Bilingual Knowledge Management System for Information Retrieval from Military Policies

Charith Gunasekara[0000−0002−7213−883X], Zachary Hamel[0009−0003−6282−4697], Rohan Ben Joseph[0000−0001−8069−5874], and Feng Du[0000−0002−8489−742X]

Department of National Defence,
Government of Canada, Ottawa, ON, Canada
{charith.gunasekara, zachary.hamel2, rohanben.joseph, feng.du}@forces.gc.ca

Abstract. Navigating through the extensive and bilingual military policies of the Canadian Armed Forces (CAF) can be a complex task. To address the need for streamlined access to these documents, this paper presents the implementation of containerized artificial intelligence (AI) models to develop a bilingual knowledge management system. This system is designed to enhance information retrieval from military policies by leveraging AI, metadata tagging, text embeddings, and vector databases. Our approach involves creating a comprehensive data processing pipeline to store and retrieve military policies and documents. We utilize a question-answering pipeline that incorporates language detection, semantic search, and large language models (LLM) to process queries in both English and French. Preliminary evaluations demonstrate an accuracy rate of 87.78% for English and 73.33% for French queries. This paper builds upon our previous work on semantic search for military policies by integrating a generative AI component, thus extending beyond traditional retrieval-based methods to provide more precise and contextually relevant responses.

Keywords: Information Retrieval · Generative AI · Retrieval Augmented Generation · Military Applications

1 Introduction

Military policies encompass a wide array of guidelines detailed in handbooks, training manuals, and online resources. These policies cover diverse aspects such as leave policies, appropriate uniforms and attire for different occasions, and drill manuals that address resource allocation and operational planning. Such documents are essential as they provide structured directives and guidelines for the armed forces, ensuring organizational cohesion and informed decision-making. Despite their importance, military policy documents can be challenging to parse due to their complex and detailed nature. The specificity and intricacy of the content often make these documents difficult to read. In recent years, the growth of digital information across various fields has underscored the necessity for efficient automated information retrieval (IR) systems capable of managing large

datasets. Traditional keyword-based retrieval methods are inadequate as they are often unable to grasp the nuances of user queries or the complexities inherent in modern information systems. This limitation has driven the development of more sophisticated search technologies. Notably, the use of semantic search approaches represents a significant advancement in enhancing the accuracy and efficiency of information retrieval as presented in our previous work [14]. These technologies not only improve the capture of contextual nuances in searches but also streamline the process of accessing relevant and complex military documents.

Semantic search refers to a form of information retrieval that goes beyond simple keyword matching to understand the intent and contextual meaning behind a user's query. Semantic similarity is used as a metric that measures the likeness between the user's query and the searchbase allowing for retrieval of relevant information. A hybrid approach can provide a combination of semantic matching and keyword search to leverage their respective strengths [22]. An architecture known as the Multitask-based Semantic Search Neural Network (MSSNN) exemplifies advanced semantic search techniques by mapping queries and responses into a unified semantic space [36]. This architecture employs k-Nearest Neighbors (k-NN) for response retrieval, enhancing the accuracy and contextual relevance of the results. Furthermore, efficient top-k retrieval methods play a crucial role in modern semantic search systems. Techniques like Approximate Nearest Neighbor (ANN) Search [12], Inverted Indexing [21], Forward Indexing [19], Term Frequency-Inverse Document Frequency (TF-IDF) [1], and Posting Lists [6] are integral to enhancing search performance. ANN methods, including Locality-Sensitive Hashing (LSH) and Random Projection Trees (RP-tree), provide rapid retrieval of approximate nearest neighbors. Inverted indexing efficiently maps terms to their corresponding documents, enabling quicker access to relevant information. Similarly, forward indexing aids in the calculation of relevance scores by storing document IDs associated with specific terms. TF-IDF ranks documents by the importance of terms within them, prioritizing content that best matches the search queries. Finally, posting lists are utilized to store and efficiently retrieve documents containing each indexed term, ensuring that users can access the most relevant information in a rapid manner.

Traditional IR has been a prime focus in the understanding and enhancement of fast retrieval algorithms, especially in the context of document retrieval. Research has explored methods like binary search, hash tables, and B-trees, known for their efficacy in quickly pinpointing relevant documents within expansive collections [23]. However, conventional IR approaches such as TF-IDF often fall short in accurately grasping the semantic relevance of documents to user queries, primarily due to their reliance on exact keyword matching. As a response to these limitations, BM25 [33] has emerged as a significant improvement over these traditional methods. It refines the use of term frequency by incorporating additional parameters like document length normalization and term saturation. These improvements enable BM25 to offer a more context-sensitive method of document ranking, thereby increasing retrieval accuracy. While BM25 marks a notable

progress over TF-IDF, it still does not fully encapsulate the semantic meanings within documents, thus positioning it as a precursor to more advanced models.

The introduction of transformer-based architectures [41] has founded a new era in information retrieval, highlighted by the development of embeddings specifically tailored for IR tasks. This new era is defined by transformative models such as BERT and SBERT, have significantly altered the landscape of information retrieval and reranking [10][32]. These models leverage deep learning to understand and process the complexities of language, enabling a more profound semantic search capability. A prime example of the advancements in modern IR systems is ElasticSearch [18], which utilizes modernized indexing and querying capabilities. At its heart are document stores that not only enhance the efficiency of searches but also ensure scalability. ElasticSearch represents a pivotal advancement in the advancement of semantic search, setting a new standard for the capability and reach of information retrieval systems.

Recent advancements in IR have led to the emergence of several commercial Large Language Model (LLM) solutions, streamlining end-to-end approaches to meet diverse user needs effectively. Microsoft's QnA Maker [26] exemplifies this trend by providing a user-friendly platform for building AI-powered question-and-answer systems. It utilizes machine learning algorithms and NLP to enable the creation of intelligent chatbots that respond to user queries in real time. Similarly, Amazon's cloud-based services include Amazon Lex [2], which allows developers to build conversational interfaces using deep learning models. Google Cloud's Dialogflow [13] further complements these offerings by providing robust tools for developing AI-driven chatbots. Features such as intent recognition, entity extraction, and context management enable developers to create virtual assistants that understand and respond to user queries effectively across multiple platforms. Additionally, OpenAI's ChatGPT [28] has significantly contributed to modern IR with its commercial APIs, especially with models like GPT-3 and GPT-4. These APIs, accessible to the general public, offer powerful NLP capabilities, enabling developers to integrate sophisticated language understanding into their applications seamlessly. Together, these modern IR solutions represent a significant leap forward in AI-based information retrieval, providing organizations with customizable tools to enhance communication, streamline workflows, and improve user experiences. Additionally,Integrating semantic search methods with knowledge bases (KBs) enhances conversational NLP and information retrieval systems by addressing key challenges like hallucinations, answer completeness, and adaptability to new queries [4]. The KnowledGPT framework has successfully bridged LLMs with KBs, improving domain-specific retrieval effectiveness [42].

For military organizations that handle sensitive data, the use of commercial APIs that involve the transfer of data can introduce significant security and privacy concerns. Hence, it is essential to implement on-premise LLM solutions that operate within an organization's own infrastructure. This method safeguards data and enhances control over the handling of sensitive information. Tailoring these models through custom training to recognize organizational-specific lin-

guistic details ensures they meet operational demands effectively. This process includes distilling complex tasks into focused, domain-specific applications using a precise, narrow AI strategy to address particular challenges within the organization.

Several strategies exist for enhancing LLMs with contextual knowledge to reduce the likelihood of producing inaccurate or irrelevant responses. By drawing on a variety of studies and methods, a detailed strategy for embedding knowledge augmentation techniques into LLMs can be developed [3]. This involves identifying appropriate domain-specific knowledge sources such as knowledge bases, ontologies, or specialized corpora that match the application's domain. Following this, methods for extracting and formatting this knowledge for LLM compatibility must be determined. This could include building knowledge graphs [17], linking entities, or performing semantic annotation [27] to ensure the knowledge is organized and semantically rich. Finally, once the knowledge is prepared, effective integration into LLMs can be achieved through techniques like fine-tuning with knowledge injection [5] [44] or employing techniques such as Retrieval-Augmented Generation (RAG)[20][15]. Additionally, other strategies might combine elements of both generative and retrieval approaches in unique configurations that differ from RAG's specific methodology[31][9]. These might involve varying the interaction between the retrieval component and the generative model, or integrating additional computational layers.

Motivated by the need for secure, accessible, and cost-effective IR solutions, particularly for environments like the military systems, a containerized retrieval framework has been developed in our prior work [14]. This framework integrates question-answering algorithms, passage selection, and document retrieval capabilities. In prior work, an IR system tailored for the military dress manual was created using SQL databases and sentence transformer architecture was used with bi-encoder[41] and cross-encoder[32] models for text encoding. The bi-encoder model conducted top-k retrieval, identifying the most relevant passages through semantic search, while the cross-encoder re-ranked these results. However, this system was confined to English language only, relied solely on retrieval-based question answering, often producing lengthy responses from policy documents and lacked generative capabilities, limiting it to existing document retrieval. Despite these limitations, the system achieved an 85.4% accuracy.

Building on our previous work [14], we introduce an enhanced Knowledge Management System (KMS) while addressing previous limitations. A key improvement is the implementation of bilingual functionality in both English and French, significantly enhancing the system's usability and accessibility, addressing the Government of Canada's bilingualism requirements. Additionally, the system now processes multiple policy documents, broadening its data source variety. A major innovation is the introduction of a generative AI component to our previously retrieval-only approach redesigning as a Retrieval Augmented Generation (RAG) system which is containerized within a secure environment, facilitating concise and targeted responses rather than returning entire passages from target policies, thereby providing users with more precise information. We also

utilize a "source of truth" approach with the generative AI response to help users verify the legitimacy of the responses. Containerization also ensures straightforward deployment across various infrastructures, maintaining both scalability and security. To overcome embedding model limitations related to context length, a chunking engine has been developed, allowing for the processing of embeddings in manageable sizes suited to our models. Furthermore, we have transitioned to using a Vector Database for data storage, which enhances retrieval efficiency and overall system performance. By separating the KB from the generative AI component, we reduce the likelihood of hallucinations and improve both the customizability and accuracy of responses. These advancements ensure that our system can scale to accommodate additional policy documents and deliver accurate, relevant information across multiple domains and languages.

2 Data Processing Pipeline

Our data processing pipeline, detailed in Figure 1, follows a four-step process designed for the efficient organization and structuring of data to facilitate easy access and analysis. We begin the process by collecting source data from military policy documents. This type of raw data is often lengthy and written in overly descriptive language. To make the data manageable, it is segmented into smaller pieces with a suitable length for NLP using a chunking engine. Next, we applied metadata tagging to each text chunk to enrich it with relevant context and information. These processed chunks are then stored in a vector database, a database optimized for the storage and retrieval of textual data. Vector databases employ vector embeddings to represent text in a high-dimensional space, enhancing the efficiency of similarity searches and retrieval. The architecture of vector databases supports large-scale text data handling, ensuring both scalability and performance.

2.1 Data Collection and Cleaning

The KMS utilizes open data available in the public domain from the Government of Canada website [35]. The scope of this research includes the text corpus of three policies (Table 1) - the Canadian Armed Forces' leave policy, dress manual, and drill manual.

Table 1. Policy Sources

Military Policy Name	Source
Leave Policy	https://www.canada.ca/en/department-national-defence/corporate/policies-standards/leave-policy-manual.html
Drill Manual	https://www.canada.ca/en/services/defence/caf/military-identity-system/drill-manual.html
Dress Manual	https://www.canada.ca/en/services/defence/caf/military-identity-system/dress-manual.html

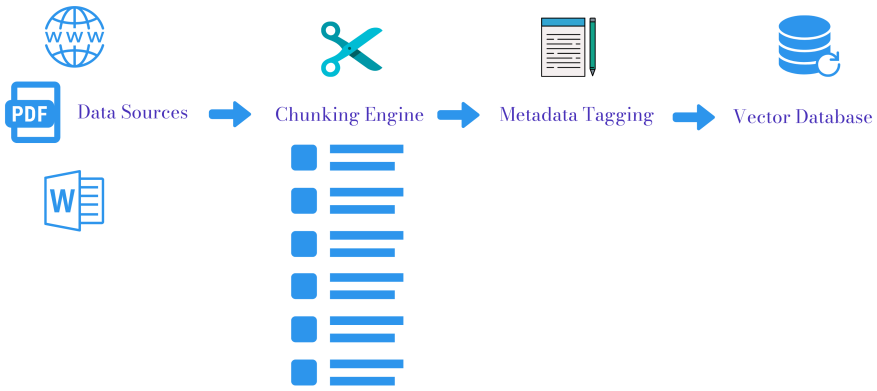

Fig. 1. Data Processing Pipeline

Both the English and French versions of the text corpus were scraped. Using Beautiful Soup [37], we targeted the index of each page to systematically download data from every link within the specified sections. This process required navigating the HTML structure of each page, identifying and retrieving data from the linked sources. Through recursive parsing of the content linked within the indices, we successfully extracted the data from all accessible sources in the designated sections.

2.2 Chunking engine

A primary constraint of sequence-to-sequence language models is their restricted context window[5], which becomes particularly problematic when dealing with lengthy documents like military policies. To overcome this, we developed a chunking engine designed to divide text of any length into manageable, contextually relevant segments. This engine plays a vital role in the data preprocessing phase, ensuring that the input data remains within the context length constraints of the language models used for text analysis.

Structure-aware chunking[7] and generative language models have been used to enhance the computational understanding of the underlying structure of web content. This approach not only facilitates the creation of concise and informative summaries tailored to user intent but also improves the overall efficacy of web page summarization. By leveraging chunking techniques alongside generative

language models, we facilitate more efficient information intake and retrieval[30], optimizing the user's experience with digital content.

Our approach utilized a preprocessing technique known as the 'Maximum Character Splitter' method. This method divides texts that exceed a character limit while prioritizing the preservation of natural sentence boundaries. It adheres to predefined separator markers such as tabs and newlines. The threshold was set to 1000 characters, allowing for a 100-character overlap. This consideration ensures that the coherence and original semantic integrity of the segmented texts are maintained.

2.3 Embeddings

To facilitate information retrieval, we employed text embeddings. Embeddings are paramount to the performance and usability of the chatbot by representing the text as a vector within a multidimensional space. Through embeddings, text is transformed into vectors where each word is mapped to a vector of real numbers within a high-dimensional space. Each dimension within this space corresponds to a unique characteristic of the text, effectively capturing its semantic properties [10].

By representing text as vectors within a multidimensional space, we enabled the possibility to search for semantic similarities among words. This capability allowed for the matching of sentences by meaning or category, rather than requiring exact or partial text matches to the policies. Such semantic matching is a key reason users can ask a chatbot questions that do not have to be partial/exact match to the policies. This alignment of embeddings in a multidimensional plane facilitates highly accurate searches, allowing the retrieval of pertinent passages for subsequent processing phases.

The embedding model selected for this project was Instructor-XL [39]. Instructor-XL demonstrates robust performance in the Massive Text Embedding Benchmark (MTEB)Benchmark, ranking sixth among comparable models. We specifically chose this model for its superior re-ranking capabilities right out of the box, combined with its compactness and remarkable speed making Instructor-XL an optimal choice given our requirements.

2.4 Vector Database

Our database architecture utilized a vector database to store embedded data which emulates a standard SQL database mode. Vector databases utilize vector embeddings to encode textual information in a high-dimensional space, enabling efficient similarity search and retrieval operations. The architecture and design of Vector databases are tailored to support large-scale text data storage and retrieval, ensuring scalability and performance. Unlike traditional SQL databases that handle various structured data types with rows and columns, Vector databases are tailored for efficiently managing high-dimensional numerical vectors, making them ideal for data that carries complex features like semantics or image characteristics[11]. Vector databases specialize in semantic

searching and similarity retrieval, optimizing searches for vectors or embeddings—particularly beneficial for recommendation systems and natural language processing tasks[43].

2.5 Metadata Tagging

An additional column is included dedicated to storing metadata in plain text. Metadata captures essential details such as source, language, and context. This design consideration enhances our data categorization and retrieval process, allowing for a more practical approach than embedding metadata directly within the primary English content. The metadata fields used in our project include:

- english_url: This tag pinpoints the exact English URL, facilitating precise location of the manual, chapter, and section relevant to the English text.
- french_content: This field stores the official French translation, which is available on every government website. It enables our system to process questions in French and deliver accurate responses in the same language.
- french_url: Similar to the english_url, this field holds the URL for the French content, assisting in identifying the exact origin of the data.

Linked URLs guide users directly to the specific webpage section from which information was retrieved while simultaneously highlighting the relevant passage.

3 Methodology

3.1 Language Detection

The KMS's Question Answering pipeline begins with language detection. This initial step determines whether user inquiries are in French or English. If the query is in English, the process proceeds to Section 3.2 as illustrated in Figure 2. If the query is in French, the process continues to Section 3.3 as illustrated in Figure 3. This ensures that the appropriate steps are taken based on the detected language, streamlining the overall question-answering workflow.

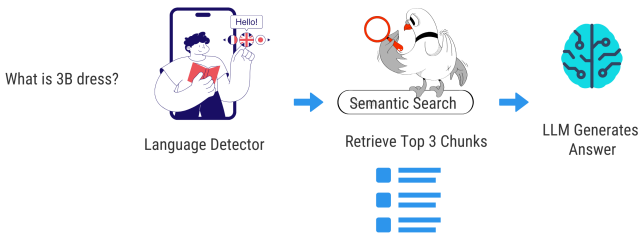

Fig. 2. Question Answering Pipeline - English

To effectively serve both French and English queries with equal proficiency, we implemented a language detection system to determine whether the inquiry was in English or French. We selected the xlm-roberta-base-language-detection [8] model for its rapid inference capabilities and efficiency, attributes stemming from Roberta-based architecture [16], and its minimal resource requirements. Given that our system currently supports only French and English, we disregard predictions in other languages, focusing solely on comparing the likelihood scores for French and English. This comparison helps us mitigate the common issue of inaccuracies with short sentences, a challenge for language detection models, which often struggle to accurately identify language from a limited number of words.

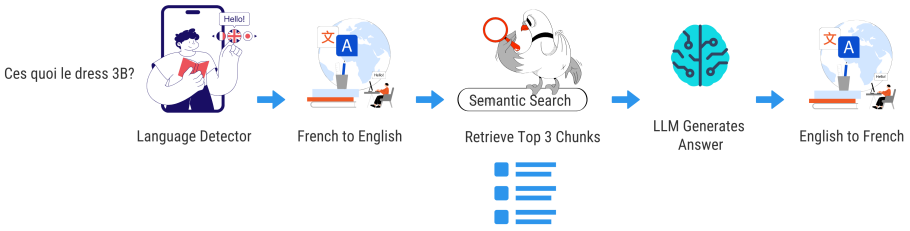

Fig. 3. Question Answering Pipeline - French

3.2 Question Answering Pipeline – English Language

Our process for generating answers deviates from the process for information retrieval by utilizing a separated knowledge base. As shown in Figure 2, we first embed the user's question and use semantic search within the vector database to retrieve the three most relevant passages. These passages are identified based on their similarity to the embedded question, ensuring they are semantically aligned. Once the most relevant passages are retrieved, they are parsed to produce contextually relevant and well-phrased answers.

Semantic Search We employ semantic search within the vector database to shortlist the top k passages, using cosine similarity, a mathematical function that measures the alignment of two vectors in terms of direction. In our process, we first embed the user's question and then compute the cosine similarity between this embedded user question and the candidate passages from the vector database. Cosine similarity outputs a normalized score between -1 and 1, where -1 indicates no semantic similarity and 1 denotes an almost identical semantic match. This metric is particularly effective in the context of LLMs, where

embeddings are situated in high-dimensional spaces, allowing cosine similarity to capture semantic relationships accurately without being influenced by vector magnitude [34]. We chose a top-k retrieval strategy, setting k to 3, which strikes a balance between gathering a sufficient number of passages to generate an informative answer and maintaining computational efficiency. We prioritize passages with the highest similarity scores for retrieval. To facilitate this, our chosen embedding model, Instructor-XL, is employed to identify and retrieve the top three passages that exhibit the strongest cosine similarity.

Generating the Final Answer After searching the vector database and identifying the relevant embeddings, we generate a prompt that allows us to transmit an instruction, a query, and the resultant embeddings to a instruction-tuned LLM version of Llama 2 [25].

An issue with LLMs is that they often generate text that is grammatically and semantically correct but factually inaccurate or irrelevant. This issue, known as hallucination [40], presents significant challenges, particularly because these inaccuracies are not easily detectable unless the user knows the correct information. In environments where precision and adherence to rules is critical, such as the military, the consequences of hallucinated content as truth can be severe.

To counteract this, we implemented a system that uses instruction-based prompts tied to the top-k passages from a vector database. Initially, we fine-tuned the LLM to answer queries only when the user's question matched the provided embeddings. If a query was unrelated to the available information, the model was instructed to respond with "I do not know," thereby avoiding the generation of irrelevant or fabricated content. This measure not only minimized our fine-tuned LLM from making uneducated guesses but also set a clear guideline for its responses. To achieve this, the beginning of our prompt begins with, "You are a question and answering bot, you will respond to the user's query based solely on the given facts. If the user's query and the given facts are irrelevant or out of context, then say 'I do not know'." This method ensures that the LLM operates within a defined scope and can be easily adapted for future requirements.

In addition, we refined the model's generation settings by reducing the temperature. Temperature affects the normalized probability of the next word chosen within the LLM. Decreasing the temperature significantly decreases randomness and creativity in the responses. This adjustment ensures that the model adheres strictly to its instructions and relies on its knowledge base, thus minimizing the chance of hallucinations. We found that a temperature of 0.1 is optimal for most scenarios, providing enough flexibility for the model to include helpful explanations when appropriate that follows a source of truth approach.

Furthermore, we set the top sampling rate to 50, allowing the model to consider a sufficient number of token options before finalizing a response. This setup is particularly beneficial in complex tasks like policy drafting, where a broader token selection reduces the amount of "I don't know" responses and enhances the quality of the output. We also modified the top-p value to 0.5 to ensure a balance in the consideration of tokens, allowing the model to remain reliable while still

capturing a variety of potential responses. By setting the p-value to 0.5, instead of considering all possible next words equally, the model only considers a subset of options. This subset is chosen based on the probability distribution of the next possible words. The "top-p" value determines the cumulative probability threshold for selecting these options.

By integrating both the first and second strategies into our LLM, we significantly enhanced its ability to generate precise answers to user queries. The model responded solely based on the facts (policies) presented to it. For this particular chatbot, we synthesized the search results (text embeddings) with the LLM's responses and the associated metadata of the embeddings. These compiled results were then forwarded to integrate into the user interface(UI).

3.3 Question Answering Pipeline – French Language

If the detected language in section 3.1 is French, we include additional translation steps highlighted in Figure 3. This is particularly more challenging than processing English questions given there is a lack of models trained with Quebec French data, the dialect predominantly used in Canada. While it is feasible to enhance these models by training them on a corpus of Quebec French collected from official Canadian and Quebec websites, such training requires significant hardware investment [38].

We have found that adding translating steps to the beginning and the end of the English pipelines and processing them in English, as shown in Figure 3, is an efficient way to process the French questions.

French to English Translation To effectively translate the French query to an English query we utilize a translation model named SeamlessM4T [24]. This model is trained on a similar corpus to No Language Left Behind (NLLB) and is extremely proficient in a neutral standard French dialect. Unlike many models that prioritize Parisian French, SeamlessM4T adopts a balanced approach, bridging the nuances between France and Quebec dialects. This distinction in syntax and conversational style makes SeamlessM4T's version of French particularly appealing to French-speaking Canadians, who are more familiar with the Quebecois variant of the French language.

Semantic Search In this step, as the query has been translated from French to English, it undergoes the same retrieval process as outlined in section 3.2. Here, we extract the top k passages from our vector database, which are then passed to the LLM to generate the final result.

Generating the Final Answer The process of generating the LLM answer followed the same methodology as described in Section 3.2. Initially, the top k passages were retrieved. These passages, along with our instructional prompt, were then provided to the LLM. The LLM analyzes the retrieved passages and formulates a response intended to assist the user.

English to French Translation The final step in producing the answer involves translating our LLM-generated English response back into French and retrieving the French policy from the metadata tags. To ensure the accuracy of the French translated answer, we divide the English response into individual sentences. This approach aligns with the NLLB corpus, which SeamlessM4T is primarily trained on, consisting predominantly of single sentences. Due to the limited presence of multi-sentence data in the training corpus, we split the text by sentences and batch the translation requests together.

3.4 User interface

The user interface (UI) was developed using Dash [29] as the primary frontend framework, chosen for its capability to facilitate rapid development while abstracting HTML and CSS code. Additionally, Dash was employed as the backend API to ensure seamless integration with the UI, thus maintaining consistent communication between the frontend and backend.

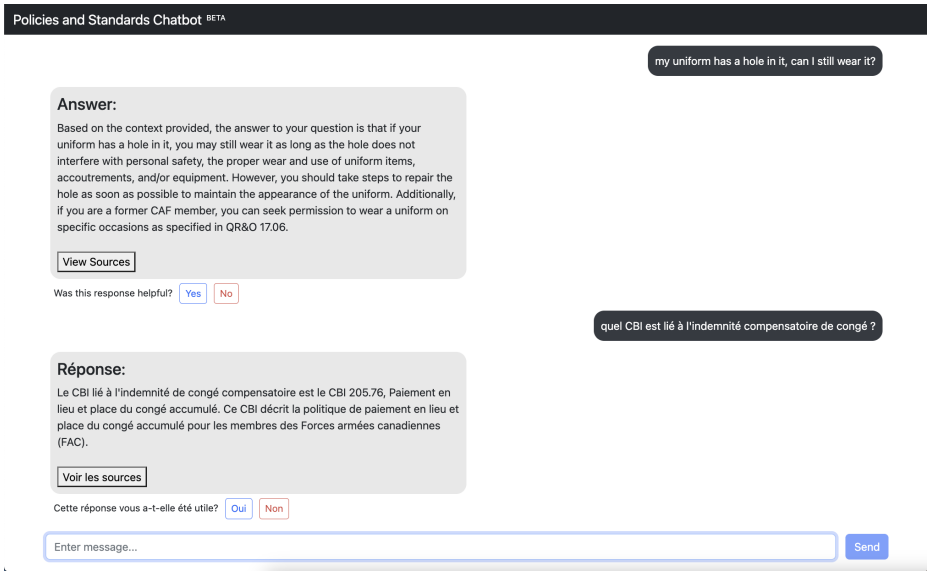

Fig. 4. User Interface

Recognizing the need for accuracy in military contexts, we adopted a "source of truth" approach, which relies on a designated, authoritative data source deemed as the most definitive and reliable reference for specific information within an organization. For our purposes, this source comprised the exact location of the policy documents the answer was generated from.

Implementing a source of truth with the UI is vital for preserving data integrity, minimizing errors, and providing a stable foundation for decision-making

within any organization. This method helps eliminate the potential for confusion or conflicting data, which can occur when multiple sources claim to provide the correct information. Additionally, it enhances transparency by allowing users to trace each step taken by our KMS, enabling them to verify the origin and application of each policy segment. This transparency is particularly beneficial for users unfamiliar with the location of specific policies, as it directs them precisely to the relevant sections, thereby facilitating easier access and understanding of the information used to generate responses.

We introduced source of truth as an additional section that remains hidden by default but can be expanded upon user interaction. This section reveals the specific chunk of text that was provided to the LLM and includes a hyperlink to the corresponding extract from the policy website. This strategy ensures data consistency and integrity across various applications, departments, and processes. In the event of conflicts or discrepancies, the source of truth can be consulted to resolve issues and maintain a single accurate version of information.

Furthermore, the metadata tagging facilitated the inclusion of URLs for the source as shown in Figure 5.

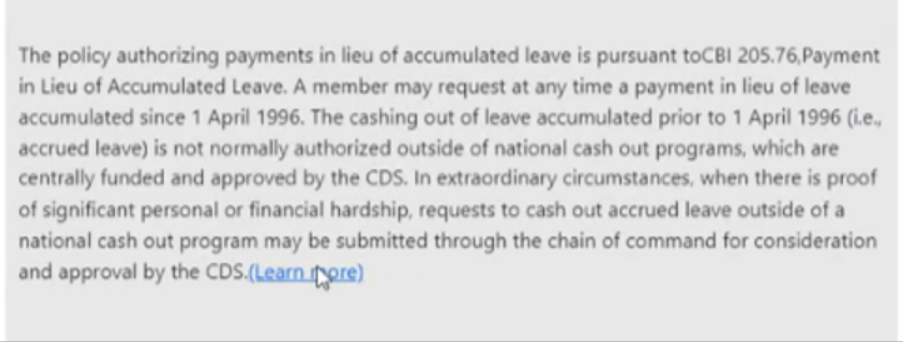

Fig. 5. An example of a Policy Section displayed on the UI with Information obtained through IR from Metadata Tagging

Once clicked, the URL leads the user to a page as seen in Figure 6, highlighting the passage which the LLM used as a source to generate the answer.

4 Performance Evaluation

The evaluation dataset was provided by members of Canadian Armed Forces and consisted of question-answer pairs, in both English and French. A snippet of this dataset is given in Table 4. Each language featured 30 questions per policy. Additionally, we removed any open-ended questions that could lead to multiple

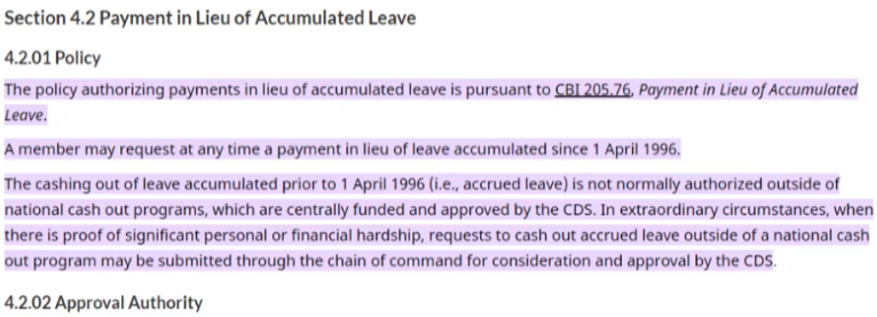

Fig. 6. Displayed Highlighted Passage provided through a hyperlink to users wishing to view the source of Truth used by the LLM

valid answers, aiming to enhance the clarity of the dataset. This curation process was designed with the goal of eliminating open-ended questions.

Question	Answer (Golden Answer)
Can head dress be removed while travelling on public transportation?	Chapter 2, Sect 1, para 32, sub-para i Public Transportation. "Personnel travelling aboard a local public conveyance may remove their headdress. Personnel travelling extended distances by aircraft, bus or rail, may remove their headdress while in transit, however, headdress shall be replaced prior to exiting the public conveyance, vehicle or aircraft"
Are CAF members permitted to use the drill movements of a foreign military?	Chapter 1, sect 1, para 21 "Canadian Armed Forces personnel, whether as individuals or formed contingents, are forbidden to use the drill movement of a foreign military or domestic organization. Only the CDS can personally, in writing, waive this direction. Requests for waivers must be staffed through the chain-of-command to DHH."
Pourquoi est-ce que les Forces canadiennes décernent les honneurs de groupe?	3-1-1 Les distinctions décernées aux groupes sont remises à titre de reconnaissance du public pour les exploits et les actes de corps militaires formés qui sortent du cadre normal de leurs fonctions et qui dépassent la norme élevée que l'on attend des militaires des Forces canadiennes.
Qui approuve les honneurs militaires?	3-1-5, 3-1-6 Le Gouverneur général approuve au nom du Souverain les conditions générales d'attribution d'honneurs de bataille qui ont permis à l'unité en cause de les mériter. Le Chef d'état-major de la Défense, sur recommandation d'un comité de honneurs de bataille spécialement constitué, approuve l'attribution des honneurs de bataille, la remise de distinctions honorifiques aux unités et la perpétuation des unités qui, suite à leur action sur le champ de bataille, ont mérité un honneur de bataille ou une distinction honorifique conformément aux coutumes établies.

Table 2. Sample French and English Questions and Human Generated Ground Truth Answers used for Evaluation of the Question Answering KMS

To ensure fair and accurate evaluations, we conducted manual assessments of the KMS's responses to verify their accuracy. We categorized the results into four groups: correct answers, wrong answers, unrelated answers, and refusal answers. A correct answer is an exactly correct response to the question, an unrelated an-

swer is a response that is irrelevant to the topic of the question, a refusal answer occurs when the model states it cannot provide an answer to the question. As a comparison, we compared our KMS's performance against OpenAI's GPT-4, utilizing the same parameters used in our KMS: a temperature of 0.1, top-p of 0.5, and a sampling rate of 50, which is fixed by OpenAI. To maintain fairness in our evaluations, we used the same retrieval settings and safety-engineered prompts for both OpenAI's model and our KMS. To be fair for GPT-4's built-in French language capabilities, we did an additional test by skipping the translation step as another comparison.

Table 3 displays the performance scores of our KMS relative to the human answers. We noted a minimal occurrence of refusal and unrelated answers. The infrequency of these types of answers suggests that our retrieval pipeline is functioning effectively.

Policy	**Type**	**English**	**French**
Leave Policy	Right answer	90.00%	73.33%
	Unrelated answer	10.00%	23.34%
	Wrong answer	0.00%	0.00%
	Refusal answer	0.00%	3.33%
Dress Policy	Right answer	86.67%	73.33%
	Unrelated answer	13.33%	26.67%
	Wrong answer	0.00%	0.00%
	Refusal answer	0.00%	0.00%
Drill Manual	Right answer	86.66%	73.33%
	Unrelated answer	6.66%	23.34%
	Wrong answer	0.00%	0.00%
	Refusal answer	6.67%	3.33%
Averages	Right answer	87.78%	73.33%
	Unrelated answer	10.00%	24.45%
	Wrong answer	0.00%	0.00%
	Refusal answer	2.22%	3.33%

Table 3. Scores from our KMS compared to the human answer

Table 4, on the other hand, presents the corresponding scores for OpenAI's GPT-4. Here, we observed a notable increase in the 'Refusal Answer' category, indicating that GPT-4 often declines to answer questions when it is uncertain of the correctness. This cautious approach is likely a result of extensive "guardrailing" incorporated during GPT-4's training, where safety mechanisms are implemented to ensure prudent responses, leading to a higher rate of refusal even when relevant data is available.

Policy	Type	English	French	French (without translation)
Leave Policy	Right answer	80.00%	70.00%	46.67%
	Unrelated answer	6.67%	10.00%	16.66%
	Wrong answer	0.00%	0.00%	0.00%
	Refusal answer	13.33%	20.00%	36.67%
Dress Policy	Right answer	63.33%	40.00%	26.67%
	Unrelated answer	3.33%	13.33%	6.67%
	Wrong answer	0.00%	0.00%	0.00%
	Refusal answer	33.33%	46.67%	66.67%
Drill Manual	Right answer	73.33%	50.00%	40.00%
	Unrelated answer	13.34%	26.67%	13.33%
	Wrong answer	0.00%	0.00%	0.00%
Averages	Refusal answer	13.33%	23.33%	46.67%
	Right answer	72.22%	53.33%	37.78%
	Unrelated answer	7.78%	16.67%	12.22%
	Wrong answer	0.00%	0.00%	0.00%
	Refusal answer	20.00%	30.00%	50.00%

Table 4. Scores from OpenAI's GPT-4 model compared to the human answer

Both tables reflect a proportional decline in accuracy in French, which can be expected given the LLMs are generally trained on more English data and some French terms using in the military policies can lead to errors in semantic meaning, a notable example of this is the drill performance "Feu de Joie," which translates as "bonfire" in both French and English. This demonstrates a similar translation issue on platforms like Google Translate, indicating a common challenge in handling ambiguous wordings across languages.

5 Conclusion

In this research, we have addressed the complexities involved in navigating bilingual military policies of the Canadian Armed Forces (CAF) by implementing a containerized AI-driven knowledge management system. By leveraging advanced AI models, metadata tagging, text embeddings, and vector databases, we developed a bilingual system capable of processing queries in both English and French. Our approach not only extends beyond traditional retrieval methods by integrating generative AI components but also ensures precise and contextually relevant responses.

Our results demonstrate the effectiveness of our system, achieving an accuracy rate of 87.78% for English queries and 73.33% for French queries. This significant improvement over traditional keyword-based retrieval methods is attributed to the semantic search capabilities and the innovative question-answering pipeline we implemented. Additionally, introducing a chunking engine and using a vector database have enhanced the system's efficiency and scalability, making it capable of handling extensive military documents. The integration of a "source of truth" approach further reinforces the reliability of the responses

generated by the system, allowing users to verify the accuracy of the information provided. By deploying our solution in a secure, containerized environment, we ensure data integrity and privacy, which are paramount for military applications.

Our study highlights the potential of AI-driven solutions in transforming information retrieval systems, particularly in domains requiring high accuracy and contextual understanding. Future work will focus on expanding the system's capabilities to include more diverse datasets and exploring the potential of continuous learning models to enhance the system's performance in English and French.

References

1. Advances in Computing and Network Communications: Proceedings of CoCoNet 2020, Volume 1. Springer Singapore (2021). https://doi.org/10.1007/978-981-33-6977-1, http://dx.doi.org/10.1007/978-981-33-6977-1
2. Amazon Web Services: Amazon lex (2024), https://aws.amazon.com/lex/, accessed: 2024-06-14
3. Andriopoulos, K., Pouwelse, J.: Augmenting llms with knowledge: A survey on hallucination prevention. North American Chapter of the Association for Computational Linguistics (2023)
4. Bast, H., Buchhold, B., Haussmann, E.: Semantic search on text and knowledge bases. Foundations and Trends® in Information Retrieval **10**(2-3), 119–271 (2016). https://doi.org/10.1561/1500000032, http://dx.doi.org/10.1561/1500000032
5. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. Advances in Neural Information Processing Systems 33 (NeurIPS 2020) (2020)
6. Carmel, D., Cohen, D., Fagin, R., Farchi, E., Herscovici, M., Maarek, Y.S., Soffer, A.: Static index pruning for information retrieval systems. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 43–50. SIGIR '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/383952.383958, https://doi.org/10.1145/383952.383958
7. Chen, H.Y., Yu, H.: Intent-based web page summarization with structure-aware chunking and generative language models. In: Companion Proceedings of the ACM Web Conference 2023. pp. 310–313. WWW '23 Companion, Association for Computing Machinery, New York, NY, USA (2023). https://doi.org/10.1145/3543873.3587372, https://doi.org/10.1145/3543873.3587372
8. Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., Stoyanov, V.: Unsupervised cross-lingual representation learning at scale. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020)
9. Deng, Y., Lam, W., Xie, Y., Chen, D., Li, Y., Yang, M., Shen, Y.: Joint learning of answer selection and answer summary generation in community question answering. Proceedings of the AAAI Conference on Artificial Intelligence 34 (2019)

10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019)
11. Fonseca, M.J., Jorge, J.: Indexing high-dimensional data for content-based retrieval in large databases. In: Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings. pp. 267– 274 (04 2003). https://doi.org/10.1109/DASFAA.2003.1192391
12. Fu, C., Xiang, C., Wang, C., Cai, D.: Fast approximate nearest neighbor search with the navigating spreading-out graph. Proceedings of the VLDB Endowment **12**(5), 461–474 (2018)
13. Google Cloud: Dialogflow (2024), https://cloud.google.com/dialogflow, accessed: 2024-06-14
14. Gunasekara, C., Sharafeldin, A., Triff, M., Kabir, Z., Joseph, R.B.: Information retrieval chatbot on military policies and standards. In: Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods (ICPRAM). p. 49 (February 2024)
15. Guu, K., Lee, K., Tung, Z., Pasupat, P., Chang, M.W.: Realm: Retrieval-augmented language model pre-training. Proceedings of Machine Learning Research (2020)
16. Hugging Face: Roberta: A robustly optimized bert pretraining approach (2024), accessed: 2024-06-14
17. Ji, S., Pan, S., Cambria, E., Marttinen, P., Yu, P.S.: A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems **33**(2), 494–514 (Feb 2022). https://doi.org/10.1109/tnnls.2021.3070843, http://dx.doi.org/10.1109/TNNLS.2021.3070843
18. Kathare, N., Reddy, O.V., Prabhu, V.: A comprehensive study of elasticsearch. International Journal of Science and Research (IJSR) (2020)
19. Leonhardt, J., Rudra, K., Khosla, M., Anand, A., Anand, A.: Efficient neural ranking using forward indexes. In: Proceedings of the ACM Web Conference 2022. WWW '22, ACM (Apr 2022). https://doi.org/10.1145/3485447.3511955, http://dx.doi.org/10.1145/3485447.3511955
20. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Neural Information Processing Systems (2021)
21. Mahapatra, A.K., Biswas, S.: Inverted indexes: Types and techniques. International Journal of Computer Science Issues **8** (07 2011)
22. Mangold, C.: A survey and classification of semantic search approaches. International Journal of Metadata, Semantics and Ontologies **2**(1), 23–34 (2007). https://doi.org/10.1504/IJMSO.2007.015073, https://www.inderscienceonline.com/doi/abs/10.1504/IJMSO.2007.015073
23. Manning, C.D., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press (2008), https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf, accessed: 2024-06-14
24. Meta: Seamlessm4t: A multimodal ai model for speech and text translations (2023), https://about.fb.com/news/2023/08/seamlessm4t-ai-translation-model/, accessed: 2024-06-14
25. Meta AI: Llama 2: Open foundation and fine-tuned chat models (2024), https://ai.meta.com/llama/, accessed: 2024-06-14
26. Microsoft: Qna maker (2024), https://www.qnamaker.ai/, accessed: 2024-06-14

27. Oliveira, P., Rocha, J.: Semantic annotation tools survey. In: 2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM). pp. 301–307 (2013). https://doi.org/10.1109/CIDM.2013.6597251
28. OpenAI: Chatgpt (2024), https://chat.openai.com/, accessed: 2024-06-14
29. Parmer, C., Johnson, A.: Dash [computer software]. https://plot.ly/dash (2018), accessed: 2024-06-14
30. Radford, A., Narasimhan, K.: Improving language understanding by generative pre-training (2018), https://api.semanticscholar.org/CorpusID:49313245
31. Rao, J., Liu, L., Tay, Y., Yang, W., Shi, P., Lin, J.: Bridging the gap between relevance matching and semantic matching for short text similarity modeling. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 5370–5381. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1540, https://aclanthology.org/D19-1540
32. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. Conference on Empirical Methods in Natural Language Processing (2019)
33. Robertson, S., Zaragoza, H.: The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends in Information Retrieval **3**, 333–389 (01 2009). https://doi.org/10.1561/1500000019
34. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (nov 1975). https://doi.org/10.1145/361219.361220, https://doi.org/10.1145/361219.361220
35. Secretariat, T.B.o.C.: Government of canada (Nov 2023), https://www.canada.ca/en/government/policy/dept.html
36. Shi, L., Zhang, K., Rong, W.: Query-response interactions by multi-tasks in semantic search for chatbot candidate retrieval (2022)
37. Steven, C.: Web scraping wikipedia using python and beautifulsoup (11 2019). https://doi.org/10.13140/RG.2.2.34480.71685
38. Strubell, E., Ganesh, A., McCallum, A.: Energy and policy considerations for deep learning in nlp. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
39. Su, H., Shi, W., Kasai, J., Wang, Y., Hu, Y., Ostendorf, M., tau Yih, W., Smith, N.A., Zettlemoyer, L., Yu, T.: One embedder, any task: Instruction-finetuned text embeddings. Findings of the Association for Computational Linguistics: ACL 2023 (2023)
40. Sun, Y., Yin, Z., Guo, Q., Wu, J., Qiu, X., Zhao, H.: Benchmarking hallucination in large language models based on unanswerable math word problem. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024) (2024)
41. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems (2023)
42. Wang, X., Yang, Q., Qiu, Y., Liang, J., He, Q., Gu, Z., Xiao, Y., Wang, W.: Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases (2023)
43. Wu, P., Manjunath, B., Chandrasekaran, S.: An adaptive index structure for high-dimensional similarity search. In: PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing. vol. 2195, pp. 71–77 (10 2001). https://doi.org/10.1007/3-540-45453-5_10

44. Ziegler, D.M., Stiennon, N., Wu, J., Brown, T.B., Radford, A., Amodei, D., Christiano, P., Irving, G.: Fine-tuning language models from human preferences (2020)