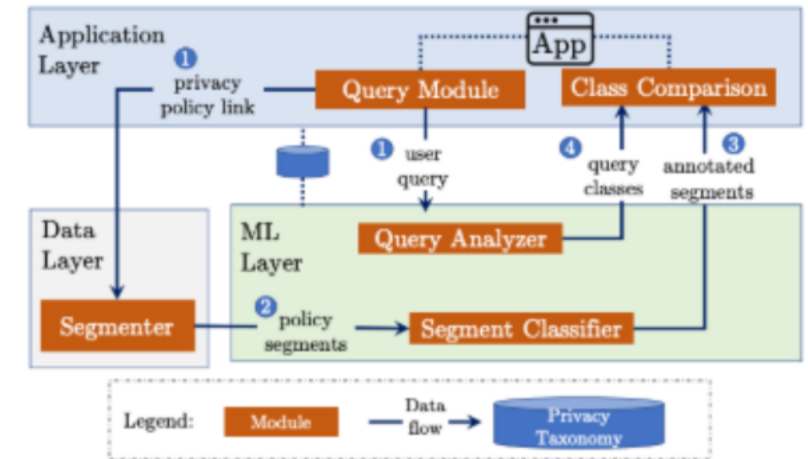
STEP:

Literature Review

Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning

[Hamza Harkous, École Polytechnique Fédérale de Lausanne (EPFL) - Switzerland; Kassem Fawaz, University of Wisconsin-Madison; Rémi Lebret, École Polytechnique Fédérale de Lausanne (EPFL); Florian Schaub and Kang G. Shin, University of Michigan; Karl Aberer, École Polytechnique Fédérale de Lausanne (EPFL)] – 27th USENIX Security Symposium

- **Objective:** Develop a deep learning framework to automate privacy policy analysis, using a free querying interface.
- **Problem Addressed:** Privacy policies are lengthy and complex, deterring user understanding.
- **Methodology:**
 - **Data Layer:** Scrapes and segments privacy policies into coherent pieces for analysis.
 - **Machine Learning Layer:** Utilizes deep learning for analyzing segmented text, employing a hierarchy of neural network classifiers.
 - **Application Layer:** Allows both structured and free-form querying of policies.
- **Key Highlights:**
 - Utilizes a dataset of 130,000 privacy policies for training deep learning models.
 - Provides tools like PriBot for querying policies using natural language and automated privacy icons for visual summaries.



My Takeaways: -

- The paper presents an approach that is similar to ours and has predefined labels for the privacy policy standards. However, since it uses Deep Learning, it can be improved by using LLMs instead.
- They also have a dataset of **130k policies** that we could leverage during our own project development.
- This paper supports structured and free querying, another aspect that could be considered to improve user ease of use.
- Since our idea borders on a web extension instead of directly pasting the link, it is an upgrade on the original paper.

Labels: -

The high-level categories used in the Polisis framework for classifying privacy policy segments are as follows:

1. First-Party Collection/Use
2. Third Party Sharing/Collection
3. User Choice/Control
4. User Access, Edit, and Deletion
5. Data Retention
6. Data Security
7. Policy Change
8. Do Not Track
9. International and Specific Audiences
10. Other

The “Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning” project demonstrated impressive results, achieving an accuracy rate of 88.4% in classifying privacy policy segments and successfully addressing user queries with PriBot, which provided correct answers within its top-3 results for 82% of the test questions. Additionally, a user study involving 700 participants found that PriBot’s responses were relevant to users in 89% of cases, underscoring the system’s effectiveness in enhancing transparency and accessibility of privacy policies.

(Bhanuka Silva , Dishanika Denipitiyage, et.al (Sydney,Australia)

- **Motivation & Problem:** The paper tackles the challenge of making long, legalistic privacy policies understandable for end users.
- **Objective:** Proposes an entailment-driven framework that uses LLMs to classify and explain privacy policy paragraphs.
- **Methodology:** Describes the multi-stage pipeline:
 - **Explained Classifier:** Generates class labels with reasoning.(Llama 2)
 - **Blank Filler:** Masks the reason and predicts the missing text.(Llama 2)
 - **Entailment Verifier:** Checks for consistency between the original reasoning and the regenerated text.(BERT Encoder)
- **Technical Highlights:** Use of autoregressive LLMs and encoder-based models.

Input: Privacy Paragraph Text

"...We are not responsible for the use made by third parties of information you post or otherwise make available in public areas of the Science Website. We retain indefinitely all the information we gather about you in an effort to make your repeat use of our website more efficient, practical, and relevant."

Stage 1: Identify classification labels and corresponding reasons for a given paragraph. These labels belong to 12 categories; e.g. "first party collection/use", "third party sharing / collection", and "user choice / control",...

Explained
Classifier

Class: **Data Retention**

Reasoning: This paragraph mentions about: "**We retain indefinitely**"

Stage 2: Hallucination detection and masking the reason from original paragraph

"...We are not responsible for the use made by third parties of information you post or otherwise make available in public areas of the Science Website. **<Blank>** all the information we gather about you in an effort to make your repeat use of our website more efficient, practical, and relevant."

Stage 3: Given the class, generate a text to fill the <Blank>

Blank Filler

<Blank>: **We will maintain**

Stage 4: Entailment verification based on previous outputs

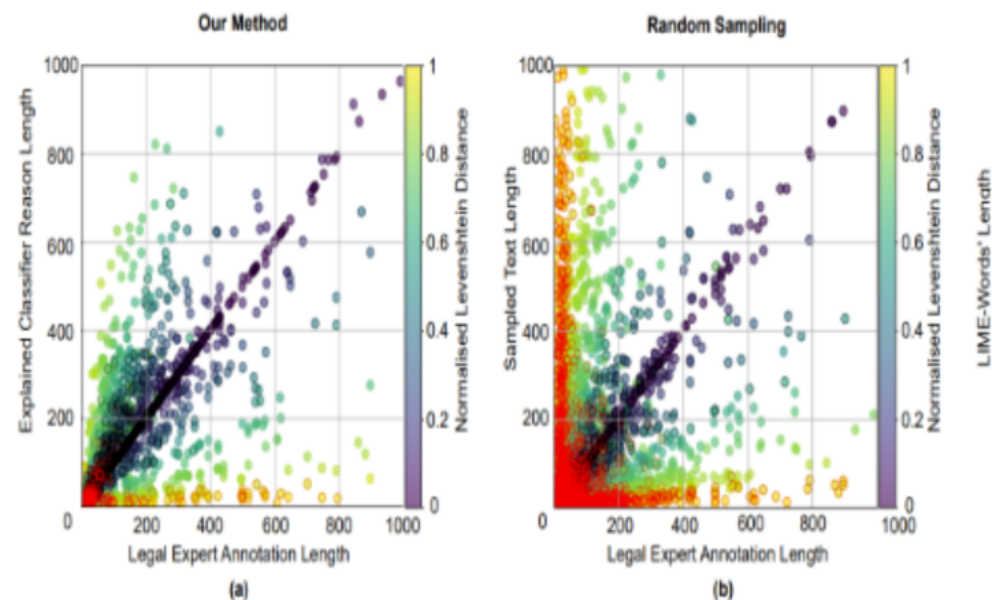
Entailment
Verifier

Entailment

STEP:

Literature Review

- **Findings:**
 - **Experimental Setup:** Use of the OPP-115 dataset with a multi-label classification setup.
 - **Results:** Key performance gains (e.g., improvements in F1 score by 8.6%, 14.5%, and 10.5% over baseline methods such as T5, GPT-4, and LLaMA 2).
- **Key Insights:**
 - The framework produces interpretable reasoning alongside predictions, reducing LLM hallucination issues.
 - Robust performance indicates potential for wider application in privacy-related domains.



STEP:

Literature
Review

FedAloT: A Federated Learning Benchmark for Artificial Intelligence of Things

(Samiul Alam, Tuo Zhang, Tiantian Feng, Hui Shen, Zhichao Cao, Dong Zhao, JeongGil Ko, Kiran Somasundaram, Shrikanth S. Narayanan, Salman Avestimehr, Mi Zhang) (Journal of Data-centric Machine Learning Research (DMLR)) (OSU,USC,Meta)
(Submitted on 29 Sep 2023 (v1), last revised 21 Aug 2024)

Background:

1. AIoT: Integration of **AI** with **IoT**, enabling smart devices to collect, analyze, and act on data in real-time;
2. Federated Learning: Decentralized **ML** approach that allows multiple devices or edge nodes to collaboratively train a model without sharing raw data;

Overview:

1. A benchmark for FL in AIoT that provides **standardized datasets**, **evaluation metrics**, and **baseline models** for comparing FL methods in AIoT scenarios;
2. Addresses **Non-IID data** challenges, **client sampling ratio**, and **device heterogeneity** in AIoT environments;
3. Simulated challenges in real-world setting: **Noisy Labels** and **Quantized Training**

Key Components:

Diverse AIoT Datasets, Flexible FL Training Schemes, Built-in FL Models, IoT-Factor Emulator;

Key Takeaways:

1. FedAloT serves as the first **standardized benchmark** to fairly compare FL methods;
2. **Non-IID data** is a major FL challenge;
3. **Higher client participation** boosts accuracy but raises costs;
4. **Noisy labels** significantly reduce FL model accuracy;

STEP:

Literature Review

PolicyPulse: Precision Semantic Role Extraction for Enhanced Privacy Policy Comprehension

(Andrick Adhikari, Sanchari Das, Rinku Dewri (University of Denver))

(Paper is scheduled to be presented and published at Network and Distributed System Security Symposium (NDSS) 2025 Feb 24-28)

Motivation:

Challenges with Privacy Policies:

- Are often **too long and complex** for the general public.
- Some statements are vague
- Users struggle to quickly find relevant information.

Existing NLP techniques for privacy policy analysis:

- Often just provide **high-level summaries** or focus on narrow tasks like extracting specific clauses only.
- Existing tools often only extract individual entities (e.g., "email is collected") without describing relationships. --> lacks **detailed explanations**
- Need for a more **flexible and comprehensive** system that can handle a variety of tasks.

PolicyPulse:

- **dissects each sentence to figure out** "who," "what," "when," and "why"
 - **who does what with what data and why**
- **Labels language semantic roles into privacy specific roles**
- It then creates a **structured database** of these details, making it easy to create **clear summaries**, **answer user questions**, or **check if a policy is missing important details**.

Methodology:

1. Corpus Preparation & Frame Extraction

- **Privacy Policies:** Collected from sources like **OPP-115** and **Princeton Privacy Crawl** (PPCrawl - 1,071,488 privacy policies from 130,604 websites 1997-2019)
- **BERT-based Semantic Role Labeling: SRL BERT (AllenNLP)**- Break each sentence into **verb-centered semantic frames (NLP labels)**. Identifies **predicates** (verbs) and **arguments** (ARG0, ARG1, ARGM, etc.).

2. Two-Level XLNet Classifiers

- Comparing **XLNet, BERT, and traditional classifiers** for classifying privacy-related text, XLNet outperformed BERT-based models in semantic frame classification & had better precision (93%) and recall (95%)

- **trained on 13,946 frames that were manually annotated for privacy-related categories**

- **Level 1: SKIP vs. KEEP Frames**
 - Filters out irrelevant or incomplete frames (e.g., fragments lacking enough context).
- **Level 2: For semantic frames labeled KEEP, a second XLNet classifier assigns them to one of five privacy categories:**
 - **FPCU** (First Party Collection/Use)
 - **TPSC** (Third Party Sharing/Collection)
 - **UCC** (User Control & Choices)
 - **DR** (Data Retention)
 - **UAED** (User Access, Edit, and Deletion)

3. Privacy-Specific Role Mapping:

- Rename the generic SRL arguments (like ARG0, ARG1) to privacy roles like 'DATA,' 'FIRST_PARTY_ENTITY,' 'RETENTION_PERIOD,' or 'OPT_OUT_MECHANISM.'
 - Transforms unstructured text to machine-readable text
- **Manual Verification** of 146 verbs to **ensure** correct "privacy" labeling (e.g., "collect," "share," "retain").

Privacy Structured Knowledge Base Creation

- The processed **semantic frames** and their associated **role labels** are stored in a **structured database**
- **Granular Representation:** For each policy, **store** frames with privacy roles & categories.
- **Enables:**
 - Policy **completeness checks** (e.g., missing retention info).
 - **Short notices & nutrition labels** (automatic summaries).
 - **User preference checking** (detect conflicts like "marketing" use).
 - **Query answering** (pull relevant frames for user queries).

Evaluation:

- **10-Fold Cross-Validation** (9:10 train-test split ratio) on the **annotated frames** from OPP-115.
- **Scalability & Large scale testing (PPCrawl Data)**
- Achieved **high accuracy**—an **F1-score around 0.97** for classifying frames.

Potential Applications:

1. **Policy Completeness Checks** → Finding missing or unclear details in privacy policies
2. **Alternative Presentations** → Generating **short summaries** (like bullet points or privacy nutrition labels) instead of long legal text.
3. **Automated Query Answering** → Users can **ask questions** like "Does this policy allow selling my data?" and get an **exact answer**.
4. **User Preference Checking** → Matching a user's privacy preferences (e.g., "I don't want my data shared") against policy statements.

types of information	how we use your information								who we share your information with		
	marketing	basic service feature	additional service feature	service operation and security	personalization and customization	analytics/research	merger/acquisition	advertising	third party	partner	advertiser
personal information	!	UTR	UTR	!	-	!	!	-	UTR	UTR	!
device identifier	-	-	-	-	!	-	-	!	-	-	-
geolocation	-	!	-	-	-	-	-	-	-	-	-
contact information	!	!	-	!	-	!	-	-	!	-	-
online activities	-	-	-	-	-	-	-	-	!	-	-
social media data	-	-	UTR	-	-	-	-	-	UTR	-	-
cookie/pixel tag	-	-	!	-	!	UTR	UTR	-	!	-	-
financial information	-	-	-	!	-	UTR	-	-	!	-	-

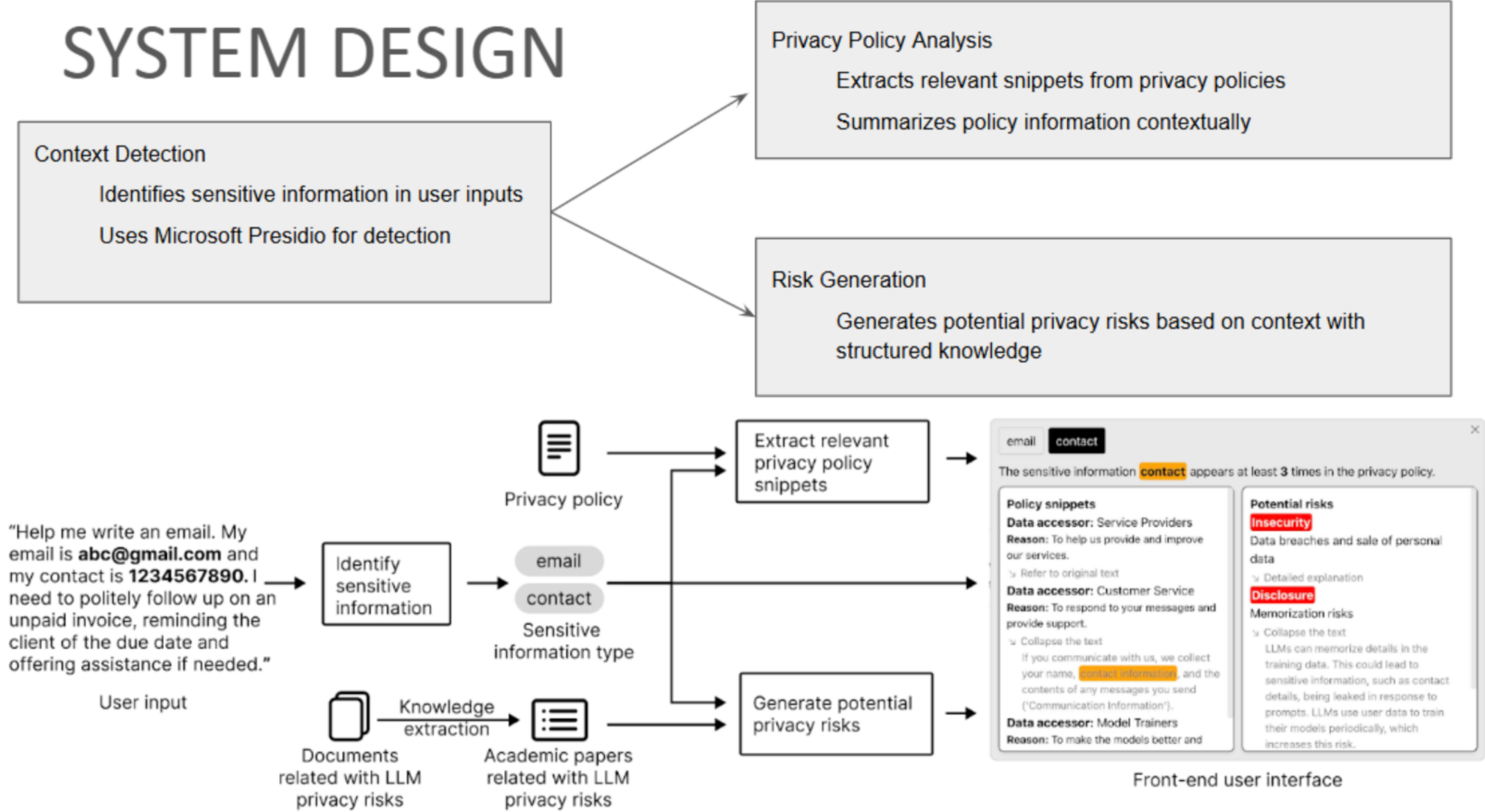
! information will be used for the purpose or shared
 UTR some user action will trigger use or sharing of the information
 - information will not be used for the purpose

Fig. 6. Privacy nutrition label generated from semantic frames on *booking.com*'s 2018 privacy policy

Rows: Types of User Information Collected
Columns: How the Information is Used
Far right: Who the data is shared with

- Extracts structured information as readable, tabular format
- Reveals potential gaps in the policy

SYSTEM DESIGN



STEP:

Literature
Review

Results

Quantitative Measurements:

System Usability Scale (5-point Likert scale)

- Case Study 1 (ChatGPT)
 - Increased awareness of data practices
 - Remove/modify sensitive information
- Case Study 2 (Gmail + Gemini)
 - Positive user experience
 - Cautious about sharing sensitive data
- Key Outcomes
 - Improved understanding of privacy risks

- Measured aspects:
 - Ease of use
 - System integration
 - Learning curve
 - Confidence in using the system
 - Information usefulness
 - Policy snippet appropriateness



Empowering Users in Digital Privacy Management through Interactive LLM-Based Agents

Accepted for ICLR-2025, Authors: Bolun Sun (Johns Hopkins University), Yifan Zhou (The University of Georgia), Haiyun Jiang (Sun Yat-sen University)

Goal: An innovative LLM-based agent that functions as an expert system for processing website privacy policies, guiding users through complex legal language without requiring them to pose specific questions.

Main Contributions:

1. Empirical proof of superiority of LLMs over traditional models for privacy policy analysis
2. LLM-based Agent for:

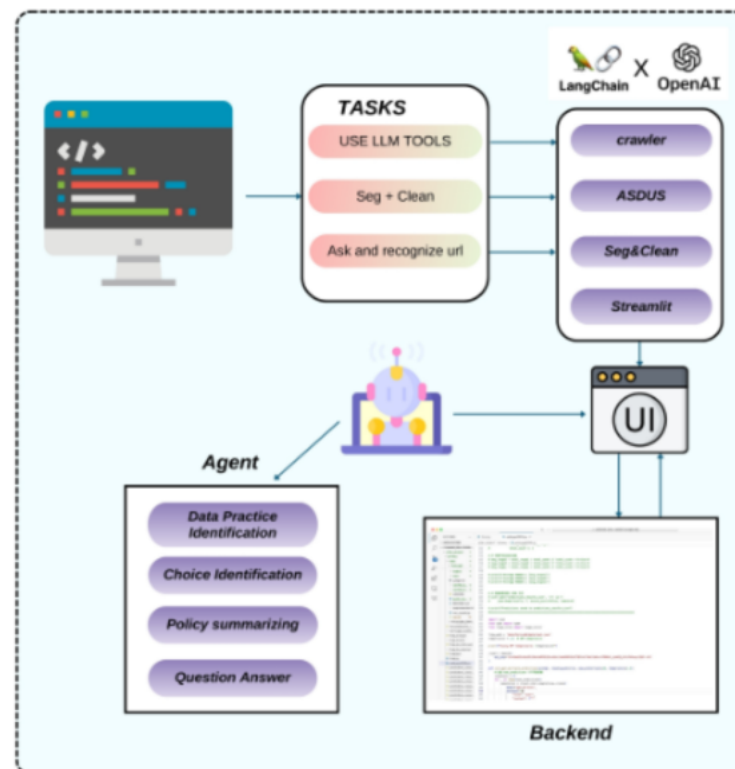
Task	Dataset	Models	Metrics
Data practice identification	OPP-115	GPT-4o-mini, LR, SVM, HMM	Precision, Recall, F1-score
Choice identification	OPP-115	GPT-4o-mini (zero shot), GPT-4o-mini (few shots), LR, BERT, fastText	Precision, Recall, F1-score, Accuracy

Other tasks: Policy QnA, Policy Summarization

3. Evaluation of agent's impact on end-users

Limitation:

1. Old dataset (OPP-115)
2. Low Recall
3. Does not address prioritization of recall and precision for individual labels



Experimental Results:

Category	GPT-4o-mini			LR			SVM			HMM		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
First Party Collection/Use	0.95	0.64	0.77	0.73	0.67	0.70	0.76	0.73	0.75	0.69	0.76	0.72
Third Party Sharing/Collection	0.84	0.69	0.75	0.64	0.63	0.63	0.67	0.73	0.07	0.63	0.61	0.62
User Choice/Control	0.88	0.43	0.58	0.45	0.62	0.52	0.65	0.58	0.61	0.47	0.33	0.39
User Access, Edit, & Deletion	0.90	0.59	0.71	0.47	0.71	0.57	0.67	0.56	0.61	0.48	0.42	0.45
Data Retention	0.96	0.16	0.27	0.10	0.35	0.16	0.12	0.12	0.12	0.08	0.12	0.09
Data Security	0.97	0.44	0.61	0.48	0.75	0.59	0.66	0.67	0.67	0.67	0.53	0.59
Policy Change	0.86	0.59	0.70	0.59	0.83	0.69	0.66	0.88	0.75	0.52	0.68	0.59
Do Not Track	0.64	0.88	0.74	0.45	1.00	0.62	1.00	1.00	1.00	0.45	0.40	0.41
International & Specific Audiences	0.95	0.77	0.88	0.49	0.69	0.57	0.70	0.70	0.70	0.67	0.66	0.66
Other	0.91	0.35	0.51	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Micro-Average	0.90	0.53	0.67	0.53	0.65	0.58	0.66	0.66	0.66	0.60	0.59	0.60

Choice Identification

- **Precision** is important when false positives are harmful.
- **Recall** is important when false negatives are harmful.

Metric	GPT-4o-mini (zero shot)	GPT-4o-mini (few shots)	LR	BERT	fastText
Precision	0.74	0.88	0.90	0.83	0.90
Recall	0.94	0.95	0.86	0.98	0.76
F1-score	0.83	0.91	0.88	0.90	0.82
Accuracy	0.94	0.93	NaN	NaN	NaN

Project Overview

If you are particularly interested in any specific part, you can also click into it.



- 1) Objectives
- 2) Literature review
- 3) Data Preparation
 - 3.1) Collection & Scrapping
 - 3.2) Chunking
 - 3.3) Labeling
 - 3.4) Train test split
 - 3.5) Vectorization
- 4) AI Experiments
 - 4.1) Zero-Shot
 - 4.2) Static Few-Shot
 - 4.3) RAG
- 5) Summary Across All Experiments
- 6) Next steps

STEP:

Data

3) Data

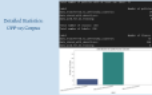


What data was collected?

How was data collected?

STEP:
Data
What Data

What data was collected

The database is composed of the following three datasets:

OPP-115 Dataset	General Privacy Policy Labelled Dataset	IoT Policy Dataset
<div><ul style="list-style-type: none">The OPP-115 Corpus is a collection of website privacy policies annotated by law graduate students to capture diverse data practices such as data collection, sharing, and usage.It does not contain clauses for AI training since it was created in 2016 when LLMs were not prevalent.Click on the diagram preview below for more information<div></div></div>	<div><ul style="list-style-type: none">A curated collection of modern privacy policies and terms of service sourced from the most visited websites in USA.Most visited websites were identified from:<ol style="list-style-type: none">similarweb.comdataforseo.comTotal Number of policies: 306Collected in April 2025Click on the diagram preview below for more information<div></div></div>	<div><ul style="list-style-type: none">A collection of privacy policies and terms of service of most popular IoT vendors' websites in USA.Most visited websites were identified from:<ol style="list-style-type: none">datamation.comiotforall.comTotal Number of policies: 95Collected in April 2025Click on the diagram preview below for more information<div></div></div>
Respective Number of Policies Related to The 3 Labels of Interest		
Data transferred outside US/EU/UK 24	Data transferred outside US/EU/UK 182	Data transferred outside US/EU/UK 65
Data shared with advertisers 46	Data shared with advertisers 294	Data shared with advertisers 83
Data used for AI training 0	Data used for AI training 63	Data used for AI training 22

What data did we collect

After duplicates between the datasets were resolved, they were merged to form the **Aggregated Dataset**.

A summary of its statistics is presented below:

Aggregated Dataset (OPP115+General+IoT)

Total Number of Policies with At Least One Label:		454
Label	Number of Policies	
data_transferred_to_unfriendly_countries	271	
data_shared_with_advertisers	423	
data_used_for_AI_training	85	
Total Number of Clauses with At Least One Label:		5549
Label	Number of Clauses	
data_transferred_to_unfriendly_countries	711	
data_shared_with_advertisers	4670	
data_used_for_AI_training	190	

STEP:

Data

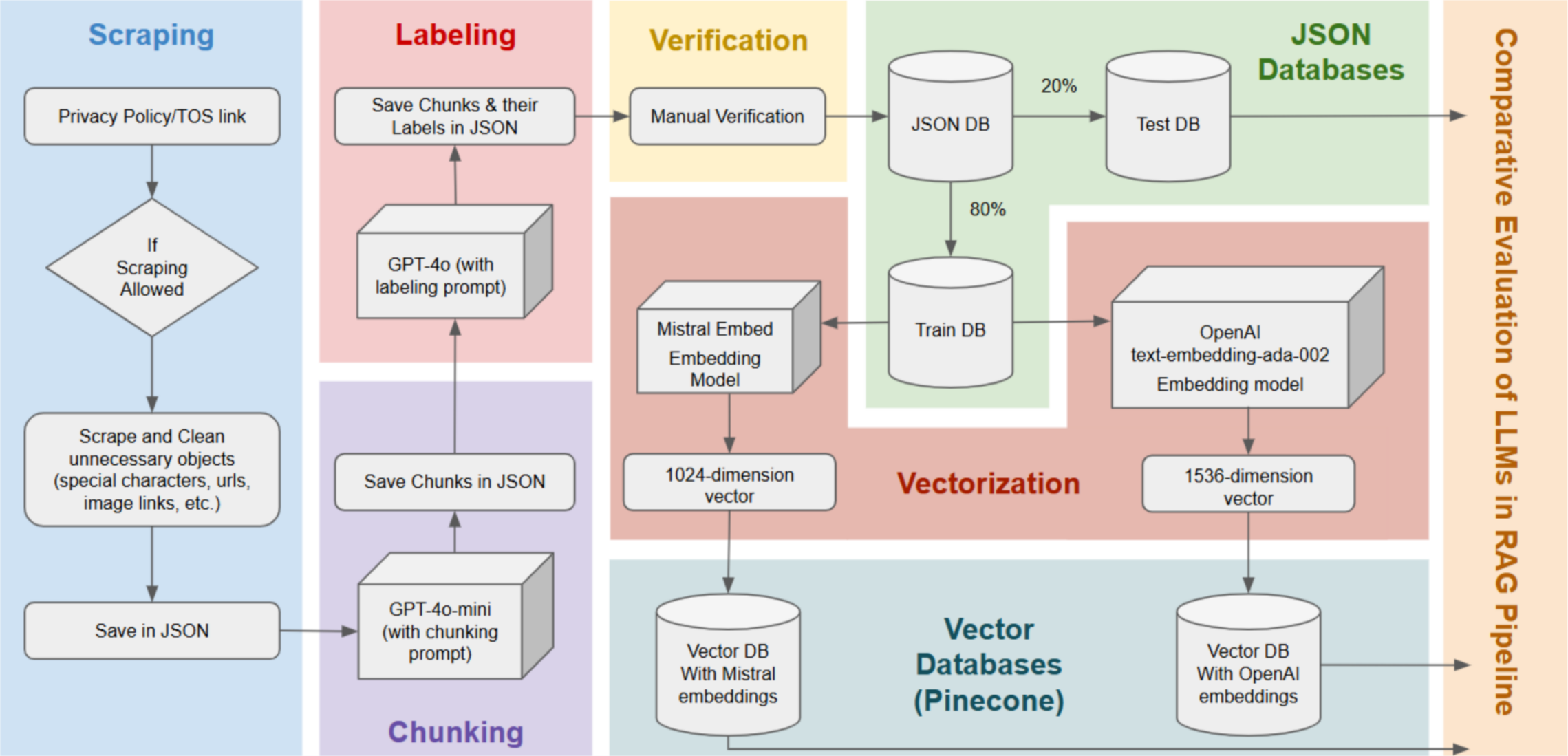
3) Data

What data was collected?

How was data collected?

STEP:
Data
How Data

How was data collected



Pipeline - Scraping

The top 1,000 most visited websites in the U.S., based on network traffic, were first identified using DataForSEO, and their URLs were subsequently downloaded in a CSV file.



Check the Top 1000 Websites in the US And 74 More Countries

By default, the table lists the top 1000 websites worldwide. However, it can also provide you with region-specific stats for free. Select a country from the drop-down list above the table, and hit *Search* to see the top 1000 websites in a particular location.

United States

Download

Rank	Domain	Keywords in SERPs	Estimated organic traffic
1	youtube.com	464041873	14925082687.04
2	facebook.com	315508273	7860966209.06
3	wikipedia.org	309027296	40976022085.87
4	instagram.com	279399070	10024312989.69
5	tiktok.com	272106215	1529777681.33
6	reddit.com	248576886	5906039636.01
7	pinterest.com	129522687	4183219077.54
8	quora.com	113361767	1328288643.54
9	amazon.com	105969428	3298148109.74

Pipeline - Scraping

- 2. Web scraping was performed for the **first time**:
 - a. GET requests were made to retrieve the HTML content of the 1,000 websites;
 - b. Links to Privacy Policies and Terms of Service were extracted from the HTML using BeautifulSoup and regular expressions;

Rank	Domain	Keywords in SERPs	Estimated organic traffic
1	youtube.com	464041873	14925082687.04
2	facebook.com	315508273	7860966209.06
3	wikipedia.org	309027296	40976022085.87



	Domain	Privacy Policy	Terms of Use
2	youtube.com	https://youtube.com/t/privacy	https://youtube.com/t/terms
3	reddit.com	https://www.reddit.com/policies/privacy-policy	https://www.redditinc.com/policies/user-agreement
4	facebook.com	https://www.facebook.com/privacy/policy/	https://www.facebook.com/terms/
5	tiktok.com	https://www.tiktok.com/legal/page/us/privacy-policy/en	https://www.tiktok.com/legal/page/us/terms-of-service/en
6	wikipedia.org	https://foundation.wikimedia.org/wiki/Special:MyLanguage/Policy:Privacy_policy	https://foundation.wikimedia.org/wiki/Special:MyLanguage/Policy:Terms_of_Use
7	instagram.com	https://help.instagram.com/155833707900388	https://help.instagram.com/termsofuse

STEP:

Data

How Data

Pipeline - Scraping

Pipeline - Scraping

- 3. All extracted links were manually verified by visiting them in a browser to ensure their correctness.
- 4. Web scraping was then performed a **second time** to collect the actual content of the Privacy Policy and Terms of Service pages:
 - a. GET requests were issued to retrieve the full page content;
 - b. Selenium was used to load pages with JavaScript-rendered elements;
 - c. Scraping permissions were confirmed by parsing each site's robots.txt file using urllib.robotparser;
 - d. Page content was extracted and cleaned using BeautifulSoup, regular expressions, and chardet for encoding detection;
 - e. The cleaned content was saved in structured JSON files;
 - f. Scraping statistics and errors were tracked and recorded in a .txt report.

```
"pps": [  
  "Awair Terms ProductsAwair ElementAwair OmniAwairNetConnectivityAwair DashboardSolutionsSchoolGive parents, students, and teachers 24/7 peace of mind with Awair  
  solutions for Schools.BusinessReassure employees and customers that your business is safe with Awair for Business.OfficeCreate transparency and monitor IAQ trends for your  
  clients, tenants, or employees with Awair for Offices.DiscoverResourcesBlogSupportEnterprise SalesLoginBuy ElementBuy ElementLegalTermsPrivacyWarrantyAPIPartnerBusiness  
  EULALast Updated - 1082020Bitfinder, Inc. and its affiliates, if any, collectively referred to as Bitfinder, we, us, our manufactures, markets, and sells Awair consumer and
```

•
•

STEP:

Data

How Data

Pipeline - Scraping

Pipeline - Scraping

This concluded the scraping process for the top 1,000 most visited websites.

Next, websites related to the Internet of Things (IoT) were explored. Our focus was to investigate whether their Privacy Policies and Terms of Service exhibit any distinct patterns related to the three labels of our interest.

The IoT categories examined included:

- a. Smart Cameras: TP-Link, Ring, Wyze, Cinnado, Eufy, Blurams
- b. Television: Samsung, LG, TCL, Panasonic
- c. Smart Appliances: Samsung SmartThings, LG ThinQ
- d. Smart Lighting: Philips Hue, LIFX
- e. Smart Thermostats: Nest, Ecobee
- f. Smart Plugs: TP-Link Kasa, Wemo
- g. Car Infotainment Systems
- h. Printers
- i. Smart Toys For Children

STEP:

Data

How Data

Pipeline - Scraping

Pipeline - Scraping

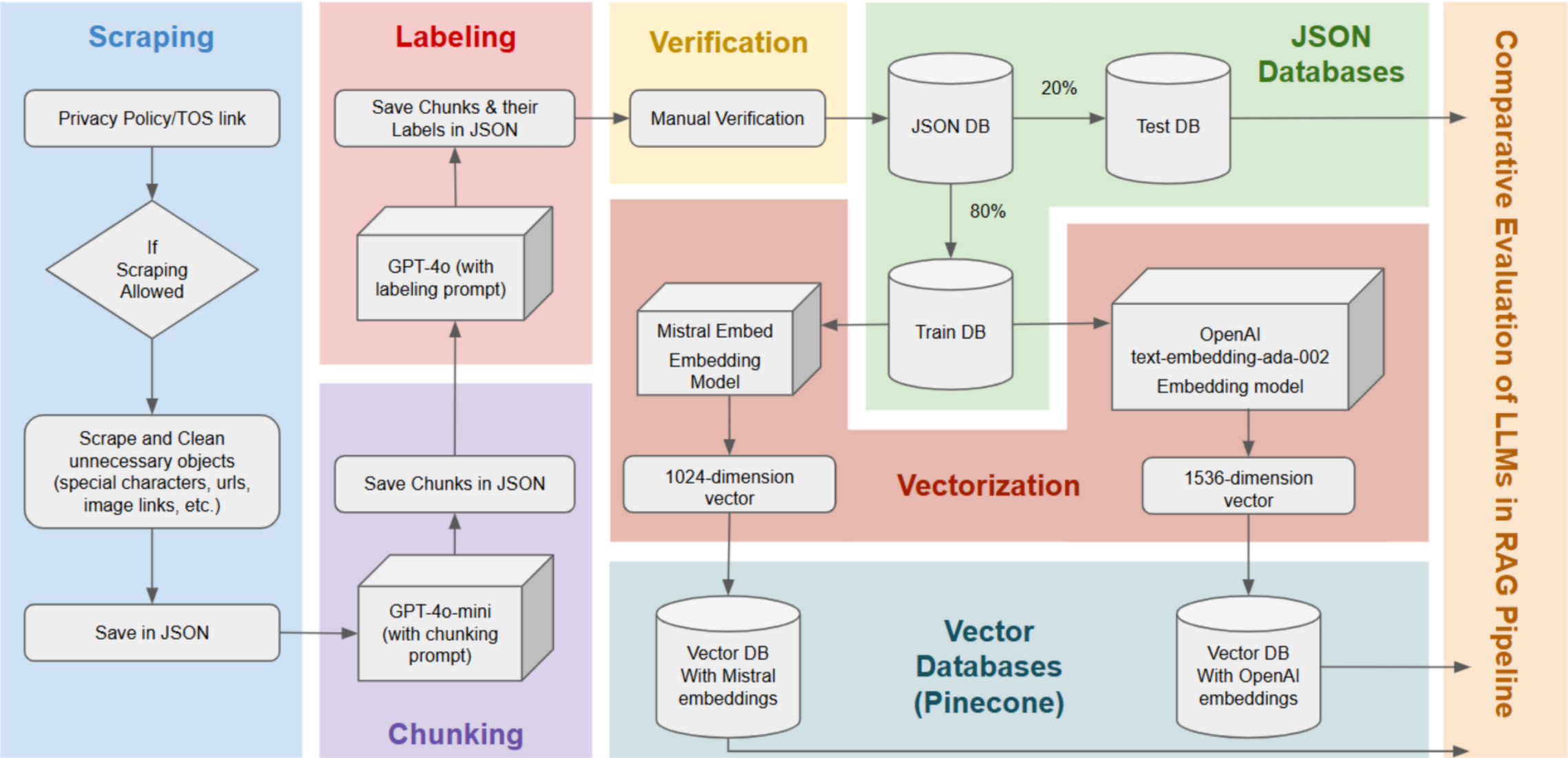
The top 130 most visited IoT websites in the U.S. were identified using the following sources:

<https://www.datamation.com/mobile/85-top-iot-devices/>

<https://www.iotforall.com/>

The same scraping process described earlier was then applied to the identified IoT websites, and their Privacy Policies and Terms of Service were stored in separate JSON files.

How was data collected



STEP:

Data

How Data

Pipeline - Chunking

Pipeline - Chunking

At this stage, the Privacy Policies and Terms of Service had been successfully stored in JSON files. However, the content was too lengthy to be easily read by humans or directly processed by LLMs.

To address this, the documents were divided into smaller, more manageable chunks to facilitate both human review and efficient LLM processing.

Two chunking strategies were adopted:

- a. GPT-4o-mini;
- b. Python scripts with regex;

STEP:

Data

How Data

Pipeline - Chunking

GPT Chunking

Pipeline - Chunking - (GPT)

The JSON files were input into GPT, which was prompted to generate chunked versions of the text. The resulting chunks were also stored in JSON format.

```
CHUNK_PRIVACY_POLICY_PROMPT = """
You are a legal expert tasked with breaking down a Privacy Policy, with no modifications, to coherently, atomically independent chunks/clauses.
# Task Description
- You will be provided with a Markdown of the privacy policy page
- Your job is to break down that Privacy Policy into clauses
- Once done, you will return an array of strings, each containing a different clause of the policy
# Important notes
- Do not repeat clauses multiple times
- Do not modify, in any shape or form, the language of the privacy policy clauses
- Do not create or deduce any new privacy policy clauses
- Return the privacy policy clauses in the order they were originally mentioned in the website
"""
```

GPT-4o-mini



```
{
  "www.getawair.com": {
    "pp": "Awair Terms ProductsAwair Elem
solutions for Schools.BusinessReassure employ
clients, tenants, or employees with Awair for
EULALast Updated - 1082020Bitfinder, Inc. and
enterprise electronics products and other pro
provides services including, but not limited
which may be updated or altered from time to
to your use of other Bitfinder services Conte
through the Mobile Apps. The term Services re
therewith.THIS IS A LEGAL AGREEMENT. By acces
Terms of Service Agreement and any additional
represent and warrant that you have the right
that you are of sufficient legal age in your
Terminationa. Overview and Relation to Other
```

Output



```
{
  "www.getawair.com": {
    "pp_chunked": [
      "Bitfinder, Inc. and its affiliates, collectively
products and other products made by Bitfinder and provides ent
      "Bitfinder additionally provides services includin
your smartphone or tablet (Mobile App) which may be updated or
analytics, suggestions or information relating to your use of
user accounts (Accounts) that may be accessed through the Mobi
      "The term Services refers collectively to the Site
      "THIS IS A LEGAL AGREEMENT. By accessing or using
Terms of Service Agreement and any additional terms incorporat
      "You represent and warrant that you have the right
      "You represent that you are of sufficient legal ag
      "I. Overview, Eligibility, Terms and Termination",
      "a. Overview and Relation to Other Agreements. The
```

STEP:

Data

How Data

Pipeline - Chunking

Python Chunking

Pipeline - Chunking - (Python)

At times, policy content exceeded GPT-4o-mini's input token limit. When this occurred, chunking was performed using a Python-based approach as an alternative.

1. Regular expressions were used to split the text into individual sentences;

```
# Function to split text into sentence-based chunks
def split_text_into_chunks(text, max_chunk_size, overlap_size):
    sentences = re.split(r'(?!\w\.\w\.) (?<[A-Z][a-z]\.)(?<=\.|\?)\s', text) # Split into sentences
```

2. Sentences were then grouped into chunks until a predefined token length threshold was reached;

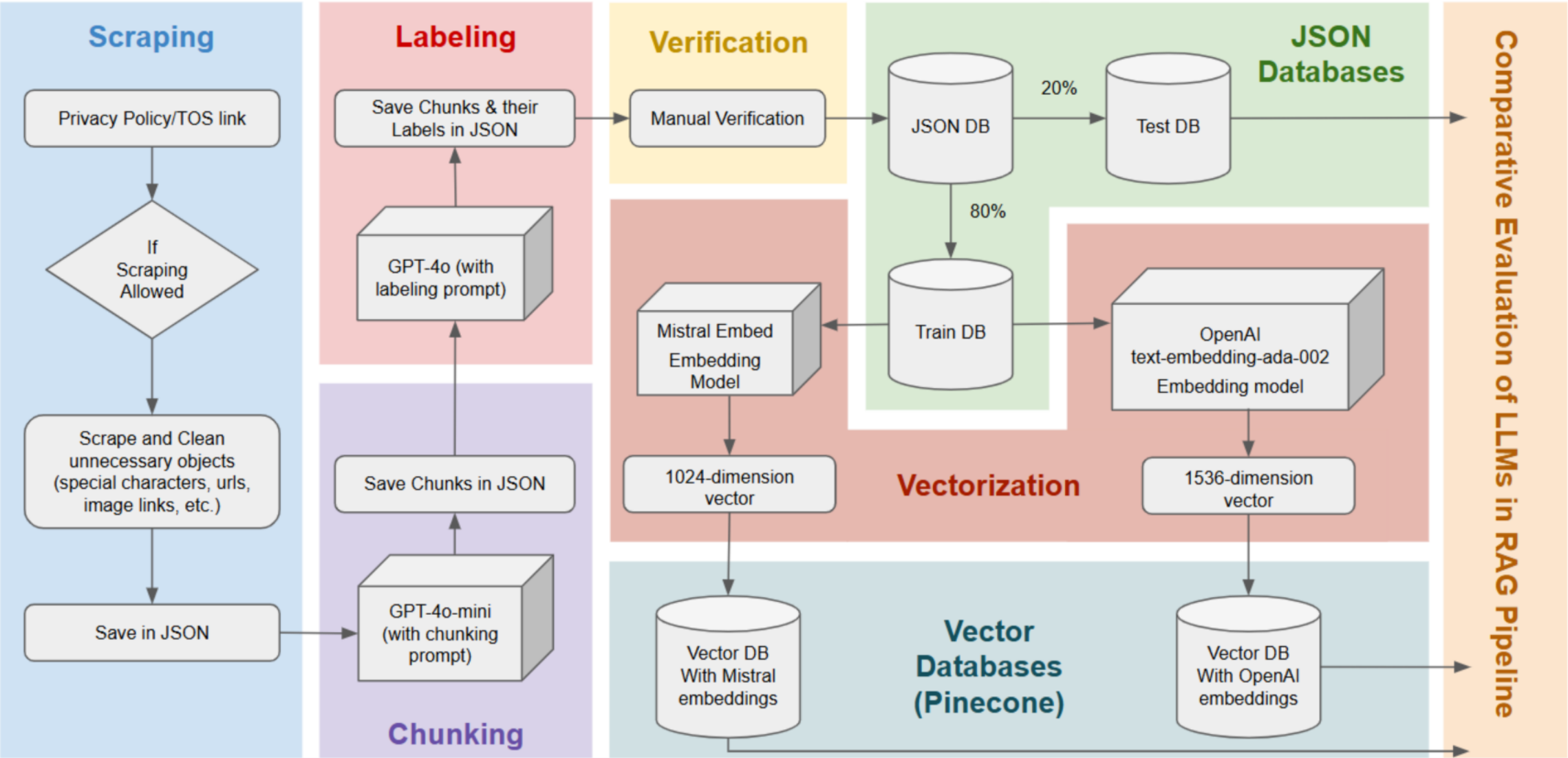
3. To preserve context between chunks, the last sentence (up to 500 tokens) of the previous chunk was appended to the beginning of the subsequent one;

```
# Define chunking parameters
MAX_CHUNK_SIZE = 4000 # Safe for GPT-4o API calls
OVERLAP_SIZE = 500 # Ensures continuity across chunks
```

4. The resulting chunked content was stored in JSON format;

```
# Save structured JSON
output_file_path = "./chunked_policies.json"
with open(output_file_path, "w", encoding="utf-8") as f:
```

How was data collected



Pipeline - Labeling

The chunked policy texts were input into GPT, which was prompted to generate labeled outputs. The labeled text was returned and stored in JSON format.

```

LABEL_PRIVACY_POLICY_PROMPT = """
You are a legal expert tasked with understanding and deciphering the meaning, intention, and implication of privacy policy clauses.
# Task Description
- You will be provided with multiple clauses from the body of a privacy policy.
- You will also be provided with 3 examples of full annotated privacy policies that could be relevant to the clauses you need to classify.
- Your job is to understand each clause based on its content and the annotated policies, then determine if each clause states and/or explicitly implies any of the following:
  -- User data will be transferred outside USA, EU, UK.
  -- User data will be shared with advertisers for customer profiling.
  -- User data will be used for AI model training.
- Once done, you will return a list of objects, where each object contains:
  -- The clause text.
  -- Three booleans indicating which of the above cases, if any, are explicitly implied by that clause.
# Important notes
- Do not attribute unstated or unimplied meaning to any of the clauses.
"""

```

GPT-4o
→

✓ LABELED_POLICIES

{}

247_www.momjunction.com_filtered.json

{}

248_www.wholefoodsmarket.com_filtered.json

{}

249_www.theatlantic.com_filtered.json

{}

250_salemmedia.com_filtered.json

{}

251_testbook.com_filtered.json

{}

252_trust.arcgis.com_filtered.json

Manual
Verification
→

✓ MANUALLY_VERIFIED_LABELED_POLICIES

{}

1_www.redditinc.com_filtered.json

{}

2_www.tiktok.com_filtered.json

{}

3_help.instagram.com_filtered.json

{}

4_policy.pinterest.com_filtered.json

{}

5_legal.yahoo.com_filtered.json

{}

6_www.fandom.com_filtered.json

{}

7_hello.mapquest.com_filtered.json

STEP:

Data

How Data

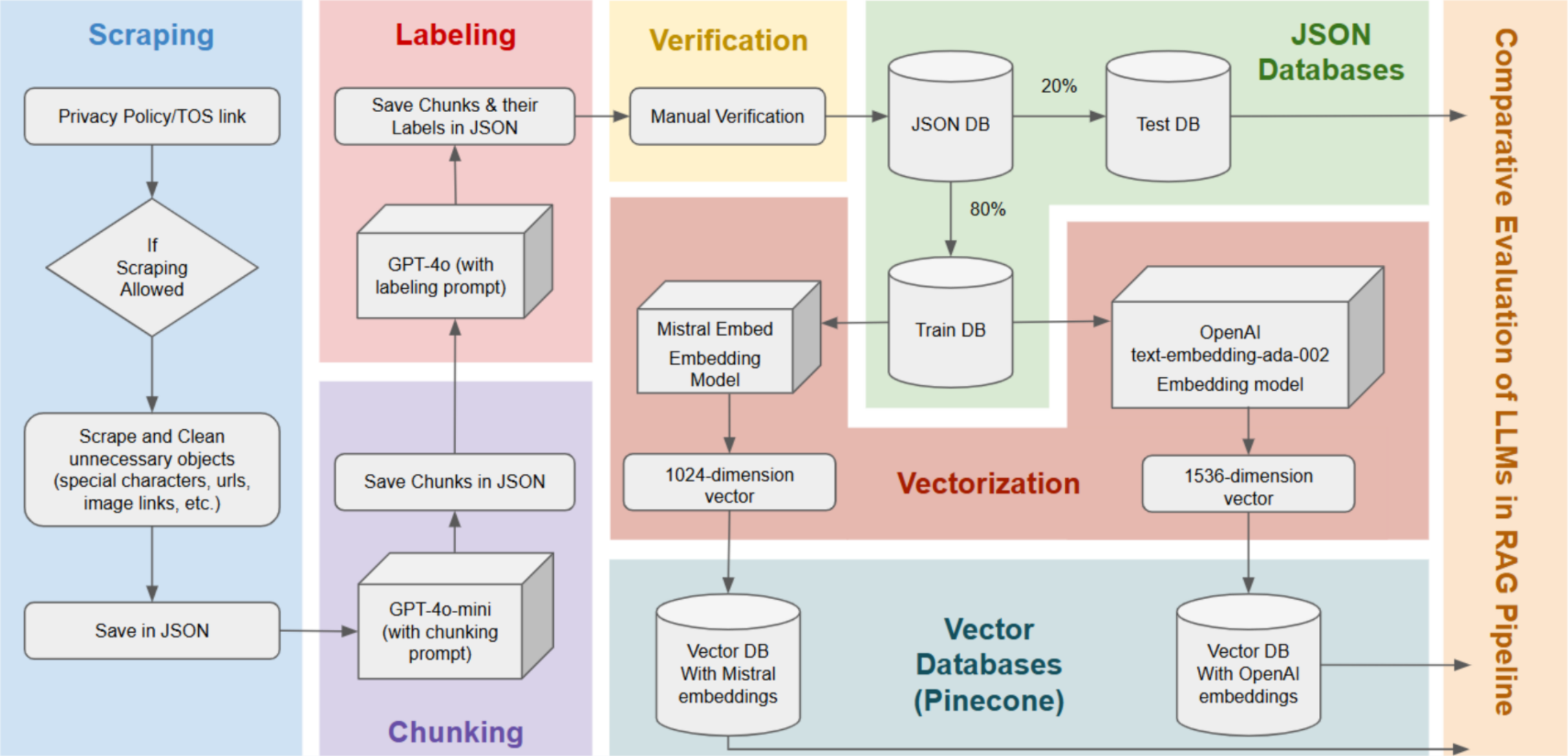
Pipeline - Labeling

Pipeline - Labeling

An example of several manually verified and labeled clauses, stored in JSON format, is presented below.

```
{
  "Where appropriate, we use technical and organizational controls to prevent it from being used improperly, but this information is important to help improve algorithms for product functionality and safety, including features such as mapping topography, obstacles, traffic and the like.": {
    "data_transferred_to_unfriendly_countries": false,
    "data_shared_with_advertisers": false,
    "data_used_for_AI_training": true
  },
  {
    "These enterprises largely fall into the following categories Advertising and marketing companies, data set and information vendors, public database providers, social media platforms, partners, providers of products or services, hosts or vendors at events or trade shows, research partners, enterprises that deploy the Intel Services or third party offerings that include Intel Services.": {
      "data_transferred_to_unfriendly_countries": false,
      "data_shared_with_advertisers": true,
      "data_used_for_AI_training": false
    }
  },
  {
    "This kind of data is used for work like improving algorithms and data models, product testing and improvement, enhancing existing products and developing new capabilities and features.": {
      "data_transferred_to_unfriendly_countries": false,
      "data_shared_with_advertisers": false,
      "data_used_for_AI_training": true
    }
  },
}
```


How was data collected



STEP:

Data

How Data

Pipeline - JSON Database

Pipeline - JSON Database

At this stage, three sets of manually verified and labeled policies had been prepared:

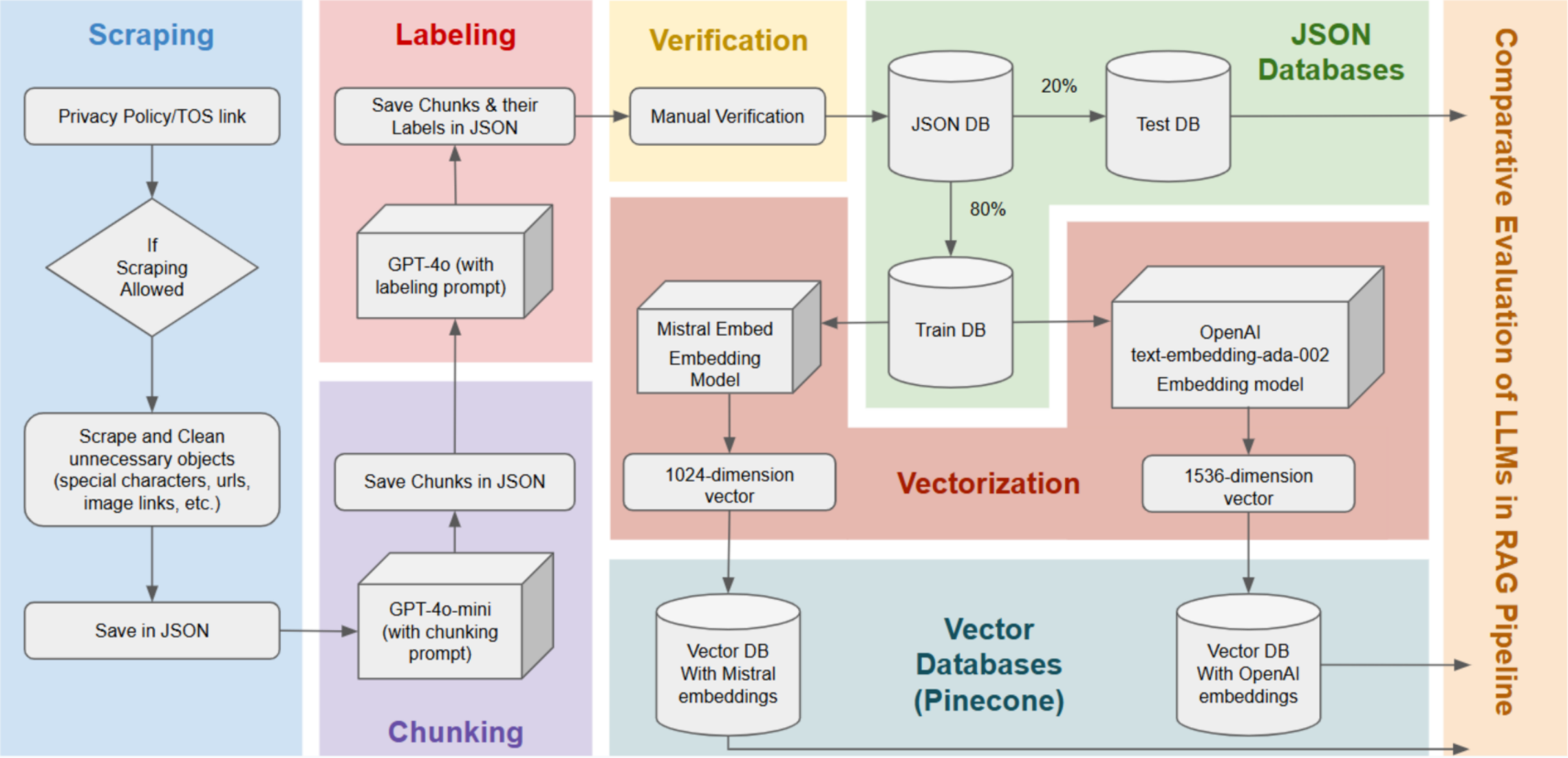
- a. OPP-115;
- b. Top 1,000 Most Visited Websites (General websites);
- c. Most Visited IoT Websites;

To maintain the integrity of each individual dataset, the original sets were preserved. Copies of all three were then created and combined to form a single **Aggregated Dataset**.


The **Aggregated Dataset** was split into a **training set** and a **test set** using an **80/20 split** with **stratified sampling** method to preserve the original ratios of three labels.

STEP:
Data
How Data

How was data collected



Pipeline - Vectorization

The next step involved embedding the natural language datasets and converting them into vector representations to enable Retrieval-Augmented Generation, a prompting technique explained here →  To ensure proper separation between training and evaluation, only the **training set** was embedded when constructing the vector databases.

One of the primary objectives was to evaluate and compare the performance of different LLMs in identifying sensitive clauses. The following three LLMs were selected for this analysis:

Model Name	Organization	Reason for Selection
GPT-4o	OpenAI	Mainstream adoption
Deepseek v3	Deepseek	Rapid emergence
Mistral 7B	Mistral AI	Lightweight

For each selected LLM, a corresponding Pinecone vector database was created using its native embedding algorithm.

STEP:

Data

How Data

Pipeline - Vectorization

Pipeline - Vectorization

However, since the DeepSeek API does not support embedding functionality as of April 2025, the GPT embedding model was used in place of a native DeepSeek embedding. Consequently, when evaluating DeepSeek during the RAG phase, data was retrieved from the GPT-based Pinecone database. The following embedding algorithms were used to construct the vector databases:

- a. **OpenAI Embedding:** *text-embedding-ada-002*
- b. **Mistral Embedding:** *Mistral Embed*

STEP:

Data

How Data

Pipeline - Vectorization

Pipeline - Vectorization - (OpenAI)

The following steps were first carried out for vectorization using GPT embeddings:

Input: Raw JSON clauses (4370 entries) with metadata flags



Vectorization Process (Python script)

- a. Defined index with **1536 dimensions** to match *text-embedding-ada-002*, using **cosine similarity** and deployed on **AWS**
- b. Created the index and connected to it
- c. Generate embeddings from policy clauses using OpenAI's *text-embedding-ada-002*
- d. Loaded the dataset from .jsonl files containing labeled policy clauses
- e. For each clause, generated an embedding and stored it along with three metadata labels:
 - 1. data_shared_with_advertiser,
 - 2. data_transferred_to_unfriendly_countries,
 - 3. data_used_for_ai_training;
- f. Batched the embeddings into groups of 50 and upserted the embeddings into the Pinecone index



Output: Searchable vector database for downstream tasks (RAG)

STEP:

Data

How Data

Pipeline - Vectorization

Pipeline - Vectorization - (Mistral)

Similarly, the following steps were followed for vectorization using Mistral embeddings. All steps remained consistent with the GPT-based process, with the exception of step (c).

Input: Raw JSON clauses (4370 entries) with metadata flags



Vectorization Process (Python script)

- a. Defined index with **1024 dimensions** to match *mistral-embed*, using **cosine similarity** and deployed on **AWS**
- b. Created the index and connected to it
- c. Generate embeddings from policy clauses by making authenticated **HTTP POST requests** to Mistral's `/v1/embeddings` endpoint using the *mistral-embed* model
- d. Loaded the dataset from .jsonl files containing labeled policy clauses
- e. For each clause, generated an embedding and stored it along with three metadata labels:
 1. `data_shared_with_advertiser`,
 2. `data_transferred_to_unfriendly_countries`,
 3. `data_used_for_ai_training`;
- f. Batched the embeddings into groups of 50 and upserted the embeddings into the Pinecone index



Output: Searchable vector database for downstream tasks (RAG)

STEP:

Data

How Data

Pipeline - Vectorization

Pipeline - Vectorization

Two examples are presented below to illustrate the structure of entries stored in Pinecone.

Showing 10 hits	
1	<div><div>ID</div><div>022063b9335a77d63976a263580c8089-400-0</div><div><div></div><div></div><div></div></div></div> <div><div>SCORE</div><div>1.0025</div><div><div>FIELDS</div><div><div>clause:</div><div>"We may share your personal information with our third party service providers such as those who provide chat features andor search engines, member support agents and te...</div></div><div><div>data_shared_with_advertisers:</div><div>true</div></div><div><div>data_transferred_to_unfriendly_countries:</div><div>false</div></div><div><div>data_used_for_AI_training:</div><div>false</div></div><div><div>policy_uid:</div><div>"022063b9335a77d63976a263580c8089"</div></div></div></div>
2	<div><div>ID</div><div>5a419731648e720fa380a7cb26db74e0-50-27</div><div><div></div><div></div><div></div></div></div> <div><div>SCORE</div><div>0.9269</div><div><div>FIELDS</div><div><div>clause:</div><div>"We may share your Personal Information in the following ways: Service Providers: We share personal information with third-party service providers who perform services on ...</div></div><div><div>data_shared_with_advertisers:</div><div>true</div></div><div><div>data_transferred_to_unfriendly_countries:</div><div>false</div></div><div><div>data_used_for_AI_training:</div><div>false</div></div><div><div>policy_uid:</div><div>"5a419731648e720fa380a7cb26db74e0"</div></div></div></div>

STEP:

Data

How Data

Pipeline - Vector Databses

Pipeline - Vector Databases

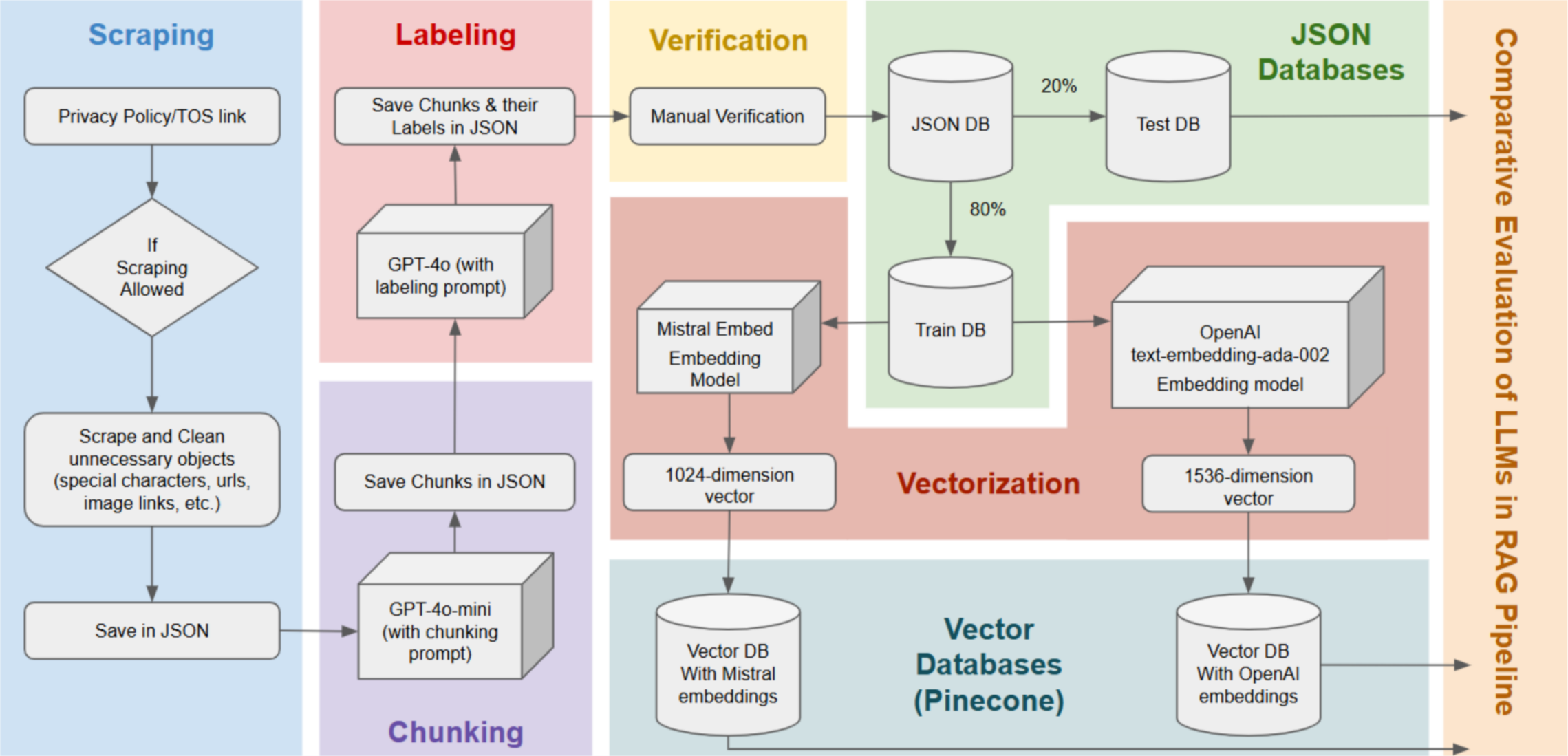
Two vector databases were constructed:

- a. Vector Database built using OpenAI embeddings
- b. Vector Database built using Mistral embeddings

These databases played a critical role in the AI experiments, where RAG was applied. Further details regarding the RAG implementation are provided in the following section.

STEP:
Data
How Data

How was data collected



STEP:

Data

3) Data

What data was collected?

How was data collected?

Project Overview

If you are particularly interested in any specific part, you can also click into it.



- 1) Objectives
- 2) Literature review
- 3) Data Preparation
 - 3.1) Collection & Scrapping
 - 3.2) Chunking
 - 3.3) Labeling
 - 3.4) Train test split
 - 3.5) Vectorization
- 4) AI Experiments
 - 4.1) Zero-Shot
 - 4.2) Static Few-Shot
 - 4.3) RAG
- 5) Summary Across All Experiments
- 6) Next steps

4) AI Experiments

A series of AI strategies were explored to evaluate the effectiveness of different models and prompting techniques in identifying sensitive information within privacy policies. The primary objective was to assess and compare the performance of three distinct approaches in classifying clauses according to the three predefined risk categories.

These approaches included:

- a. Zero-Shot Prompting
- b. Static Few-Shot Prompting
- c. Retrieval-Augmented Generation

To ensure a fair and consistent evaluation, the same **test set**, sourced and described in the Data Collection section, was used across all models and prompting strategies.

AI - Zero-Shot

Zero-Shot Prompting was first applied to establish a performance baseline for comparison with more advanced prompting strategies.

This approach was evaluated using three models: **GPT-4o**, **DeepSeek V3**, and **Mistral 7B**.

AI - Zero-Shot Statistics

Zeroshot GPT	Precision	Recall	F1 Score	Accuracy
data_transferred_to_unfriendly_countries	0.98	0.74	0.85	0.97
data_shared_with_advertisers	1.00	0.80	0.88	0.82
data_used_for_AI_training	1.00	0.48	0.65	0.98

Zeroshot DeepSeek	Precision	Recall	F1 Score	Accuracy
data_transferred_to_unfriendly_countries	0.64	0.71	0.61	0.84
data_shared_with_advertisers	1.00	0.86	0.92	0.88
data_used_for_AI_training	1.00	0.65	0.78	0.98

The **Mistral** component of the zero-shot prompting evaluation is currently in progress. The corresponding performance statistics will be updated upon completion of this portion of the work.

AI - Static Few-Shot

Next, the focus was shifted to Few-Shot Prompting. In this approach, each prompt included three static examples of manually verified and labeled clauses to guide the model's classification.

```
static_examples = [  
    {"role": "user", "content": "We operate globally and may transfer your personal information to third parties  
in locations around the world for the purposes described in this Privacy Policy. Whenever we transfer your personal  
information outside of the United States, we ensure that appropriate safeguards are in place by entering into  
agreements with recipients that include standard contractual clauses approved by relevant regulatory authorities. By  
using our Services, you acknowledge and consent to your personal information being transferred to and processed in  
countries that may have different data protection rules than your country."},  
    {"role": "assistant", "content": json.dumps({  
        "data_transferred_to_unfriendly_countries": True,  
        "data_shared_with_advertisers": False,  
        "data_used_for_AI_training": False  
    })},  
    {"role": "user", "content": "We may share your personal information with third-party advertisers and  
advertising networks to deliver personalized advertisements that may be relevant to your interests. This information  
may include, but is not limited to, device identifiers, browsing behavior, purchase history, and demographic data.  
These third parties may use cookies, web beacons, and other tracking technologies to collect information about your  
use of our Services and other websites to provide targeted advertising. You can opt out of certain targeted  
advertising by adjusting your privacy settings or following the opt-out instructions provided in our advertising  
communications."},  
    {"role": "assistant", "content": json.dumps({  
        "data_transferred_to_unfriendly_countries": False,  
        "data_shared_with_advertisers": True,  
        "data_used_for_AI_training": False  
    })},  
    {"role": "user", "content": "We may use your data, including but not limited to your interactions with our  
Services, content you create or upload, and feedback you provide, to train and improve artificial intelligence and  
machine learning models. This processing helps us enhance our Services, develop new features, and improve user  
experience. While we may use aggregated and anonymized data for these purposes, we may also process individual-level  
data in accordance with applicable laws and regulations. By using our Services, you acknowledge that your data may be  
used for these artificial intelligence and machine learning purposes."},  
    {"role": "assistant", "content": json.dumps({  
        "data_transferred_to_unfriendly_countries": False,  
        "data_shared_with_advertisers": False,  
        "data_used_for_AI_training": True  
    })},  
]
```

As with the zero-shot setup, this method was tested on **GPT-4o**, **DeepSeek V3**, and **Mistral 7B**.
The results from these experiments are presented below.

AI - Few-Shot Statistics

Fewshot GPT	Precision	Recall	F1 Score	Accuracy
data_transferred_to_unfriendly_countries	0.76	0.65	0.70	0.94
data_shared_with_advertisers	1.00	0.85	0.92	0.87
data_used_for_AI_training	1.00	0.64	0.78	0.98
Fewshot DeepSeek	Precision	Recall	F1 Score	Accuracy
data_transferred_to_unfriendly_countries	0.18	0.74	0.29	0.60
data_shared_with_advertisers	1.00	0.93	0.96	0.94
data_used_for_AI_training	1.00	0.82	0.90	0.99

The **Mistral** component of the few-shot prompting evaluation is currently in progress. The corresponding performance statistics will be updated upon completion of this portion of the work.

AI - Zero-Shot & Few-Shot Analysis

Detailed analyses were conducted and they are organized by individual label categories.

One key consideration in this experimental setting is the prioritization of **false negative** reduction.

A false negative happens when a truly sensitive clause is missed. This poses a greater risk, as users may unknowingly consent to harmful data practices.

In contrast, a false positive simply results in a benign clause being flagged, which is a safer and more acceptable error.

For this reason, greater emphasis was placed on **recall** during evaluation, ensuring that as many relevant clauses as possible were captured, even at the cost of a slightly higher false positive rate.

AI - Zero-Shot & Few-Shot Analysis

1. data_transferred_to_unfriendly_countries:

- GPT Zeroshot achieves both high precision (0.98) and high recall (0.74), making it effective at correctly identifying problematic clauses while minimizing false positives.
- DeepSeek Fewshot reaches the same recall but suffers from extremely low precision (0.18), resulting in a large number of false positives.
- However, given our **priority to minimize false negatives**, DeepSeek's high recall still makes it valuable.
- **Insight: GPT Zeroshot** is the best overall for this label: it captures as many true positives as DeepSeek Fewshot but with far fewer false alarms.

AI - Zero-Shot & Few-Shot Analysis

2. data_shared_with_advertisers:

- All models achieved a precision of 1.00.
- DeepSeek Fewshot leads with the highest recall (0.93), making it the strongest performer for this clause. It captures nearly all true positive cases without introducing any false alarms.
- DeepSeek Zeroshot follows closely with a strong recall of 0.86 and an equally perfect precision, outperforming both GPT Fewshot (recall 0.85) and GPT Zeroshot (recall 0.80).
- **Insight: DeepSeek Fewshot** is the best model for this label: it achieves near-complete coverage of true violations with no compromise in precision.

AI - Zero-Shot & Few-Shot Analysis

3. data_used_for_AI_training

- DeepSeek Fewshot again outperforms other models, achieving the highest recall (0.82), resulting in an F1 score of 0.90. This indicates it catches the majority of true violations while avoiding false positives.
- **Insight: DeepSeek Fewshot** is the winner for this label, offering the best protection against false negatives while maintaining precision.

AI - Zero-Shot & Few-Shot Analysis

Additional Analysis: **Time Cost**

- The test set contained 1,179 entries, meaning 1,179 API calls were made per model.
 - **DeepSeek**: total execution time was 9,454.64 seconds, with an average API call time of 8.02 seconds;
 - **GPT**: total execution time was 618.99 seconds, averaging 0.53 seconds per call.
- This means that, for the same number of queries, DeepSeek took approximately **15.3 times longer** than GPT to process the full test set. This finding highlights a significant difference in response latency between the two models.

AI - Zero-Shot & Few-Shot Analysis

Additional Analysis: **Monetary Cost**

- Input Token Costs:
 - DeepSeek-Chat:
 - Cache hit: \$0.07 / 1M tokens
 - Cache miss: \$0.27 / 1M tokens
 - GPT-4o:
 - Cached input: \$1.25 / 1M tokens
 - Standard input: \$2.50 / 1M tokens
- Output Token Costs:
 - DeepSeek-Chat: \$1.10 / 1M tokens
 - GPT-4o: \$10.00 / 1M tokens

Insights

- DeepSeek is significantly more affordable than GPT-4o across both input and output tokens.
- For input, GPT-4o is approximately **5x** to **35x more expensive** depending on whether a cache hit occurs.
- For output, GPT-4o is nearly **9x more expensive** than DeepSeek.

AI - Retrieval-Augmented Generation

What is Retrieval-Augmented Generation (RAG)?

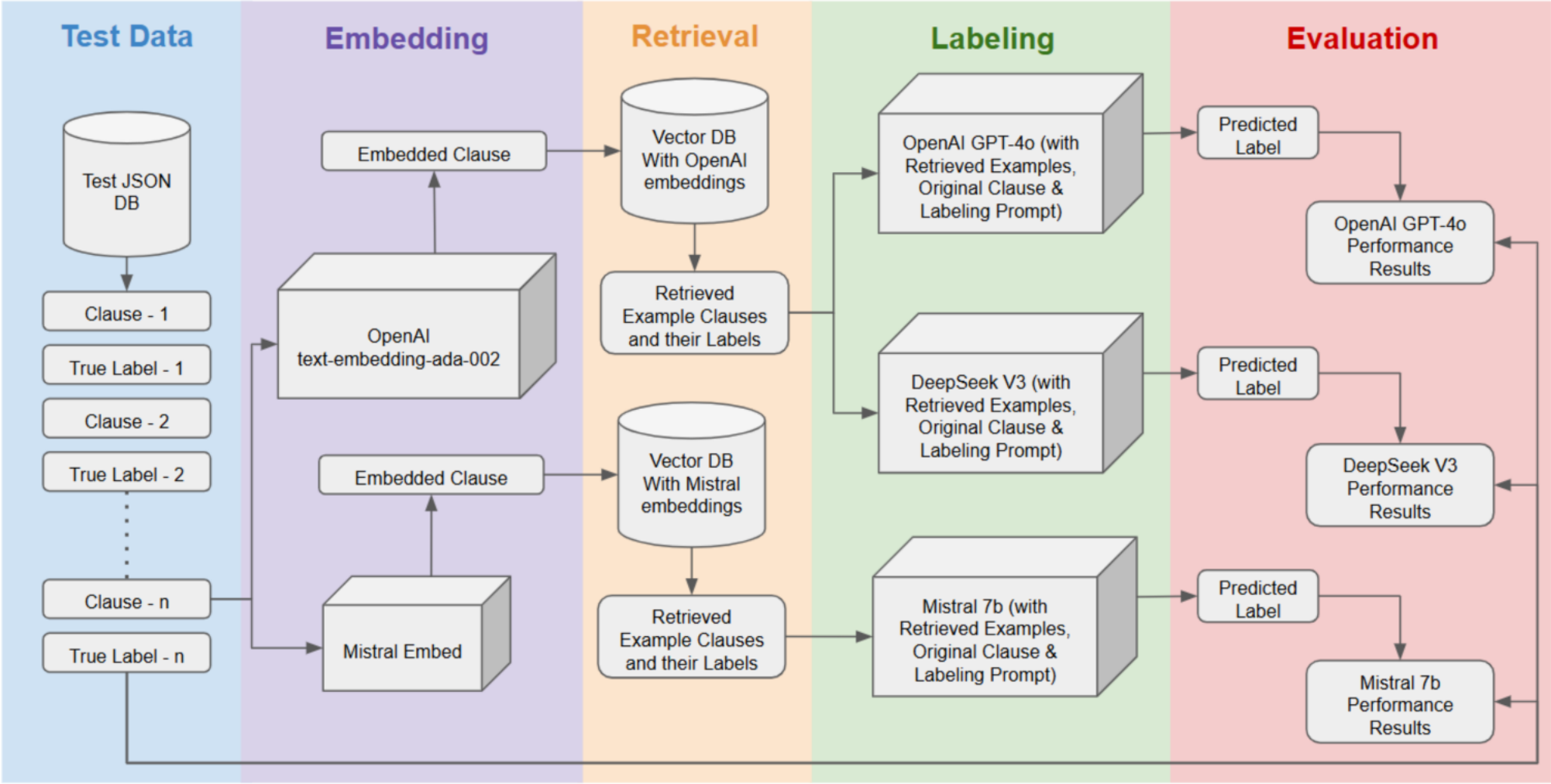
RAG is a **prompting technique** that enhances a language model's responses by incorporating external information.

Instead of relying solely on the model's internal knowledge, **RAG retrieves relevant information from an external source (such as a vector database) and adds it into the prompt.**

This allows the model to generate more accurate and context-aware responses.

In other words, RAG can be seen of as a form of **dynamic few-shot prompting**, where relevant examples are retrieved on the fly rather than manually inserted into the prompt.

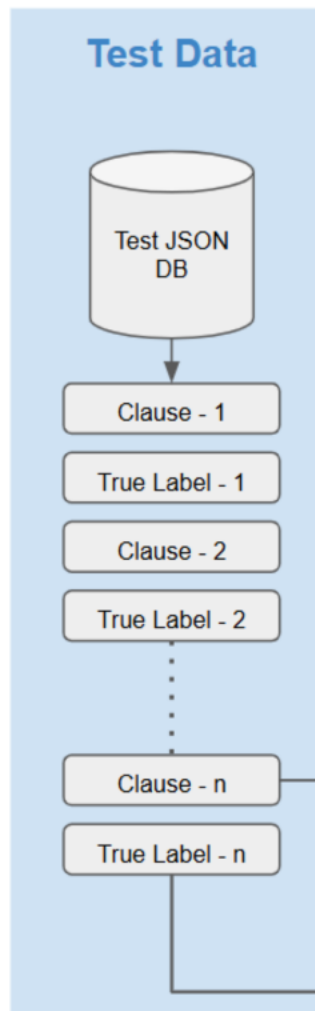
AI - Retrieval-Augmented Generation - Pipeline



AI - Retrieval-Augmented Generation - Pipeline

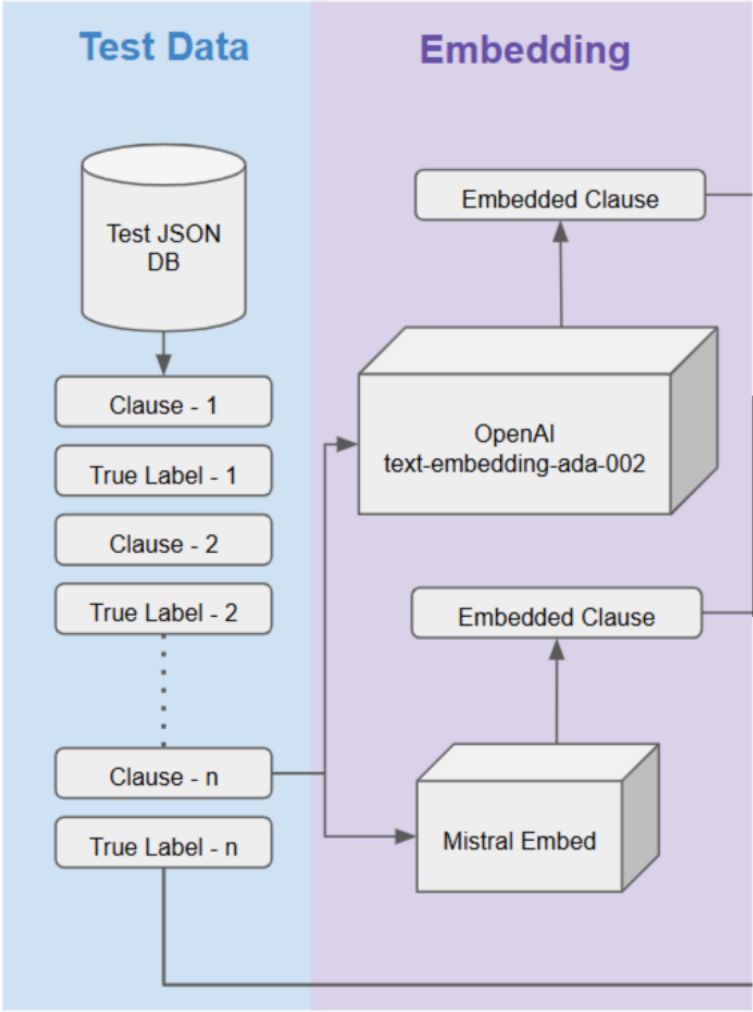
Test Data

The test dataset consisted of a total of 1,179 entries. All JSON files were first loaded from the designated directory, and the relevant entries were extracted. Only the textual content was submitted to the embedding model for vectorization, while the ground truth labels were stored separately for use during the evaluation phase.



```
def load_dataset(directory):  
    entries = []  
    for filename in os.listdir(directory):  
        if filename.endswith('.json'):  
            filepath = os.path.join(directory, filename)  
            try:  
                with open(filepath, 'r', encoding='utf-8') as f:  
                    data = json.load(f)  
                    for entry in data:  
                        text = list(entry.keys())[0]  
                        labels = list(entry.values())[0]  
                        entries.append({  
                            'text': text,  
                            'ground_truth': labels  
                        })  
            except json.JSONDecodeError:  
                print(f"Error reading {filename}")  
    return entries
```

AI - Retrieval-Augmented Generation - Pipeline



Embedding

When building the vector database (as described earlier), **two** different embedding algorithms were used: OpenAI's embedding and Mistral's embedding.

```
openai_client = OpenAI(  
    api_key=os.environ.get("OPENAI_API_KEY")  
)  
  
mistral_client = Mistral(  
    api_key=os.environ.get("MISTRAL_API_KEY")  
)
```

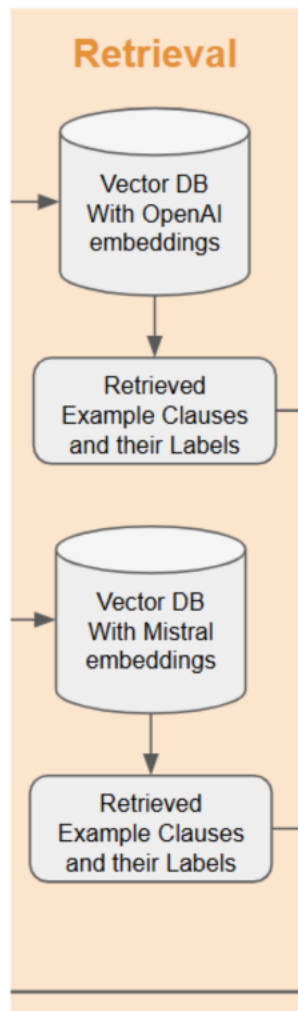
```
if embedding_provider == "openai":  
    response = openai_client.embeddings.create(  
        model="text-embedding-ada-002",  
        input=clauses  
    )  
    for i, clause in enumerate(clauses):  
        pairs[clause] = np.array(response.data[i].embedding)  
  
elif embedding_provider == "mistral":  
    response = mistral_client.embeddings.create(  
        model="mistral-embed",  
        inputs=clauses  
    )  
    for i, clause in enumerate(clauses):  
        pairs[clause] = np.array(response.data[i].embedding)
```

As a result, when querying the two vector databases, the clause of interest must be embedded using the corresponding embedding method associated with each database.

AI - Retrieval-Augmented Generation - Pipeline

Retrieval

Next, from the databases, the top k entries were retrieved from the databases, where k is an adjustable parameter (default: 3), based on their cosine similarity to the clause of interest.



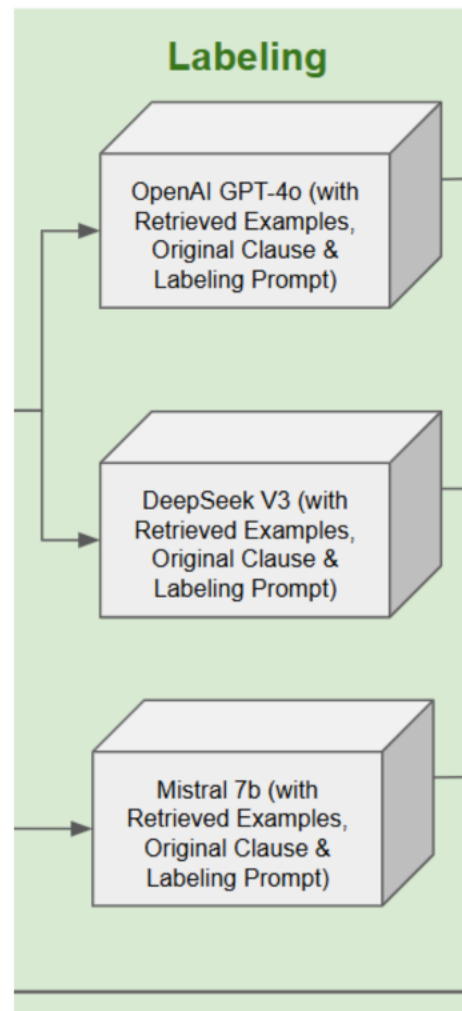
```
def retrieve_top_k(query, pairs, top_k=3):
    embedding = pairs[query].tolist()
    query_results = index.query(vector=embedding, top_k=top_k)
    # print(query_results)
    match_ids = [record["id"] for record in query_results["matches"]]
    fetched_records = index.fetch(match_ids)
    retrieved_texts = [
        fetched_records["vectors"][id]["metadata"]["text"] for id in match_ids
    ]

    return retrieved_texts
```

AI - Retrieval-Augmented Generation - Pipeline

Prompting & Labeling

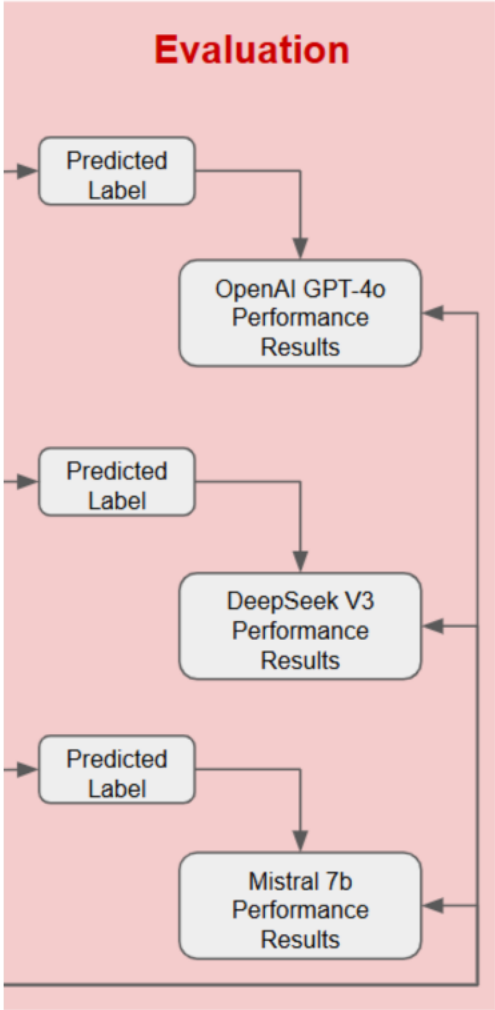
After retrieval, the top three clauses were inserted into the prompt sent to the language models, along with their corresponding ground truth labels. As these retrieved clauses exhibited high cosine similarity to the clause of interest, they were assumed to be semantically similar. By providing the language models with both the text and labels of these similar clauses, the models' ability to accurately predict the label for the target clause was expected to be enhanced.



AI - Retrieval-Augmented Generation - Pipeline

Evaluation - GPT-4o

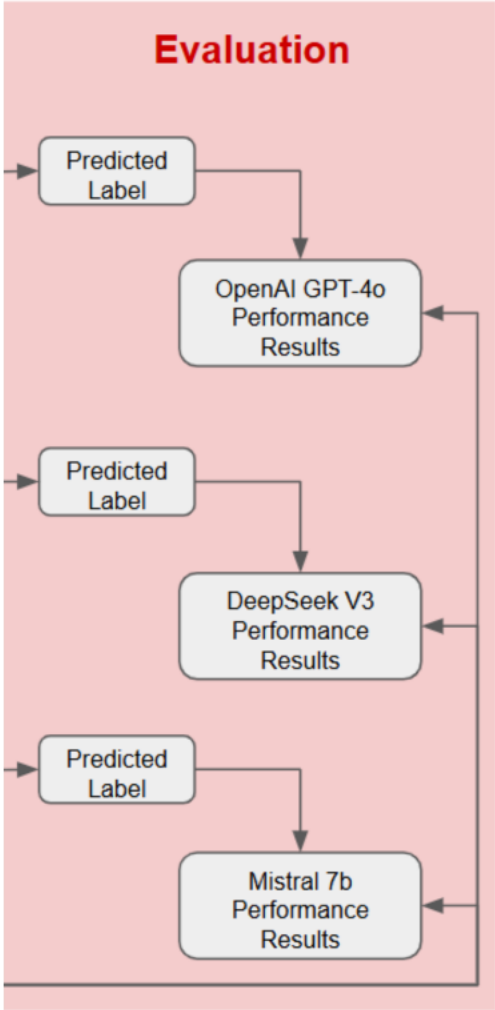
Label	Precision	Recall	F1 Score	Accuracy
data_transferred_to_unfriendly_countries	0.97	0.96	0.97	0.98
data_shared_with_advertisers	0.99	0.98	0.99	0.99
data_used_for_AI_training	0.86	0.97	0.91	0.98



AI - Retrieval-Augmented Generation - Pipeline

Evaluation - Deepseek V3

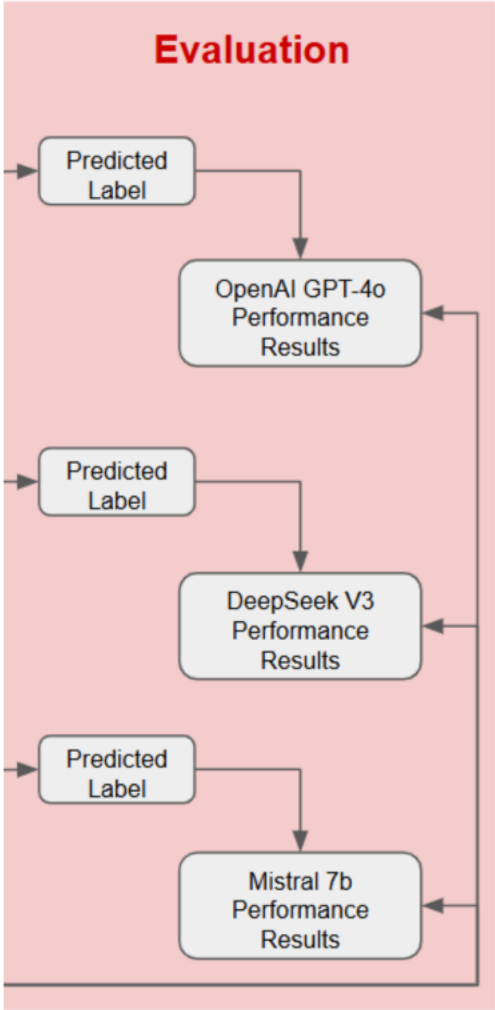
Label	Precision	Recall	F1 Score	Accuracy
data_transferred_to_unfriendly_countries	0.95	0.97	0.96	0.99
data_shared_with_advertisers	1.00	0.97	0.98	0.97
data_used_for_AI_training	1.00	0.86	0.92	0.99



AI - Retrieval-Augmented Generation - Pipeline

Evaluation - Mistral 7B

Label	Precision	Recall	F1 Score	Accuracy
data_transferred_to_unfriendly_countries	0.74	0.87	0.80	0.87
data_shared_with_advertisers	0.80	0.83	0.82	0.85
data_used_for_AI_training	0.75	0.86	0.80	0.85



AI - Retrieval-Augmented Generation - Analysis

1. data_transferred_to_unfriendly_countries

Model	Recall	Precision	F1	Accuracy
DeepSeek RAG	0.97	0.95	0.96	0.99
GPT RAG	0.96	0.97	0.97	0.98
Mistral RAG	0.87	0.74	0.80	0.87

- **Insight:** DeepSeek RAG achieves the highest recall (0.97), at some cost of Precision. Although its precision is slightly lower than that of GPT, **DeepSeek is considered the stronger performer** for this label, given the greater emphasis placed on minimizing false negatives.

AI - Retrieval-Augmented Generation - Analysis

2. data_shared_with_advertisers

Model	Recall	Precision	F1	Accuracy
GPT RAG	0.98	0.99	0.99	0.99
DeepSeek RAG	0.97	1.00	0.98	0.97
Mistral RAG	0.83	0.80	0.82	0.85

- **Insight:** GPT RAG captures the most true positives (recall = 0.98), at a tiny cost of losing a little Precision. Again, since the greater emphasis is placed on minimizing false negatives, **GPT wins this label.**

AI - Retrieval-Augmented Generation - Analysis

3. data_used_for_AI_training

Model	Recall	Precision	F1	Accuracy
GPT RAG	0.97	0.86	0.91	0.98
DeepSeek RAG	0.86	1.00	0.92	0.99
Mistral RAG	0.86	0.75	0.80	0.85

- **Insight:** GPT RAG captures the most true positives (recall = 0.97). **GPT wins this label.**

Project Overview

If you are particularly interested in any specific part, you can also click into it.



- 1) Objectives
- 2) Literature review
- 3) Data Preparation
 - 3.1) Collection & Scrapping
 - 3.2) Chunking
 - 3.3) Labeling
 - 3.4) Train test split
 - 3.5) Vectorization
- 4) AI Experiments
 - 4.1) Zero-Shot
 - 4.2) Static Few-Shot
 - 4.3) RAG
- 5) Summary Across All Experiments
- 6) Next steps

STEP:

Summary Across all models

4) Summary Across All Models

A summary across all models is presented below, with the aim of identifying the most effective model to date and informing the direction of future work.

4) Summary Across All Models

Best Recall Per Label

Label	Highest Recall	Model
data_transferred_to_unfriendly_countries	0.97	DeepSeek RAG
data_shared_with_advertisers	0.98	GPT RAG
data_used_for_AI_training	0.97	GPT RAG

GPT RAG shows up as the strongest overall performer across all models and labels. It achieves the highest recall on two out of three clause types, while also maintaining strong F1 scores and accuracy. Its performance remains consistent across tasks without significant drops in precision.

DeepSeek RAG narrowly outperforms GPT RAG on the "data_transferred_to_unfriendly_countries" label but slightly trails on the other two.

4) Summary Across All Models

Best Recall Per Label

Label	Highest Recall	Model
data_transferred_to_unfriendly_countries	0.97	DeepSeek RAG
data_shared_with_advertisers	0.98	GPT RAG
data_used_for_AI_training	0.97	GPT RAG

Meanwhile, Fewshot and Zeroshot models from earlier experiments fall behind the RAG-based models, particularly in recall.

Mistral RAG, while competitive, underperforms in recall compared to both GPT and DeepSeek RAG, with the difference especially notable on the "data_shared_with_advertisers" clause.

Project Overview

If you are particularly interested in any specific part, you can also click into it.



- 1) Objectives
- 2) Literature review
- 3) Data Preparation
 - 3.1) Collection & Scrapping
 - 3.2) Chunking
 - 3.3) Labeling
 - 3.4) Train test split
 - 3.5) Vectorization
- 4) AI Experiments
 - 4.1) Zero-Shot
 - 4.2) Static Few-Shot
 - 4.3) RAG
- 5) Summary Across All Experiments
- 6) Next steps

6) Next Step

For future improvement, **DSPy**, a framework developed by researchers at **Stanford** that optimizes prompt configuration with LLMs, was planned to be integrated into the system.

Instead of relying on manual prompt engineering, DSPy introduces a programming-like paradigm where one specifies input-output examples, and the system automatically optimizes the underlying prompt through iterative refinement.

At its core, DSPy includes a "**prompt optimizer**" engine that learns how to best **configure prompts to maximize performance** on a given task.

This removes the need for manually crafting prompts and instead treats prompt design as an optimization problem guided by data and task-specific objectives.



6) Next Step

As part of future work, DSPy is planned to be integrated into the RAG pipeline to further enhance labeling accuracy.

Specifically, the manual message construction step will be replaced with a DSPy **Chain-of-Thought** module, which will take both the retrieved context (from Pinecone) and the clause query as input.

Through this integration, **prompt generation will be automated**, enabling the system to dynamically learn and refine optimal prompt formats tailored to each clause type. This addition is expected to **reduce human intervention** while **improving performance** across various label categories.



Project Overview

If you are particularly interested in any specific part, you can also click into it.



- 1) Objectives
- 2) Literature review
- 3) Data Preparation
 - 3.1) Collection & Scrapping
 - 3.2) Chunking
 - 3.3) Labeling
 - 3.4) Train test split
 - 3.5) Vectorization
- 4) AI Experiments
 - 4.1) Zero-Shot
 - 4.2) Static Few-Shot
 - 4.3) RAG
- 5) Summary Across All Experiments
- 6) Next steps