



OPEN-SOURCE GENERATIVE ARTIFICIAL INTELLIGENCE FOR PUBLIC SERVICES BACKLOG FROM COMMUNITY CHALLENGE

About the community challenge

One of the most significant applications of AI technology in the public sector is retrieval-augmented generation (RAG). RAG enables AI services to be context-specific, unlocking the potential of the data that public institutions hold.

In April-May 2024, the International Telecommunication Union (ITU) organized [a community challenge](#) to encourage the development of open-source RAG solutions tailored to the public sector.

More than 30 open-source models and tools were tested, including eight text and sentence embedding models. The outcomes demonstrated the potential for leveraging open-source tools to build functional prototypes for public services. However, they also identified a number of challenges that need to be addressed to bring such solutions to production-grade level.

Building on the momentum created by the challenge and the strong interest from the stakeholder community, ITU proposes to advance this work by addressing some of the key challenges identified.

Overview of proposed approaches

The most popular approaches among the solutions tested involved the combining two different retrievers (typically a keyword-based retriever, like the BM25 function, and a vector-based retriever) with a re-ranker. The BAAI family embedding models, along with ChromaDB or Faiss, were the most popular tools for constructing vector databases and conducting similarity searches. The above approach was likely driven by the challenge's requirement to develop RAG systems capable of running on personal devices in offline mode. Combining different retrievers with a re-ranker, allows to achieve a considerable improvement in accuracy while still keeping computational demands at a comparatively low level.

A few solutions also incorporated query pre-processing techniques, such as multi-prompting, and/or context compression using local LLMs. However, these methods were found to considerably increase both computational requirements and output latency.

The Recursive Character Text Splitter emerged as the most used common chunking technique, with only a few participants experimenting with semantic chunking. The results from these semantic chunking experiments were mixed, suggesting the need for further exploration.

While these approaches proved reliable for some retrieval tasks, they faced challenges when dealing with more complex documents and queries, which are typical in the public sector. Below is a list of key issues and challenges identified.

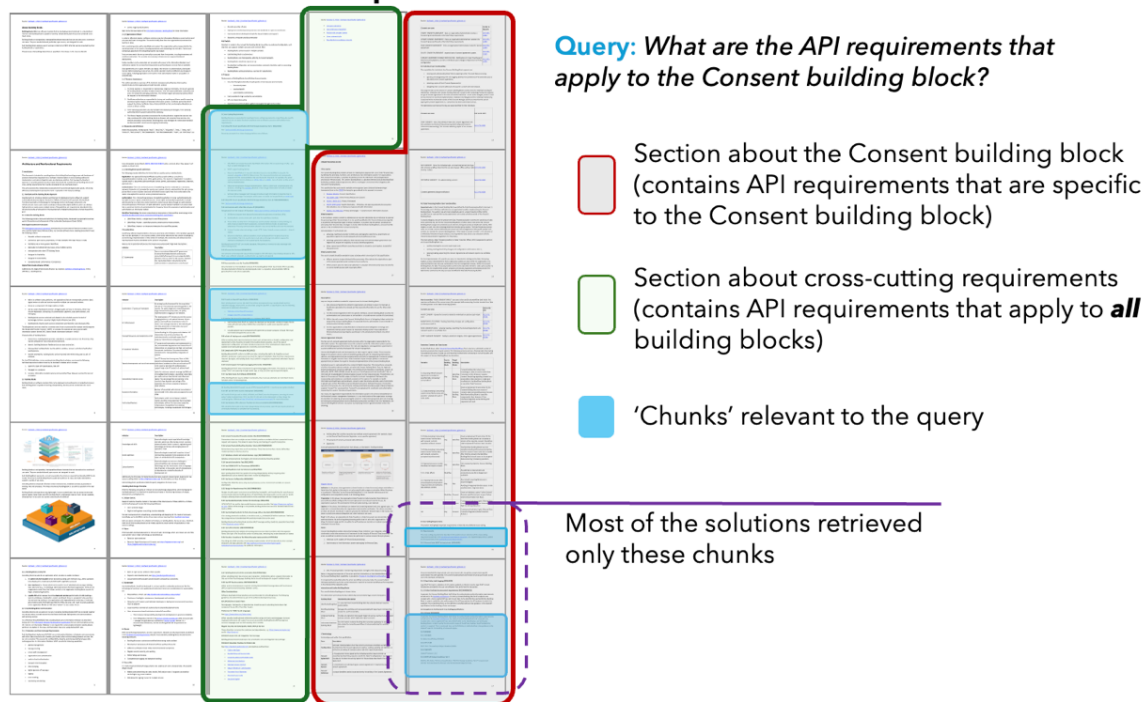
Key issues and limitations identified

1. Inability to Identify Related Content Dispersed Throughout a Document

A significant challenge encountered was the failure of the system to recognize and connect related content when it is dispersed across different sections of a document and lacks many matching keywords. This issue is particularly evident in scenarios where a query requires some level of reasoning over the initially retrieved context to establish connections with other parts of the document that may contain additional relevant details—either directly or indirectly related to the query.

This challenge is especially common in legal and regulatory documents, where certain articles or sections often refer to other parts of the document or even to entirely separate documents. The complexity and interrelated nature of these references make it difficult for the system to retrieve all relevant information based solely on keyword matching or isolated retrieval processes. The following example illustrates this issue:

Test document 4: GovStack Specs



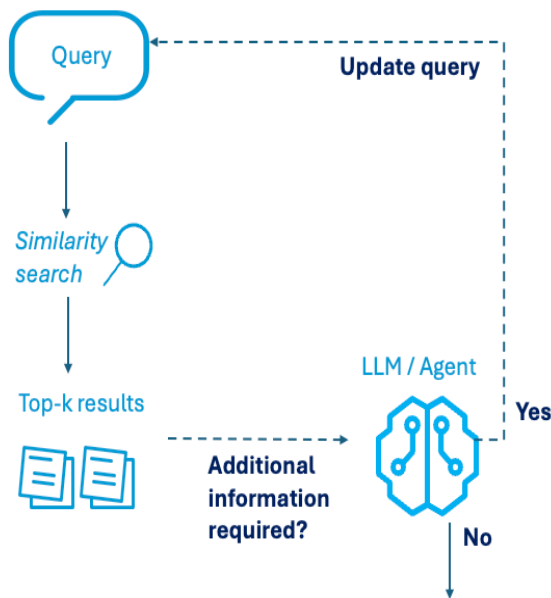
Query: *What are the API requirements that apply to the Consent building block?*

- Section about the Consent building block (contains API requirements that are specific to the Consent building block)
- Section about cross-cutting requirements (contains API requirements that apply to **all** building blocks)
- 'Chunks' relevant to the query

Most of the solutions retrieved only these chunks

Addressing this issue would likely require a more sophisticated RAG architecture capable of capturing interconnections between different parts of the context, extending beyond just keywords and basic semantic similarity. This could be achieved through enhancements at the data indexing level or by introducing agents to perform reasoning and dynamically adjust the retrieval process:

- Better indexing of contextual data, potentially through the introduction of knowledge graphs (e.g., using [Cypher/Neo4j](#)) or techniques like a variation of the [RAPTOR approach](#).
- Introduction of reasoning over retrieved content (e.g., through the integration of LLM-agents as shown below).



However, it's important to acknowledge that these approaches are computationally intensive. One of the primary challenges would be to implement these advanced methods in a way that optimizes computational resources, particularly for systems intended to run on personal devices or in resource-constrained environments.

In this context, the use of knowledge graphs may be a more feasible option, as the most computationally demanding step—constructing the graph—only needs to be performed once during the initial setup. Once the graph is constructed, the retrieval process can be more efficient, leveraging the graph's structure to make more accurate connections between related content.

2. Need to reduce the noise

A significant challenge observed across nearly all top-performing solutions was the high level of noise—irrelevant information—returned alongside relevant chunks. While these solutions excelled in terms of output completeness, they often included a substantial amount of extraneous content. On average, the tested solutions output approximately 5,500 irrelevant characters across 10 retrieval tasks.

This issue primarily stems from the use of relatively large chunks (often 1,000 tokens or more), which, while helpful in ensuring that relevant information is included, also increases the likelihood of retrieving unnecessary content. Large chunks make it difficult for the system to precisely target only the relevant parts of the document, leading to a higher noise level.

To address this, there is a clear need to explore methods that can minimize noise without compromising completeness. Potential solutions include:

- **Enhanced Indexing of Context Data:** By improving the way context data is indexed, the retrieval system could become more precise. Combining dense and sparse vector embeddings could be one of the solutions to allow the system retrieve more precise bits of information when necessary while not missing on the broader context.
- **Dynamic Chunking Techniques:** Implementing dynamic chunking techniques, which adjust chunk size and boundaries based on the context of the query, could also help in reducing noise.

Importantly, reducing noise is critical for improving the overall quality of retrieval outputs, particularly in public sector applications where precision is essential, such as for retrieval from legal documents, institutional documents and forms.



3. More precise chunking

Closely related to the issue of noise is the challenge posed by the use of the Recursive Character Text Splitter for chunking. This method often splits the text in places that are not optimal for preserving the full meaning and context of the content.

In the majority of the tests conducted, the solutions developed during the challenge returned outputs that included only portions of the relevant chunks. This is a critical issue, particularly in the context of public sector documents, where even a small omission can lead to significant misinterpretation. For instance, if a retrieved chunk contains 90-95% of the relevant information, there is a high risk that the missing 5-10% could include a crucial detail, which could dramatically impact the understanding of the document's content.

This limitation underscores the need for chunking techniques that are more sensitive to the natural boundaries of context and meaning within the text. The Recursive Character Text Splitter, while effective in some scenarios, may not be the best tool for tasks requiring a high degree of accuracy in context preservation, such as the analysis of legal, regulatory, or policy documents.

The following example from the test data illustrates this issue:

Query: *What should be the minimum size of health warnings and messages on tobacco products, and where should they be placed?*

Relevant

...**each unit packet and package of tobacco products and** any outside packaging and labelling of such products also carry health warnings describing the harmful effects of tobacco use, and may include other appropriate messages. These warnings and messages:

- (i) shall be approved by the competent national authority,
- (ii) shall be rotating,
- (iii) shall be large, clear, visible and legible,
- (iv) should be 50% or more of the principal display areas but shall be no less than 30% of the principal display areas

Retrieved

...any outside packaging and labelling of such products also carry health warnings describing the harmful effects of tobacco use, and may include other appropriate messages. These warnings and messages:

- (i) shall be approved by the competent national authority,
- (ii) shall be rotating,
- (iii) shall be large, clear, visible and legible,
- (iv) should be 50% or more of the principal display areas but shall be no less than 30% of the principal display areas

4. Lack of fine-tuned embedding models due to shortage of curated datasets

Retrieval accuracy correlates with the performance of embedding models. Larger models perform better than the smaller ones (nearly all best-performing solutions used larger-size embedding models). Fine-tuning these larger embedding models to optimize them specifically for public sector needs presents a promising strategy for enhancing retrieval accuracy.

Public sector documents frequently utilize specialized terminology and employ sentence structures that deviate from everyday language. This specialized language and structure can



pose challenges for semantic similarity search algorithms, as they may struggle to accurately interpret and match the unique linguistic patterns found in these documents. As a result, this complexity can undermine the quality and accuracy of information retrieval, leading to incomplete or irrelevant search results. For example, in UN documents, the term 'deliverables' is very often found alongside the terms 'project' and 'report'. However, most open-source embedding models fail to capture the relation between these terms:

```
Query term: deliverables

Compare term: project
Cosine Similarity nomic-embed-text: 0.28449301649656
Cosine Similarity mxbai-embed-large: 0.6816990047633592

Compare term: report
Cosine Similarity nomic-embed-text: 0.3169439197432541
Cosine Similarity mxbai-embed-large: 0.5981404731324258

Compare term: goods
Cosine Similarity nomic-embed-text: 0.3978022001712965
Cosine Similarity mxbai-embed-large: 0.612249724637433

Compare term: services
Cosine Similarity nomic-embed-text: 0.37418290999418924
Cosine Similarity mxbai-embed-large: 0.6503712207152618

Compare term: promise
Cosine Similarity nomic-embed-text: 0.47427282960418016
Cosine Similarity mxbai-embed-large: 0.5496129986876679

Displaying top 3 for each model:

nomic-embed-text:
[('promise', 0.47427282960418016), ('goods', 0.3978022001712965), ('servi

mxbai-embed-large:
[('project', 0.6816990047633592), ('services', 0.6503712207152618), ('goc
```

Importantly, the effectiveness of fine-tuning large embedding models for public sector documents is contingent upon the availability of large, high-quality datasets of curated content for training. Unfortunately, such datasets are currently scarce and not easily accessible, which poses a significant barrier to the development of fine-tuned models.

To address this challenge, there is a pressing need to create and share curated datasets tailored to public sector applications. Making these datasets available to the broader community would enable the fine-tuning of embedding models that are better equipped to handle the unique requirements of public sector document retrieval, ultimately leading to more accurate and reliable outputs.

5. Structured data integration

Public sector documents often include a blend of narrative text and structured data, such as tables, lists, or forms. RAG systems, however, tend to struggle with effectively interpreting and retrieving information from these structured data formats. This limitation can lead to fragmented or incomplete retrieval results, as the system may overlook or misinterpret key details embedded within these structured elements.



Ensuring that RAG systems can accurately handle and retrieve information from both narrative and structured data is crucial for providing comprehensive and reliable outputs. The challenge lies in enhancing the system's ability to recognize and process these structured formats without compromising the quality of the retrieval.