

Proceedings of Statistics Canada Symposium 2024: The Future of Official Statistics

Automated Document Analysis for Physical Flow Account of Plastic Material

by Oladayo Ogunnoiki and Alexandre Istrate

Release date: September 8, 2025



Statistics
Canada

Statistique
Canada

Canada

Automated Document Analysis for Physical Flow Account of Plastic Material

Oladayo Ogunnoiki and Alexandre Istrate¹

Abstract

The Physical Flow Account for Plastic Material (PFAPM) aims to enhance environmental-economic analysis by tracking plastic material flows within the Canadian economy. To help streamline this complex process, the project leveraged advanced natural language processing (NLP) such as large language models (LLM) techniques to automate sector classification and summarize the impact of COVID-19 from company reports. By integrating machine learning models and retrieval-augmented generation (RAG) methods, the manual workload was significantly reduced, improving data analysis efficiency, and leading to higher quality insights.

Key Words: NLP; RAG; LLM.

1. Introduction

1.1 Description

The Physical Flow Account for Plastic Material (PFAPM) is an environmental-economic account that estimates the flow of plastic through the Canadian economy. The account provides annual estimates by product category, resin type, and province and territory. Within this program, analysts rely on a diverse array of annual reports sourced from various companies and organizations. These alternative data sources are essential to conduct thorough research and validation activities pertinent to the account. However, the manual examination of a substantial corpus of PDF documents is laborious and inefficient, requiring a lot of time from analysts.

By using novel machine learning techniques to reduce this burden, the program stands to benefit from increased efficiency, empowering analysts to extract invaluable insights from more data sources, with greater precision and effectiveness.

1.2 Objectives

Within the context of analyzing alternative data sources on plastic material flows, consisting of diverse reports from various companies, the project aimed to leverage advanced natural language processing (NLP) techniques to achieve two key objectives:

Sector Classification: Utilize NLP algorithms to classify clients mentioned in company reports into distinct sectors, namely: Residential, Commercial, Institutional, Industrial, and Construction. This classification process enables a comprehensive understanding of the diverse stakeholders involved in plastic flow activities, facilitating a more granular and disaggregated analysis.

COVID-19 Impact Summarization: Develop algorithms to automatically summarize the impact of the COVID-19 pandemic on the plastic-related activities of companies. This involves extracting pertinent information from reports to provide concise yet informative insights into the pandemic's effects on plastic flow dynamics such as collection and recycling rates, logistical disruptions, etc.

¹Oladayo Ogunnoiki, Statistics Canada, oladayo.ogunnoiki@statcan.gc.ca; Istrate Alexandre, Statistics Canada, alexandre.istrate@statcan.gc.ca;

The ultimate goal was to construct an algorithmic pipeline capable of seamlessly accomplishing these objectives and generating documents containing the synthesized results.

2. Methodology

2.1 Data

The dataset comprised annual reports from a diverse array of companies and organizations involved in the management of plastic flows, primarily stored as text-based PDF documents. These annual reports encompass operational and financial summaries, encapsulating information pertinent to their plastic-related activities. In total, the dataset consisted of over 500 documents totalling more than 20,000 pages. Given the annual nature of the corpus, multiple reports for different years from 2015 to 2023 from the same organization were included. The sector classification task used all available years, whereas the Covid-19 impact summarisation focused on reports produced from 2020 onwards, or about 112 reports.

2.2 Analysis tools and techniques

2.2.1 Vector database

A vector database is one of the standard components used in projects leveraging Large Language Models. It is used to store text and the vector representations that captures its semantic meaning. Documents are first split into chunks, for which the vector embeddings are computed by an embedding model and are stored together in this database. The database provides not only storage, but also efficient retrieval algorithms based on semantic similarity search. A user can then retrieve from a document the chunks that have the most similar meaning to a query.

This approach solves the issue created by the limited size of the context window, instead of feeding an entire PDF document with hundreds of pages to a model, only a small subset of relevant text is used.

For this project, we stored the PDF documents in a vector database called Chroma due to the additional metadata querying capabilities it provides, allowing us for instance to limit the query only to certain years or company names.

2.2.2 Retrieval augmented generation: Context compression

Retrieval augmented generation is an Artificial Intelligence technique designed to enhance the responses generated by Large Language Models (LLM). It provides the LLM data that it did not see during its' training. It has two key components: a retriever and a generator. The retriever interacts with the vector database to find the most relevant data to a query. While the generator uses the data returned by the retriever to perform specific tasks like summarization, classification etc. Examples of RAG retrievers include ensemble retriever, multi-Query retriever, and contextual compression. The RAG retriever technique selected for this project was Contextual Compression.

Contextual Compression distills and condenses the vast amount of information available in the retriever phase into a smaller, more relevant set of information that the generator (summarizer) can use effectively (Hou et al 2024) (Lo et al., 2023). This is needed as sometimes the retriever might return information that is close enough semantically but not relevant to the task defined in the prompt. The compressor solves this by filtering irrelevant information.

2.3 Automation process

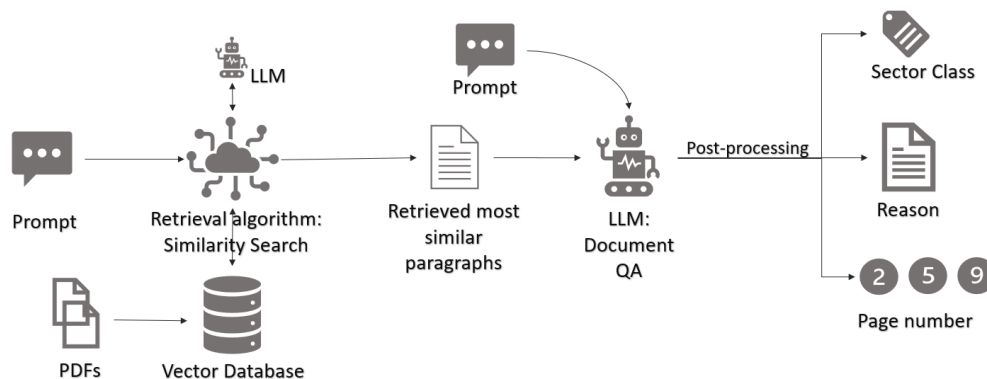
2.3.1 Sector classification

The sector classification pipeline employs a combination of retrieval algorithms, language models, and post-processing techniques to achieve accurate and efficient sector identification. The process is outlined below:

1. **Prompt Initialization:** At the onset of the pipeline, a specific prompt is formulated to query the retrieval algorithm. For instance, a typical prompt could be, "Are there any residential clients in this document?"
2. **Vector Conversion and Retrieval:** The formulated prompt undergoes vectorization, converting it into a numerical representation suitable for computational processing. The retrieval algorithm then utilizes the similarity search capability of the vector database to identify paragraphs within the document that closely align with the given prompt.
3. **LLM Response Generation:** Upon retrieval of relevant paragraphs, both the retrieved documents and the original prompt are embedded and provided as input to the LLM. The objective of the LLM is to generate an intelligible response to the initial prompt based on the context extracted from the retrieved documents.
4. **Post-Processing and Result Extraction:** The response generated by the LLM undergoes a post-processing step aimed at extracting pertinent information. This step includes identifying the sectors mentioned in the document, providing reasoning behind the identification, and noting the page number where the conclusion is derived from.
5. **Iteration and Compilation:** The entire process iterates for all documents within the dataset. Results from each iteration, including identified sectors, reasons, and page references, are compiled into an Excel document for further analysis and review.

Figure 2.3.1-1 illustrates the process of information retrieval and processing using a large language model (LLM). Starting from the left, the process begins with a prompt, followed by a retrieval algorithm that performs a similarity search in a vector database. The most similar paragraphs are then retrieved and processed through the LLM for document question answering. Post-processing occurs before the final outputs, which include sector class, reason, and page number, are determined.

Figure 2.3.1-1
A visual representation of the sector classification pipeline.



2.3.2 Sector Classification: Generative Model

In the sector classification pipeline, GPT-3.5 was employed as the core generative model, offering advanced capabilities in processing, and understanding complex textual data. Its advanced natural language understanding allowed for the nuanced detection and classification of industry-specific terminology and context, enabling the assignment of texts to their relevant sectors. The application of GPT-3.5 was key in the processing efficiency, reducing the time required for manual data categorization while maintaining a high level of Recall.

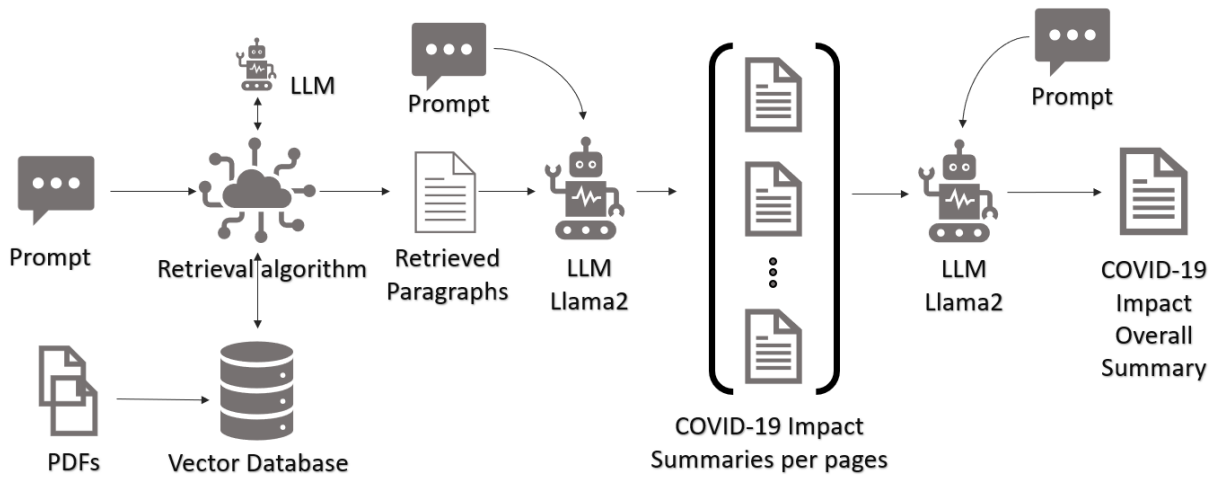
2.3.3 COVID-19 impact summarization

This section provides an overview of the pipeline illustrated in Figure 2.3.3-1 below, which is designed to generate summaries of the impact of COVID-19 within documents. The pipeline leverages retrieval algorithms, language models, and post-processing techniques to produce concise and informative summaries. The process is outlined below:

1. **Prompt Initialization:** The pipeline commences with the formulation of a specific prompt tailored to query the retrieval algorithm. For instance, a typical prompt could be, "What is the impact of COVID-19?". In our case we used the Wikipedia description of the pandemic and its impacts.
2. **Vector Conversion and Retrieval:** Following prompt formulation, the prompt is converted by the embedding model into a vector representation suitable for computational processing. The retrieval algorithm utilizes the similarity search capability of the vector database to identify pages within the document that closely match the topic of COVID-19 and the effects of the pandemic.
3. **LLM Response Generation - Per Page Summaries:** Upon retrieval of relevant pages, they are embedded together with a new prompt and provided as input to the LLM. The LLM's objective is to generate a summary of the impact of COVID-19 within the retrieved page.
4. **Summarization and Aggregation:** All the page summaries, along with a new prompt, are embedded as vectors. These embeddings are then inputted into the LLM to produce an overarching summary of the COVID-19 impact on the organization. The summary is organized by topic (Operational disruptions, financial impacts, impacts on collection and recycling rates,..), each topic having multiple bullet points, with references to the page number where the information can be found.
5. **Storage and Documentation:** The two summaries, per-page and overall impact, are compiled and stored in a Word document for easy access and reference. The analyst can therefore easily refer back to the overall summary, refer to a page mentioned in a topic and if needed look it up in the source document.
6. **Iterative Process:** The aforementioned steps are repeated iteratively for all reports produced after 2020 within the dataset to ensure comprehensive coverage and accurate summarization of COVID-19 impacts.

Figure 2.3.3-1

A visual representation of the summary generation pipeline. It finds content related to COVID-19 in the document and generates a summary of the impact.



2.3.4 Iterative development of the pipeline

Finding the right prompts to direct the LLM towards producing reports usable for the analyst, required a lot of back and forth with the subject matter experts. Because there is no metric that can automatically tell the data scientist when a summary is useful, this required extensive reviewing and feedback of the model outputs. After which, to fix the issues raised by the analyst, the data scientist had to try different prompt engineering techniques and variations at the different steps of the pipeline. This process continued until the analysts were satisfied that the reports will be a useful source of information for their research.

2.3.5 COVID-19 summarization: Generative models

Both the GPT 3.5 from Azure Open-AI as well as the open source Llama2-13B model hosted on our own servers were explored for this pipeline. The subject matter experts were given a choice on which to pursue, based on their assessment of summary quality and exhaustiveness. The Llama-2 model was chosen as it was found to better summarize the reports and organize them into topics. After this, the team pursued their prompt engineering efforts to improve the summaries quality.

The open-source model proved to be quite capable at a task like summarization, however using the Llama-2 provided distinct challenges compared to the Open-AI solution:

- Smaller context window: with only 2k tokens it limited the amount of information the pipeline can handle. Documents with many pages seem to have less precise summaries.
- Specific prompt techniques: these models are more susceptible to small changes in the prompt, with things like punctuation or the usage of capitalization improving the results. Overall, it seems more difficult to get the model to comply compared to GPT3.5, especially when the prompt gets longer because of the amount of information in the PDF document.
- Inference time: the model is significantly slower to produce outputs, which increases the time necessary to find adequate prompts.

3. Results

3.1 Assessing the quality of the sector classification with recall

Sector classification is a multilabel classification problem with 5 possible classes: Residential, Commercial, Institutional, Industrial, and Construction. An organization could collect plastic material from any of those classes, the information necessary being spread throughout the reports.

To assess the quality of the results the subject matter expert already had labelled about one hundred reports, identifying the sectors manually. To evaluate the quality of the classifier we would typically use a metric like a weighted average F1 score to assess how well a model is performing. However, in our case the performance on the labelled set seemed quite poor, after investigating this we realized that the recall was very high at 92%, whereas precision was quite low. Basically the model was predicting extra classes compared to manual processing.

We asked the subject matter expert to analyse a sample of a dozen predictions, using the justifications the model provided together with the identified pages to verify its output. In practically all cases, after verifying the source the expert agreed with the model, acknowledging that the labels were not previously identified, but should be included. Meaning that the manual labelling work missed a significant number of true positives within the documents. Given that client didn't have more resources to verify manually a sample big enough to compute a meaningful F1 score, we used the recall as a metric of classification quality. The metric will be then assessing how many times the model did correctly identify the labels that the expert also identified.

3.2 Key performance indicators

The streamlined pipelines and their outputs generated the following results:

- **Improved efficiency** in analyzing reports on the effects of the Covid pandemic:
 - Locating relevant content
 - Summarizing the impacts by theme
- **Improved efficiency** in identifying sectors of focus in the reports
- **Increase in the number of reports** that can be analyzed compared to the manual approach
- Produced **summaries for quick reference** during iterative validation work
- **Identified sector labels that were missed** via manual labeling

Table 3.2-1
An outline of the KPI's and Metrics

KPI's and Metrics	
Decrease in processing time – locating Covid impacts	70%
	(14-32 days saved)
Decrease in processing time. – sector identification	97%
	(± 40 days saved)
Recall for sector classification	92%
Extra number of files reviewed	x 3 times

As seen in Table 3.2-1, Recall was chosen as a metric because it was important to identify all possible positive instances, even if that meant possibly including false positives. The results produced based on a high recall can then be further scrutinized.

On top of these metrics, the analysts expressed the added value that these reports will bring to future projects. They will now have a reference that allows them to quickly understand an entire PDF and determine which documents contain important information, and then go directly to the relevant pages within those documents.

4. Conclusion

The project successfully achieved its objectives of creating efficiencies through reduction of manual work and helping improve the quality of analytic insights. It demonstrated that LLMs can be leveraged effectively for text classification and summarization tasks, facilitating the use of alternative data in the PFAPM.

References

- Hou, H., Ma, F., Bai, B., Zhu, X., and Yu, F. (2024), "Enhancing and accelerating large language models via instruction-aware contextual compression", arXiv preprint arXiv:2408.15491.
- Lo, K., Chang, J. C., Head, A., Bragg, J., Zhang, A. X., Trier, C., Anastasiades, C., August, T., Authur, R., Bragg, D., Bransom, E., Cachola, I., Candra, S., Chandrasekhar, Y., Chen, Y.-S., Cheng, E. Y.-Y., Chou, Y., Downey, D., Evans, R., Fok, R., Hu, F., Huff, R., Kang, D., Kim, T. S., Kinney, R., Kittur, A., Kang, H., Klevak, E., Kuehl, B., Langan, M., Latzke, M., Lochner, J., MacMillan, K., Marsh, E., Murray, T., Naik, A., Nguyen, N.-U., Palani, S., Park, S., Paulic, C., Rachatasumrit, N., Rao, S., Sayre, P., Shen, Z., Siangliulue, P., Soldaini, L., Tran, H., van Zuylen, M., Wang, L. L., Wilhelm, C., Wu, C., Yang, J., Zamarron, A., Hearst, M. A., and Weld, D. S. (2023), "The Semantic Reader project: Augmenting scholarly documents through AI-powered interactive reading interfaces", arXiv preprint arXiv:2303.14334.
- Verma, S. (2024), "Contextual Compression in Retrieval-Augmented Generation for Large Language Models: A Survey", Published in arXiv: 2409.13385 [cs.CL], <https://arxiv.org/abs/2409.13385>