CSC490 Assignment A2 - Data processing Pipelines

Armaan Rehman Shah - 1009641309
Boaz Cheung - 1007673607
Alice Sedgwick - 1009301355
Yuan Yu - 1008782195

October 1, 2025

1 Part One: Aspirational Datasets

Overview

Our aspirational dataset design for a Canada-focused immigration RAG chatbot is driven by three principles: provenance (link back to official sources), historical coverage (track rule changes), and embedding-readiness (clean, chunked text). Data will be scraped then ingested into **AWS S3**, and indexed in **AWS RDS (Postgres with pgvector)** for retrieval.

Because this is aspirational, we also describe datasets that do not currently exist in usable form (e.g., structured officer reasoning logs, semantically marked-up legislation). These represent our "magic wand" wish-list that would dramatically improve system performance.

Meta-Summary of Datasets

Table 1 provides a high-level overview of the datasets, their purpose, update cadence, and primary storage location.

Dataset	**Purpose**	**Update Cadence**	**Storage**
IRCC Official Pages	Core website content, guides, instructions, historical versions	Weekly/daily	AWS S3
IRCC Open Data Stats	Admissions, processing times, numeric evidence	Monthly/quarterly	AWS S3
IRCC Notices	Policy changes, program launches/closures	As needed	AWS S3
IRCC Forms (PDFs)	Application forms + OCR text	Rarely	AWS S3
Legislation	IRPA and Regulations (legal texts)	On amendment	AWS S3
FAQ Corpus	Curated Q&A pairs for tuning and eval	Iterative	AWS S3
Historical Change Log	Version diffs for rules over time	Derived daily	AWS S3
Aspirational Reasoning Logs	Officer decision rationales, anonymized and structured	Non-existent/ magical	AWS S3

Table 1: Meta-summary of aspirational datasets

Datasets

1. IRCC Official Pages

Description: Core corpus of IRCC official website content. Ideal case: machine-readable JSON feeds with semantic tags; realistic case: scraping HTML. Stored in AWS S3.

```
CREATE TABLE ircc_official_pages (
  page_id        VARCHAR PRIMARY KEY,
  url            VARCHAR,
  language       VARCHAR,
  title          VARCHAR,
  content_clean  STRING,
  chunk_id       VARCHAR,
  chunk_index    INT,
  date_published DATE,
  date_scraped   TIMESTAMP,
  version_hash   VARCHAR
);
```

Usage: Primary retrieval source, citation backbone, historical tracking. *Aspirational:* Semantic annotations for sections (e.g., ”eligibility”, ”fees”) would make retrieval smarter.

2. IRCC Open Data Statistics

Description: Operational datasets (admissions, processing times, etc.). Currently CSV/Excel, aspirationally provided in APIs with consistent schemas.

```
CREATE TABLE ircc_open_stats (
  stat_id       VARCHAR PRIMARY KEY,
  dataset_name  VARCHAR,
  year          INT,
  month         INT,
  geography     VARCHAR,
  metric_name   VARCHAR,
  metric_value  FLOAT,
  source_url    VARCHAR,
  date_ingested TIMESTAMP
);
```

Usage: Chatbot can answer ”What is the average wait time for a study permit this year?” with authoritative numbers.

3. IRCC Notices and News

Description: Policy updates and announcements. Aspirationally, these would be available as structured RSS feeds with machine-readable metadata.

```
CREATE TABLE ircc_notices (
  notice_id    VARCHAR PRIMARY KEY,
  title        VARCHAR,
  content      STRING,
  url          VARCHAR,
  date_posted  DATE,
  tags         ARRAY,
  date_scraped TIMESTAMP
);
```

Usage: Supports recency-aware answers like ”What programs launched in 2024?”

4. IRCC Forms (PDFs)

Description: Application forms (e.g., IMM 1294). Today only as PDFs; aspirationally, machine-readable JSON schemas with auto-validation.

```
CREATE TABLE ircc_forms (
  form_id        VARCHAR PRIMARY KEY,
  form_name      VARCHAR,
  pdf_url        VARCHAR,
  ocr_text       STRING,
  extracted_fields OBJECT,
  language       VARCHAR,
  date_scraped   TIMESTAMP
);
```

Usage: Allows chatbot to guide users: ”Form IMM1294 requires these fields...”

5. Legislation and Regulations

Description: IRPA and IRPR. Today in static HTML/PDF; aspirationally, linked open data with machine-readable citations.

```
CREATE TABLE legal_texts (
  law_id         VARCHAR PRIMARY KEY,
  title          VARCHAR,
  section_ref    VARCHAR,
  text           STRING,
  citation       VARCHAR,
  date_effective DATE,
  date_updated   TIMESTAMP
);
```

Usage: Authoritative grounding. Chatbot can cite sections directly.

6. FAQ Corpus

Description: Curated Q&A pairs. Aspirationally, official FAQs tagged with difficulty levels and categories.

```
CREATE TABLE faq_pairs (
  faq_id        VARCHAR PRIMARY KEY,
  question      STRING,
  answer        STRING,
  source_url    VARCHAR,
  language      VARCHAR,
  created_by    VARCHAR,
  date_collected TIMESTAMP
);
```

Usage: Helps with fine-tuning and evaluation.

7. Historical Change Log

Description: Records diffs between versions, complementing `ircc_official_pages`.

```
CREATE TABLE change_log (
  change_id      VARCHAR PRIMARY KEY,
  page_id        VARCHAR,
  version_old    VARCHAR,
  version_new    VARCHAR,
  diff_summary   STRING,
  date_detected  TIMESTAMP
);
```

Usage: Enables queries like ”What changed in study permit rules since 2021?”

Note: The `ircc_official_pages` table stores page content with a `version_hash`, which allows us to detect when something has changed. The `change_log` table then records *what* changed, storing diffs across versions. This two-tier design avoids duplication: one is for detection, the other for explanation.

8. Aspirational Officer Reasoning Logs

Description: Hypothetical anonymized logs of how IRCC officers made decisions, with structured reasoning chains. Not available in reality, but extremely valuable.

```
CREATE TABLE officer_reasoning_logs (
  decision_id    VARCHAR PRIMARY KEY,
  case_type      VARCHAR,
  anonymized_facts STRING,
  reasoning_steps STRING,
  outcome        VARCHAR,
  date_logged    DATE
);
```

Usage: Would allow chatbot to explain not only rules but also *how they are applied in practice.*

End Note: While the aspirational datasets have different schemas, they would be unified downstream via a common representation in the embeddings index (chunk_id, content, source, embedding). This ensures schema heterogeneity does not block integration.

2 Part Two: Reality Check

Table 2: Datasets

Relevant Item	Description	Commentary
Immigration and Refugee Protection Act (IRPA)	Primary legislation governing immigration, refugee claims, enforcement, and admissibility.	The IRPA forms the statutory foundation for all Canadian immigration law. It establishes key principles such as admissibility, refugee protection, and enforcement powers. Referencing the IRPA is essential for research, legal analysis, or system design to ensure models and interpretations are aligned with statutory mandates, and to accurately model decision rules and exceptions. TLDR: This is the big document that has the most basic, abstract Canadian Immigration law in it.
Immigration and Refugee Protection Regulations (IRPR)	Detailed subordinate legislation under the IRPA, covering procedures, eligibility, and compliance rules.	The IRPR defines the operational rules that officers follow to implement the IRPA. It includes procedural steps, timelines, eligibility criteria, and compliance mechanisms. Understanding these regulations is critical for interpreting how decisions are applied in practice by administrative staff and legal practitioners, ensuring consistency in enforcement and application of the law. TLDR: This is an extention of the IRPA that concretizes some of the principles stated in it for legal use.
IRCC Application Overview	Public-facing portal explaining immigration and citizenship application processes.	This portal translates statutory and regulatory requirements into actionable steps for applicants. It contains instructions, eligibility checklists, and process timelines. Analyzing this data enables linking legal requirements to end-user workflows, validating procedural models, and designing user-facing guides that reflect the practical application of the law. TLDR: This site provides a tips and tricks manual for filling out immigration applications.

Relevant Item	Description	Commentary
IRCC Program Delivery Instructions	Operational bulletins and manuals guiding IRCC officers in applying policies and procedures.	These manuals serve as a bridge between abstract law/regulation and actual administrative practice. They include instructions for officers, case handling examples, and internal interpretations. Studying these manuals is crucial for understanding decision-making patterns, modeling procedural logic, and imitating real-world adjudication processes. TLDR: This website provides concrete instructions for policy enforcement and document processing.
IRCC Website Pages (Core URLs)	A collection of official IRCC web pages (applications, programs, notices, fees, biometrics, etc.).	Captures how IRCC presents information to the public, including updates over time. Useful for research on public communication and verifying how legal requirements are operationalized. This dataset is key to connecting legal text with practical instructions received by applicants. TLDR: This site has time sensitive updates and announcements relevant to immigrants.
IRCC Forms and Guides (e.g., IMM5710, IMM1295, CIT0001)	Official application forms and accompanying guides (PDF + HTML).	These forms represent the end-user interface for applying immigration rules. Each form captures specific legal and procedural requirements, including eligibility criteria, supporting document requests, and applicant declarations. They are essential for replicating workflows and ensuring that models or systems account for procedural realities. TLDR: These are the actual forms users would have to fill out.

Relevant Item	**Description**	**Commentary**
Refugee Law Lab – Canadian Legal Data (RAD + RPD)	Bulk data of tribunal decisions from the Refugee Appeal Division and Refugee Protection Division.	Contains detailed case decisions, reasoning, citations, and outcomes for refugee claims. This dataset allows for answering personal immigration questions, evaluating reasoning processes, and testing retrieval or decision-support systems. Provides both procedural and substantive examples of Canadian refugee law applied in practice. TLDR: These datasets contain Legal cases from real life immigrants with real life immigrant problems and explains how they were handled.

References

[1] Government of Canada. *Immigration and Refugee Protection Regulations (SOR/2002-227)*. Justice Laws Website, 2002, https://laws-lois.justice.gc.ca/eng/regulations/sor-2002-227/. Accessed 24 Sept. 2025.

[2] Government of Canada. *Immigration and Refugee Protection Act (S.C. 2001, c. 27)*. Justice Laws Website, 2001, https://laws.justice.gc.ca/eng/acts/i-2.5/. Accessed 24 Sept. 2025.

[3] Government of Canada. *Apply to Come to Canada*. Immigration, Refugees and Citizenship Canada, https://www.canada.ca/en/immigration-refugees-citizenship/services/application.html. Accessed 24 Sept. 2025.

[4] Government of Canada. *Operational Bulletins and Manuals*. Immigration, Refugees and Citizenship Canada, https://www.canada.ca/en/immigration-refugees-citizenship/corporate/publications-manuals/operational-bulletins-manuals.html. Accessed 24 Sept. 2025.

[5] Government of Canada. *Immigration, Refugees and Citizenship Canada – Official Website*. Immigration, Refugees and Citizenship Canada, https://www.canada.ca/en/immigration-refugees-citizenship.html. Accessed 24 Sept. 2025.

[6] Government of Canada. *Application Forms and Guides (e.g., IMM5710, IMM1295, CIT0001)*. Immigration, Refugees and Citizenship Canada, https://www.canada.ca/en/immigration-refugees-citizenship/services/application/application-forms-guides.html. Accessed 24 Sept. 2025.

[7] Refugee Law Lab. *Canadian Legal Data: Refugee Appeal Division and Refugee Protection Division Decisions*. Refugee Law Lab, https://refugeelab.ca/bulk-data/. Accessed 24 Sept. 2025.

3 Part Three: Data-processing pipelines

3.1 Data Schemas

Scope. We persist data in two layers: (i) an S3 *data lake* with raw/clean/curated zones (JSON/Parquet), and (ii) a serving database (PostgreSQL + `pgvector`) used for semantic search. Schemas below reflect the implemented system.

A. Data Lake (S3) — Curated Column Schema

Description: Chunked, cleaned documents ready for embedding. Stored as Parquet with the following columns.

```
id:              STRING        -- unique chunk id
title:           STRING
section:         STRING
content:         STRING        -- cleaned text
source:          STRING        -- e.g., IRCC URL or source tag
date_published:  DATE          -- if available
date_scraped:    TIMESTAMP
granularity:     STRING        -- page | section | paragraph
```

B. Serving DB (PostgreSQL + pgvector)

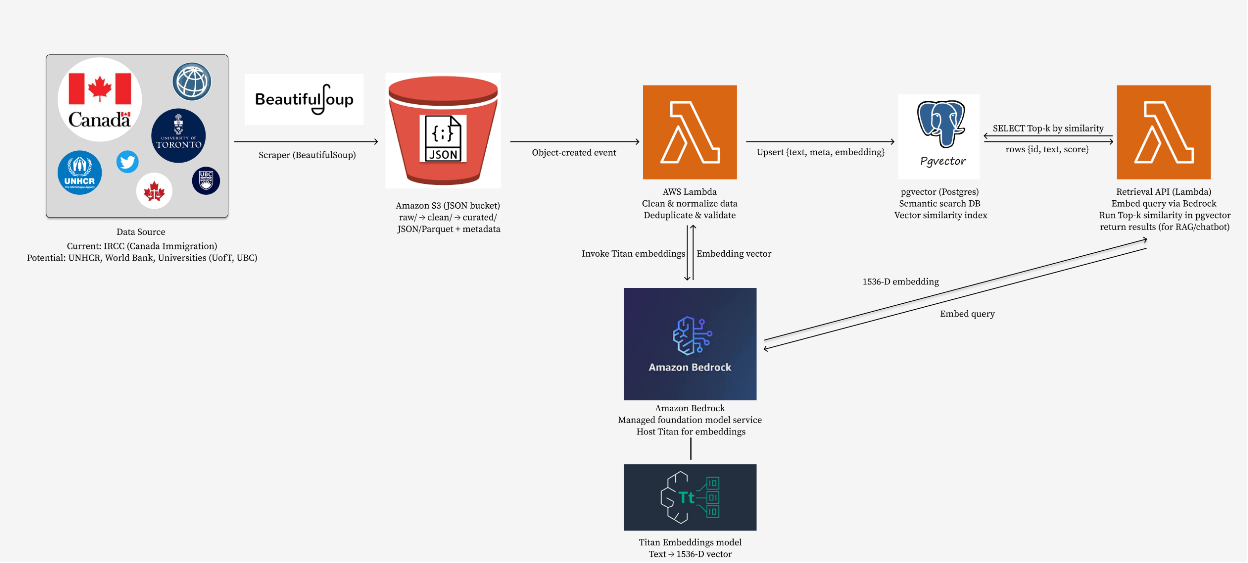
Description: Stores chunks and their embeddings for semantic retrieval.

```
CREATE TABLE documents (
  id              VARCHAR PRIMARY KEY,     -- maps to S3 curated.id
  title           VARCHAR,
  section         VARCHAR,
  content         TEXT,
  source          VARCHAR,                 -- canonical source or URL
  date_published  DATE,
  date_scraped    TIMESTAMP,
  granularity     VARCHAR,                 -- page | section | paragraph
  embedding       VECTOR(1536)             -- Titan Embeddings (via Bedrock)
);

CREATE INDEX documents_embedding_ivf
  ON documents USING ivfflat (embedding vector_cosine_ops)
  WITH (lists = 100);
```

Notes. This implemented schema is a pragmatic subset of the Q1 aspirational design. For example, `id` corresponds to Q1's `form_id/law_id`; we omitted optional fields like `version_hash` for now.

3.2 Pipeline diagrams



3.3 When the pipelines will run and for which use cases

The pipeline runs in three modes. First, an *initial load* ingests and structures historical IRCC and law data into the data lake and serving database. Second, a *scheduled refresh* runs nightly to capture new IRCC bulletins and legal updates, ensuring that recent policy changes are reflected promptly. Third, *on-demand runs* may be triggered in response to urgent announcements or structural schema changes. These executions directly support downstream use cases: (i) chatbot retrieval, where embeddings in `pgvector` are kept up-to-date for accurate policy guidance; (ii) summarization, enabled by maintaining a curated, chunked corpus; (iii) legal and policy Q&A, where users can query statutes and bulletins in natural language. Together, these runs ensure that both historical context and current developments remain accessible, reliable, and consistent.

3.4 Initial Code

See our github: https://github.com/UofT-CSC490-F2025/Immigreat

Scraping code is under /src/scraping/
Pipeline is under /src/lambda/sample.py

3.5 Next Steps

- **Additional data sources.** Extend the ingestion layer to include Statistics Canada datasets, UNHCR bulletins, or academic policy analyses, enabling richer retrieval and cross-source validation.
- **Retrieval API.** Implement the usage-facing retrieval component: a Lambda function (triggered via API Gateway) that embeds user queries with Bedrock, executes Top-k similarity search in `pgvector`, and returns structured results to a chatbot or web interface.
- **Implement raw → clean → curated zones**. At present all ingested data flows directly into a single S3 bucket; we will introduce explicit tiers to enable reproducibility, clearer debugging, and consistent lineage checks across ingestion, validation, and embedding stages.
- **Advanced data validation.** While basic cleaning is implemented, future work will integrate frameworks such as *Great Expectations* or *Deequ* to enforce data quality checks (e.g., no missing fields, consistent date formats, valid URLs).
- **Monitoring and alerts.** Add observability for scraping and transformation jobs (e.g., pipeline health metrics, failure alerts) to ensure robustness in production settings.