# Information Retrieval Chatbot on Military Policies and Standards

Charith Gunasekara[1][a], Alaa Sharafeldin[1], Matthew Triff[1], Zareen Kabir[2] and Rohan Ben Joseph[3]

[1]*Department of National Defence, Government of Canada, Ottawa, ON, Canada*

[2]*Department of Engineering Physics, McMaster University, Hamilton, ON, Canada*

[3]*Department of Computing Science, Simon Fraser University, Burnaby, BC, Canada*

Keywords: Natural Language Processing, Information Processing, Chatbots.

Abstract: In the Canadian Armed Forces (CAF), navigating through extensive policies and standards can be a challenging task. To address the need for streamlined access to these vital documents, this paper explores the usage of artificial intelligence (AI) and natural language processing (NLP) to create a question-answering chatbot. This chatbot is specifically tailored to pinpoint and retrieve specific passages from policy documents in response to user queries. Our approach involved first developing a comprehensive and systematic data collection technique for parsing the multi-formatted policy and standard documents. Following this, we implemented an advanced NLP-based information retrieval system to provide the most relevant answers to users' questions. Preliminary user evaluations showcased a promising accuracy rate of 88.46%. Even though this chatbot is designed to operate on military policy documents, it can be extended for similar use cases to automate information retrieval from long documents.

## 1 INTRODUCTION

Individuals across various professions often face the difficult task of navigating through long and dense legal and procedural documents. The sheer volume and complexity of these documents make them challenging to understand and apply effectively. Parsing through these documents can take significant amounts of time and may require help from an expert to clearly understand. Automating the parsing of these documents could significantly reduce the burden on individuals, eliminating the need for them to sift through the content themselves.

This challenge has led many organizations to utilize question-answering systems for information retrieval based on user queries (Bayan Abu Shawar, 2007). Many of these question-answering systems come in the form of "chatbots" to help support a better user experience and allow for a smoother conversation rhythm. This approach allows for a more streamlined information retrieval process and allows the user to experience a more manageable and less time-consuming process.

Some of the earliest methodologies used for chatbots were rule-based chatbots such as Artificial Linguistic Internet Computer Entity (ALICE) and Elizabeth which were chatbots that utilized pattern-matching techniques to answer questions(Shawar and Atwell, 2002). This pattern-matching relied on the usage of syntactic and semantic structures. Further advancements for these chatbots were done through the usage of AIML (Artificial Intelligence Markup Language) which used special markers and tags to have a robust dialogue identification and pattern recognition infrastructure (Marietto, 2013). These additions helped a more natural dialogue flow between users and chatbots.

Scripting languages have also played a crucial role in many rule-based chatbots. The usage of scripting languages such as RiveScript (Gupta et al., 2015) and ChatScript (Ramesh et al., 2017) has supported dynamic user inputs which had been a key limitation in many rule-based chatbots. These created key structures to be able to better understand emphasis, conversational redirects and contextual awareness. Although pattern-matching methodologies are easy to implement, they have key limitations which make them difficult to work with. These algorithms require pre-written patterns for many different user queries which creates a lot of work for developers. Additionally, they have been known to be difficult and inefficient in managing large-scale data-sets(Adamopoulou and Moussiades, 2020).

[a] https://orcid.org/0000-0002-7213-883X

Gunasekara, C., Sharafeldin, A., Triff, M., Kabir, Z. and Ben Joseph, R.
Information Retrieval Chatbot on Military Policies and Standards.
DOI: 10.5220/0012351200003654
Paper copyright by his Majesty the King in Right of Canada as represented by the Minister of National Defence
In *Proceedings of the 13th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2024)*, pages 714-722
ISBN: 978-989-758-684-2; ISSN: 2184-4313
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

Key innovation in this field was a necessity and diverged into many different areas. One such notable development was the creation of the QnA Maker by Microsoft which was a cloud-based natural language processing (NLP) service that uses a knowledge base (KB) comprised of both structured and semi-structured data (Agrawal et al., 2020). This tool provided new innovation for parsing semi-structured data such as FAQ pages and product manuals which wasn't possible with previous chatbots models. A similar style of chatbot was created by Google called Dialogflow which was also able to parse through structured and semi-structured content(Peyton and Unnikrishnan, 2023). However, a key disadvantage with this type of chatbot framework is that they require manual training through the usage of specific question-answer pairs, which is difficult to scale for organization with large and complex knowledge bases.

While the aforementioned innovations have significantly improved user interaction and conversation flow in chatbots, the field has witnessed substantial advancements with the integration of semantic search methodologies (Kim and Kim, 2022). Semantic search has been a key addition in question-answering systems as they allow chatbots to better understand user queries and provide a more natural and relevant response by recognizing the user's intent and the contextual meaning of terms as they appear in the searchable database. It goes beyond pattern matching, by understanding the context and the intention behind the question, offering more accurate and relevant results. The addition of semantic search could help to significantly improve simpler pattern-match-based chatbots to better understand user inputs.

Another fundamental component in the construction of effective question-answering systems is the integration of information retrieval (IR) systems. These systems enable chatbots to quickly and efficiently sift through vast amounts of data to identify relevant responses. One such IR technique is used in Best Matching 25 (BM25) , a bag-of-words ranking function (Amati, 2009). BM25, known for its efficacy in retrieving key documents based on term frequency and inverse document frequency, offers a speedy preliminary filtering mechanism. While BM25 has been instrumental in many information retrieval tasks, its primary design is not specifically tailored for answering questions but rather for ranking documents in order of relevance to a given query. BM25, rooted in the probabilistic information retrieval model, functions as a bag-of-words mechanism, emphasizing term frequency and inverse document frequency. This makes it swift and efficient, commonly used in traditional search engines. However, it inherently lacks a deep contextual understanding, sometimes missing contextually relevant documents without exact term matches and struggles with synonyms or paraphrasing. On the other hand, and Bi-directional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) a deep learning model, excels in grasping context by analyzing text bidirectionally, taking into account both word order and semantics. This depth allows BERT to perform exceptionally in diverse NLP tasks, from sentiment analysis to question-answering. However, this sophistication comes at the cost of computational intensity, which can hinder its use in real-time applications or resource-restrained environments. BM25 and BERT based models represent a distinct paradigms in information retrieval and natural language processing. Ultimately, choosing between BM25 and BERT hinges on the task's specific needs, available computational resources, and desired contextual depth.

Further advancements, like the introduction of clarifying questions and the development of a Multitask-based Semantic Search Neural Network (MSSNN) have also marked a significant progress in chatbot capabilities. These advancements, explored by (Aliannejadi et al., 2019) and (Shi et al., 2022) respectively, showcase progressive strides in improving the chatbot's capability to understand and respond to complex user queries more naturally and relevantly.

A key example of the usage of this type of system is in policy documents. Documents such as those in regards to privacy policies for companies and institutions are very complex documents that are long and messy. Experiments such as (Ravichander et al., 2019) conducted by Carnegie Mellon University, Penn State University and Fordham Law School, have shown that question-answering techniques can be used for users to get answers about a policy document. In their experiment, they used various models including BERT to benchmark the PrivacyQA dataset. Given the complexity of these policies, the ability to ask questions about a policy document would be incredibly beneficial. The results in (Ravichander et al., 2019) achieved a precision of 44% vs. the 69% of a human question-answering baseline.

In light of recent technological advancements, there remains an evident gap in ensuring domain-specific and contextually rich understandings, especially within specialized sectors like the military forces. Notably, the military forces houses detailed policy documents loaded with essential directives that personnel must diligently adhere to. Yet, the sheer volume and complexity of these documents pose a challenge to extracting relevant information efficiently. Presently, queries regarding these policy and standards documents are routed through various committees, a mechanism that, although reliable, is inherently time-consuming. This is primarily because addressing these queries constitutes a secondary responsibility for committee members, leading to potential delays in response times.

This research endeavours to bridge the aforementioned gap by tailoring advanced information retrieval and reranking algorithms specifically for military policies and standards documentation within the Canadian Armed Forces (CAF). Harnessing the power of the BERT architecture and its variants, our approach promises an elevated calibre of Natural Language Understanding (NLU) over CAF policy documents. The main objective of our study is to architect a scalable system proficient in interpreting user queries articulated in natural language and adeptly pinpointing the segment within a long policy document that most likely addresses the user's inquiry.

## 2 METHODOLOGY

### 2.1 Data

#### 2.1.1 Policies and Standards Data

The data-set utilized in this project was the Canadian National Defence Policies and Standards, referenced at (Canada, 2023). This data-set is made up of data stored in individual web pages with inconsistent formats making them challenging to access and parse. The policies in this web pages are formatted hierarchically with bullet points that can have sub-points with additional information on the parent point. Additionally the policies could also be contained in tables which provides an additional layer of challenge as deviates from the typical format.

#### 2.1.2 Data Collection

A systematic method for parsing this data is a necessity for creating a suitable data-set which can be used for the model. This is also an essential step to have an up-to-date data-set which can be updated in real-time. A set of Python scripts were used to extract the data when executed. The scripts were designed to accommodate the unique formats of the following policy documents: Canadian Forces General Messages (CANFORGENS), Canadian Forces Dress Instructions, Canadian Forces Leave Policy Manual, The Canadian Forces Manual of Drill and Ceremonial, and The Heritage Structure of the Canadian Forces,

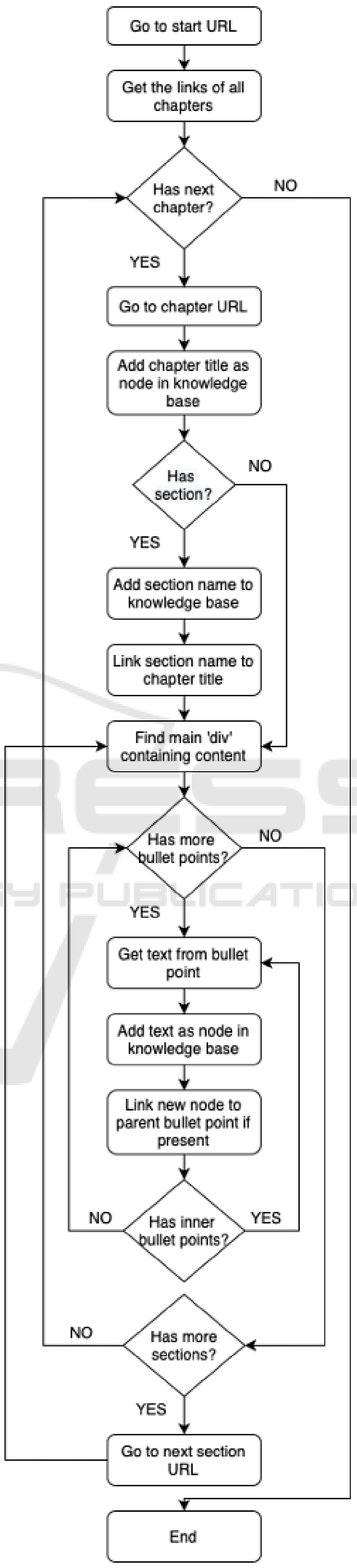

Go to start URL
Get the links of all chapters
Has next chapter?
NO
YES
Go to chapter URL
Add chapter title as node in knowledge base
Has section?
NO
YES
Add section name to knowledge base
Link section name to chapter title
Find main 'div' containing content
Has more bullet points?
NO
YES
Get text from bullet point
Add text as node in knowledge base
Link new node to parent bullet point if present
NO
Has inner bullet points?
YES
NO
Has more sections?
YES
Go to next section URL
End

Figure 1: Block diagram of the data collection algorithm.

with each document being scraped in both official languages, English and French.

The policies and manuals consist of text, images, and images with text. The images with text provide useful information and details that must not be left out of the parsed data. To parse through these various structures, the scripts utilized the web-scraping package BeautifulSoup (Zheng et al., 2015) for text and Pytesseract (Saoji et al., 2021) for the text embedded images. A recursive algorithm was developed to parse through each sub-point in the list of bullet-points until the algorithm reached the leaf points at which point it halted. The text was then stored along with the Hyper-Text Markup Language (HTML) tags to used in the Information Retrieval system. Figure 1 provides a block diagram of the data collection algorithm.

#### 2.1.3 Data Structure

To enable efficient querying and referencing of the policy documents and their respective formats, an appropriate data structure was required. Considering the nested bullet point formatting of the original documents a tree structure was found to be the best representation. We build the knowledge base using this structure, mirroring the hierarchical nature of the policy documents. The key components of the knowledge base are nodes and edges

*Nodes.* Represents an individual bullet points from the policy documents, and stores all relevant information about the text.

*Edges.* Symbolize the connections between nodes, indicating sub-points and their associated parent points. Every edge has parent and child specifications to identify the node level in the tree.

Each document is assigned a unique identifier (ID) to identify between different manuals or policies and account for language variations. This ID system ensures that we can query the database based on policy type and language preference.

Along with the policy information, storing all interactions between the chatbot and users was essential to refine the chatbot further and improve the passage retrieval models. To store these dialogues, we utilized sessions and utterances.

*Session.* Represents the entirety of an interaction between a user and the chatbot.

*Utterance.* Denotes individual messages within the broader dialogue context.

A comprehensive representation of this data structure is provided in Figure 2. See Table 1 in the Appendix for a detailed explanation of the database, components and functionalities.

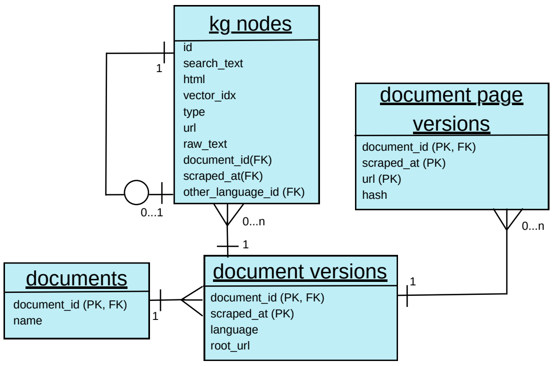


Figure 2: Database structure of knowledge base and dialogue management.

### 2.2 Answering Algorithms

Ensuring the accuracy and reliability of responses is pivotal in the deployment of question-answering systems, particularly within a context as nuanced and critical as the military. A retrieval-based chatbot system has been strategically developed to minimize user confusion and maximize trustworthiness. Such a system derives answers to user inquiries directly from a carefully curated knowledge corpus, thereby preserving the integrity and reliability of the information provided. The methodology centers around establishing a passage retrieval mechanism engineered to retrieve the most relevant passage from a policy document in response to a user query.

Central to our experiments is using BERT for encoding text into vectors, subsequently leveraging this technique to extract the most relevant passage in response to a given query. BERT, a language representation model, utilizes a Transformer model (Vaswani et al., 2017), departing from the commonly employed recurrent neural network in language modelling. Transformer models, exclusively based on attention mechanisms, enhance learning speed and reduce language model complexity. Notably, BERT trains bidirectionally, encoding each word within an embedding context-aware from both sides, considering preceding and following text. In BERT's architecture, the Transformer (Vaswani et al., 2017) model encoder is used. While the original Transformer, proposed by Google, presented an encoder-decoder architecture for sequence-to-sequence modelling, BERT has been adapted purely as a text encoder for varied tasks, such as classification and question answering. Employing multiple encoder blocks, BERT utilizes six blocks, each composed of a multi-head attention component and a rudimentary feed-forward network.

The multi-head attention mechanism employs several "heads" of dot-product attention, concatenating resultant vectors. Three vectors, key, value, and query, are established by multiplying input vectors by three weight matrices. The dot product between the query and the key vectors is computed, the values normalized using the softmax function, and subsequently multiplied with the value vector. This process enables the model to retain enhanced information regarding dependencies between tokens in the text.

The following discussion outlines the retrieval and rank algorithm that has been implemented, along with an exploration of the models that power it.

#### 2.2.1 Retrieval Algorithm

Considering the large number of possible passages that could potentially contain answers to a question, an algorithm capable of encoding passages into a vector, which functions proficiently in a question-answering setting, is needed. We have opted for the msmarco-distilbert-base-v3 model (Reimers and Gurevych, 2019), an encoder model pre-trained on the MSMARCO Passage Ranking data-set (Bajaj et al., 2016). MSMARCO, a vast information retrieval corpus conceived for semantic search applications, is derived from authentic user search engine queries (Nguyen et al., 2016). Although the full corpus encompasses 8.8 million passages, this model was trained on 500k examples (Nguyen et al., 2016). Utilizing this model as our encoder, we encode each passage derived from English policy documents, using our scrappers, and store it into a large matrix for subsequent retrieval.

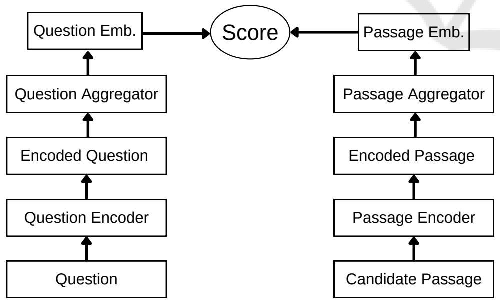


Figure 3: Bi-encoder architecture overview.

In the bi-encoder model, displayed in Figure 3, the encoder embeds both the question and the candidate passage independently. Initially, the encoder fragments the text into tokens, then employs a transformer model to encode all tokens (Devlin et al., 2019) , with a special token appended, into a set of vectors. A reduction function is subsequently utilized to convert the set of vectors into a singular vector. In our application, the vector of the special token represents the entirety of the text (Humeau et al., 2019). Following the encoding of the passages, we utilize the semantic search function from the sentence transformers package in Python (Reimers and Gurevych, 2019), filtering passages based on cosine similarity against the question vector.

#### 2.2.2 Re-Ranking Algorithm

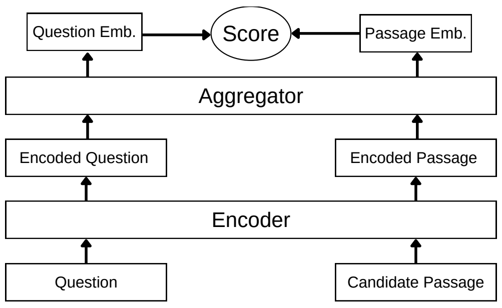


Figure 4: Cross-encoder architecture overview.

Upon extraction of the top k most relevant passages, a method for re-ranking was requisite to ensure that the passages presented were optimally arranged to satisfy the user's query. To accomplish this, a cross-encoder as shown in Figure 4 was employed. Divergent from bi-encoders, which assess the question and resultant passages independently, a cross-encoder embeds the question and passage text collectively, computing a score based on their relevance (Thakur et al., 2020) (Vig and Ramea, 2018). Although cross-encoders excel at identifying the most relevant passage, their computational intensity renders them suboptimal for processing extensive texts, thus justifying our initial employment of the bi-encoder. The ability of the cross-encoder to embed both the question and candidate passage collectively enables a richer interaction between the question and passage, permitting each token in the passage to engage with tokens in the input, thereby facilitating a sophisticated self-attention process. Consequently, the cross-encoder realizes a more refined attention and distance calculation between the question and passage compared to its bi-encoder counterpart. Please note that our two-step approach for retrieval and reranking semantic search steps differs from the Poly-encoder approach proposed by (Humeau et al., 2019), which is a one-step solution. The one-step poly-encoder solution by (Humeau et al., 2019) is an attempt to compose the accuracy of the cross-encoder layer to speed up the processing time by allowing real-time processing, which is not helpful in our use case because a couple of seconds delay in a two-step approach is justified with increased accuracy.

Prior to initiating this process, it is critical to concatenate the question and passage, introducing a special token between them as exhibited in (Humeau et al., 2019). Our implementation utilized the cross-encoder/ms-marco-TinyBERT-L-4 model from (Hugging Face, 2021), facilitated by the sentence transformer package (Reimers and Gurevych, 2019). This model, trained on the MSMARCO Passage Re-ranking Task, is expressly designed for information re-ranking. Following the ranking of all passages, those with the highest scores and answer relevance from the cross-encoder are returned to the user.

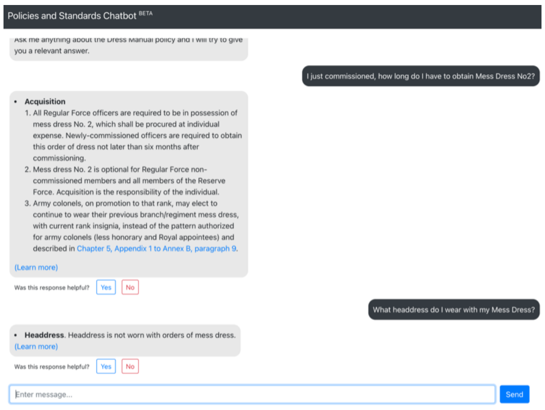
### 2.2.3 Language Detection

In order to detect the language of the questions asked, we employ an open-source language detection model papluca/xlm-roberta-base-language-detection from (Ou and Li, 2020). This model uses a multi-class text classification algorithm and is a fine-tuned version of xlm-roberta-base on the Language Identification data-set, where XLM-RoBERTa is a multilingual version of RoBERTa (Liu et al., 2019). XLM-RoBERTa is pre-trained on 2.5TB of filtered CommonCrawl data containing 100 languages, while the fine-tuned version papluca/xlm-roberta-base-language-detection was trained on 70k passages and currently supports 20 languages (Ou and Li, 2020). The Language Identification data-set is a collection of 90k samples consisting of text passages and their corresponding language label (Ou and Li, 2020). The current version of the chatbot only works in English; therefore, the retrieval mechanism is activated if the language is detected as English. Future work on this project involves adding in the ability for the model to work in the French language as well therefore this is a crucial feature in the chatbot to detect the language of the questions.

## 2.3 User Interface (UI) Development

A web application was built as shown in Figure 5 where users could interface with the chatbot and we could utilize that data to iteratively improve on our application. We used React as a front-end framework for the User Interface (UI) development. For the back-end of the application, we used Flask as it is a lightweight basic Application Programming Interface (API) framework that suited the needs of the application.

Integrating the chatbot through Microsoft Teams was also am important addition to have further user involvement, as it is a common form of communication throughout the organization. Therefore, we developed a gateway to communicate with the chatbot through Teams using the Microsoft Bot Framework (Biswas, 2018).



Ask me anything about the Dress Manual policy and I will try to give you a relevant answer.

I just commissioned, how long do I have to obtain Mess Dress No2?

- **Acquisition**
  1. All Regular Force officers are required to be in possession of mess dress No. 2, which shall be procured at individual expense. Newly-commissioned officers are required to obtain this order of dress not later than six months after commissioning.
  2. Mess dress No. 2 is optional for Regular Force non-commissioned members and all members of the Reserve Force. Acquisition is the responsibility of the individual.
  3. Army colonels, on promotion to that rank, may elect to continue to wear their previous branch/regiment mess dress, with current rank insignia, instead of the pattern authorized for army colonels (less honorary and Royal appointees) and described in Chapter 5, Appendix 1 to Annex B, paragraph 9.

(Learn more)

Was this response helpful? Yes No

What headdress do I wear with my Mess Dress?

- **Headdress**. Headdress is not worn with orders of mess dress.

(Learn more)

Was this response helpful? Yes No

Figure 5: Web UI built with React.

## 2.4 Deployment

In order to deploy the application, it has been split into three components. A front-end web application, a back-end API server, and a database. To facilitate easy and scalable deployment of the tool, both the front-end and back-end have been separated into their own Docker containers. Containers provide the benefit of isolating application components from underlying system dependencies and requirements. This allows applications to be repeatably deployed without dealing with significant and complex setup procedures on the servers from which they are run.

The chatbot comprises two main components: a back-end API server and a front-end web application. The back-end, as described in Section 2.2, is computationally intensive, processing user questions to determine the most appropriate answer. The front-end web application consists of the interface described in Section 2.3. The front-end is relatively lightweight and does not require significant computational power to run. A SQLite database is used within the back-end container to store the policy corpus, as well as previous questions and feedback from users.

Given that the chatbot relies solely on publicly accessible policy information, it is well-suited for deployment in a cloud environment. The most straightforward deployment of the system could use an Infrastructure-as-a-Service (IaaS) offering from a cloud provider. A cloud virtual machine (VM) with Docker installed could run both the front-end and back-end containers (the back-end container also includes the SQLite database).

Access to this VM could then be granted to users who would access the chatbot through the web interface to submit questions.. Figure 6a shows a system diagram for this deployment model.

However, the simple deployment approach is inadequate for a system that is deployed to the en-

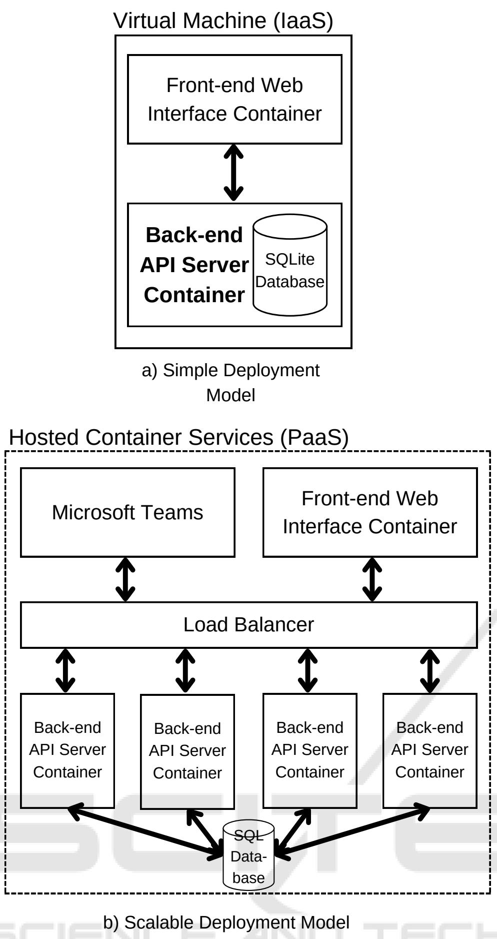

Virtual Machine (IaaS)
Front-end Web Interface Container
Back-end API Server Container
SQLite Database
a) Simple Deployment Model
Hosted Container Services (PaaS)
Microsoft Teams
Front-end Web Interface Container
Load Balancer
Back-end API Server Container
Back-end API Server Container
Back-end API Server Container
Back-end API Server Container
SQL Data-base
b) Scalable Deployment Model

Figure 6: A comparison of both simple and scalable deployment models.

tire DND/CAF organization particularly when faced with a substantial volume of concurrent requests. Instead, the following changes should be implemented in the future. Recognizing that the back-end API server is the most resource-intensive component and likely to become a system bottleneck, additional Docker containers can be leveraged to deploy multiple instances of the back-end. To facilitate this, the SQLite database would be substituted with a dedicated, hosted SQL database, which all back-end containers would connect to. Additionally, a load balancer component would be introduced to assign incoming sessions to specific back-end instances, ensuring an equitable distribution of requests. When a new session is opened with the chatbot, the load balancer will assign that session to a specific instance of the back-end and ensure that no instance has more requests than it can handle.

Finally, the front-end web interface should not be the only way for users to connect with the system. Microsoft Teams can be integrated as alternative front-end in addition to the web interface. All of these components can be deployed using a Platform-as-a-Service (PaaS) offering from a cloud provider. Such PaaS solutions eliminate the need for administrators to manage individual VMs, reducing the overall resources required for system maintenance. Figure 6(b) illustrates the system configuration for this more scalable deployment model.

## 3 PERFORMANCE EVALUATION

To assess the performance of our chatbot, we compiled a test bank comprising questions and corresponding passages containing the most accurate answers to the best of our knowledge.

In order to create a robust knowledge base of questions, we conducted early demonstrations of our application during the testing phase. These demonstrations served the dual purpose of introducing our application to users at the earliest possible stage and obtaining real-world queries from them. These user interactions yielded a substantial pool of questions, which in turn facilitated an initial assessment of the chatbot's performance.

Once we obtained a sufficient number of question-answer pairs, we proceeded to develop a script capable of evaluating the chatbot's performance. The evaluation script systematically traverses each question within the test bank, comparing the chatbot's responses with the expected answers from the test bank. The number of accurately answered questions was tallied, and to quantify our evaluation using a metric, we computed the ratio of correctly answered questions to the total number of questions in the test bank, yielding a percentage indicative of the chatbot's success rate.

Our evaluation suggests that our model excels at answering fundamental questions but faces challenges when questions necessitate additional contextual information. The chatbot gave 23 out of 26 questions the correct answers, and three questions received incorrect answers. Hence, the chatbot's accuracy, as assessed using the current test bank, can be calculated as follows.

$$\begin{aligned} Accuracy &= \frac{23 \times 100 + 3 \times 0}{26} \times 100\% \\ &= 88.46\% \end{aligned} \tag{1}$$

## 4 FUTURE WORK

While our initial results are intriguing, there is still room for improvement.

We are currently training our model to process queries in French. For the French text, a pre-trained French encoder model such as camembert-large model from (Martin et al., 2019) would be suitable. However, our early tests show that it does not perform as well as the English encoder model for information retrieval tasks; therefore, further fine-tuning is required.

Currently, the question answering algorithm is retrieval-based, meaning that all the responses it gives are directly from a policy. While this ensure that chatbot does not hallucinate, some of the responses it returns can be long since they come from a policy document where the length of text is not an issue. In order to solve this, we could use a large language model to perform text summarization on the response from the chatbot. Text summarization is an NLP task where a model attempts to shorten the input text as much as possible while maintaining all the vital information. In doing so, the responses from the chatbot will be much shorter and easier for the users to read.

We also intend to fine-tune both the bi-encoder and cross-encoder models from the BERT encoder to strengthen the chatbot's performance. Recognizing the potential of specialization, we aspire to train the models to more policy and military-specific language and contexts. Central to this enhancement strategy will be user feedback, which we plan to harness for both stages of fine-tuning. Initially, we'll fine tune the bi-encoder model using techniques from (Wolf et al., 2020), with the data collected from user feedback. Subsequently, the cross-encoder model will also be fine-tuned for optimal answer re-ranking (Thakur et al., 2020).

## 5 CONCLUSIONS

In conclusion, recognizing the complexity of long military policy documents, we introduced an NLP-based scalable and automated system to extract information from web documents texts. We developed a chatbot that leverages two BERT based encoder models: the bi-encoder for initial retrieval of the most relevant answers and the cross-encoder for re-ranking based on relevance. This chatbot is housed within a user-friendly web application crafted with Flask and React, eliminating the need for separate installations. While our evaluations highlight the chatbot's capability to address fundamental questions with approximately 88.46% accuracy on military policy topics, there are areas to enhance. Specifically, more precise responses could be elicited when users pose vague questions. The areas for future improvements have also been identified, including fine-tuning retrieval models based on user feedback and using large language models with semantic search to generate more accurate responses.

## REFERENCES

Adamopoulou, E. and Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, 2:100006.

Agrawal, P., Menon, T., Kam, A., Naim, M., Chouragade, C., Singh, G., Kulkarni, R., Suri, A., Katakam, S., Pratik, V., Bansal, P., Kaur, S., Duggal, A., Chalabi, A., Choudhari, P., Satti, S. R., Nayak, N., and Rajput, N. (2020). Qnamaker: Data to bot in 2 minutes. In *Companion Proceedings of the Web Conference 2020*, WWW '20, pages 131–134, New York, NY, USA. Association for Computing Machinery.

Aliannejadi, M., Zamani, H., Crestani, F., and Croft, W. B. (2019). Asking clarifying questions in open-domain information-seeking conversations. *CoRR*, abs/1907.06554.

Amati, G. (2009). *BM25*, pages 257–260. Springer US, Boston, MA.

Bajaj, P., Campos, D., Craswell, N., Deng, L., Gao, J., Liu, X., Majumder, R., McNamara, A., Mitra, B., Nguyen, T., et al. (2016). Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Bayan Abu Shawar, E. A. (2007). Chatbots: Are they really useful? *Journal for Language Technology and Computational Linguistics*, 22(1):29–49.

Biswas, M. (2018). Microsoft bot framework. *Beginning AI Bot Frameworks: Getting Started with Bot Development*, pages 25–66.

Canada (2023). Policies, standards, orders, directives and regulations from the Department of National Defence and the Canadian Armed Forces. https://www.canada.ca/en/department-national-defence/corporate/policies-standards.html. [Online; accessed 05-July-2023].

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Gupta, S., Borkar, D., de Mello, C. S. B., and Patil, S. S. (2015). An e-commerce website based chatbot.

Hugging Face (2021). cross-encoder/ms-marco-tinybert-l-4. https://huggingface.co/cross-encoder/ms-marco-TinyBERT-L-4. [Online; accessed 05-October-2023].

Humeau, S., Shuster, K., Lachaux, M., and Weston, J. (2019). Real-time inference in multi-sentence tasks with deep pretrained transformers. *CoRR*, abs/1905.01969.

Kim, M. and Kim, D. (2022). A suggestion on the lda-based topic modeling technique based on elasticsearch for indexing academic research results. *Applied Sciences*, 12(6):3118.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Marietto, M. D. G. B., d. A. R. V. B. G. D. O. B. W. T. P. E. F. R. D. S. . d. S. V. L. (2013). Artificial intelligence markup language: A brief tutorial. *CoRR*, abs/1307.3091.

Martin, L., Müller, B., Suárez, P. J. O., Dupont, Y., Romary, L., de la Clergerie, É. V., Seddah, D., and Sagot, B. (2019). Camembert: a tasty french language model. *CoRR*, abs/1911.03894.

Nguyen, T., Rosenberg, M., Song, X., Gao, J., Tiwary, S., Majumder, R., and Deng, L. (2016). MS MARCO: A human generated machine reading comprehension dataset. *CoRR*, abs/1611.09268.

Ou, X. and Li, H. (2020). Ynu@ dravidian-codemix-fire2020: Xlm-roberta for multi-language sentiment analysis. In *FIRE (Working Notes)*, pages 560–565.

Peyton, K. and Unnikrishnan, S. (2023). A comparison of chatbot platforms with the state-of-the-art sentence bert for answering online student faqs. *Results in Engineering*, 17:100856.

Ramesh, K., Ravishankaran, S., Joshi, A., and Chandrasekaran, K. (2017). A survey of design techniques for conversational agents. In Kaushik, S., Gupta, D., Kharb, L., and Chahal, D., editors, *Information, Communication and Computing Technology*, pages 336–350, Singapore. Springer Singapore.

Ravichander, A., Black, A. W., Wilson, S., Norton, T., and Sadeh, N. (2019). Question answering for privacy policies: Combining computational and legal perspectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4947–4958, Hong Kong, China. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Saoji, S., Eqbal, A., and Vidyapeeth, B. (2021). Text recognition and detection from images using pytesseract. *J Interdiscip Cycle Res*, 13:1674–1679.

Shawar, B. and Atwell, E. (2002). *A comparison between Alice and Elizabeth chatbot systems*. University of Leeds, School of Computing research report 2002.19. Shawar, BA and Atwell, E (c) 2002, University of Leeds. Reproduced with permission from the copyright holders.

Shi, L., Zhang, K., and Rong, W. (2022). Query-response interactions by multi-tasks in semantic search for chatbot candidate retrieval.

Thakur, N., Reimers, N., Daxenberger, J., and Gurevych, I. (2020). Augmented SBERT: data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. *CoRR*, abs/2010.08240.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762.

Vig, J. and Ramea, K. (2018). Comparison of transfer-learning approaches for response selection in multi-turn conversations.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Saux, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing.

Zheng, C., He, G., and Peng, Z. (2015). A study of web information extraction technology based on beautiful soup. *J. Comput.*, 10(6):381–387.

## APPENDIX

Table 1: Description of database structures.

| Table Name | Field | Description |
|---|---|---|
| **kg_nodes** | id | Unique identification number of text |
| | search_text | Scraped text (with acronyms replaced) to be retrieved when queried |
| | html | Scraped text with HTML tags |
| | type | Text type (section or body) |
| | url | URL of the policy that the text was retrieved from |
| | raw_text | Scraped text (with acronyms) |
| | document_id | Unique identification number of the URL |
| | scraped_at | Time text was scraped |
| | other_language_id | ID of the same text in another language |
| **document_page_versions** | document_id | Unique identification number of the URL |
| | scraped_at | Time the webpage was scraped |
| | url | URL of the page scraped |
| | hash | Hashes of individual pages |
| **document_versions** | document_id | Unique identification number of the URL |
| | scraped_at | Time the policy was scraped |
| | root_url | URL of the main page scraped |
| **documents** | document_id | Unique identification number of the URL |
| | name | Name of the policy |
| | language | Language of policy (English or French) |
| | other_language_id | document_id of the same policy in another language |