IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 January 2024, accepted 12 March 2024, date of publication 21 March 2024, date of current version 28 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380006

RESEARCH ARTICLE

Transparent, Low Resource, and Context-Aware Information Retrieval From a Closed Domain Knowledge Base

SHUBHAM RATERIA[1] AND SANJAY SINGH[2], (Senior Member, IEEE)
[1]Custiv Manufacturing, Bengaluru 560034, India
[2]Department of Information and Communication Technology, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
Corresponding author: Sanjay Singh (sanjay.singh@manipal.edu)

ABSTRACT In large-scale enterprises, vast amounts of textual information are shared across corporate repositories and intranet websites. Traditional search techniques that lack context sensitivity, often fail to retrieve pertinent data efficiently. Modern techniques that use a distributed representation of words require a considerable training dataset and computation, thereby presenting financial and operational burdens. Generative models for information search suffer from problems of transparency and hallucination, which can be detrimental, especially for organizations and their stakeholders who rely on these results for critical business operations. This paper presents a non-goal oriented conversational agent based on a collection of finite state machines and an information search model for text search from an extensive collection of stored corporate documents and intranet websites. We used a distributed representation of words derived from the BERT model, which allows for contextual searching. We minimally fine-tuned a BERT model on a multi-label text classification task specific to a closed-domain knowledge base. Based on DCG metrics, our information retrieval model using distributed embeddings from the minimally trained BERT model and Word Movers Distance for calculating topic similarity is more relevant to user queries than BERT embeddings with cosine similarity and BM25. Our architecture promises to significantly expedite and improve the accuracy of information retrieval in closed-domain systems without the need for a massive training dataset or expensive computing while maintaining transparency.

INDEX TERMS Natural language processing, conversational agent, information retrieval, closed domain knowledge base, BERT word movers distance.

I. INTRODUCTION

In today's data-rich environment, the daily expansion of information has intensified the challenge of identifying and retrieving the relevant data. Organizations have a wealth of knowledge stored in documents and on intranet websites. Efficient search and retrieval are crucial when such information is distributed globally.

Identifying and searching for relevant information and documents given a query is a core challenge for information retrieval (IR). The applications range from web searching, text mining, and document search from highly domain-specific documents. For applications that involve highly domain-specific documents, the content matching of the query to the document is crucial. These documents use words and syntax that may be specialized for a particular domain.

When we consider finding documents relevant to a query, the traditional approach uses the frequency of the words in the document. This approach has led to successful algorithms such as TF-IDF [1] and BM25 [2]. However, traditional information retrieval algorithms, do not model the semantics or context of words. The system has no method for determining how similar or dissimilar the two terms are. Advances in Natural Language Processing have allowed us to model embeddings for words in a language that encodes the

The associate editor coordinating the review of this manuscript and approving it for publication was Fu Lee Wang.

contextual similarities between words. Word2Vec [3] was the first neural network architecture to produce word embeddings that showed some linguistic contexts. Furthermore, state-of-the-art studies on neural network architectures, such as GloVe [4], Elmo [5], and BERT [6] have been developed to represent natural language in a latent space and learn embeddings for words based on different downstream uses. Moreover, the Elmo and BERT models can be pre-trained and fine-tuned for a specific downstream task. This is similar to the transfer learning in computer vision. Learning the context of words using a neural network opens a new search domain.

Neural models for information retrieval tend to use a setup known as telescoping [7], [8], [9]. In this system, we search for the documents in question using a traditional IR model, such as TF-IDF and BM25 and form a set of candidate documents. The neural model then examines the candidate documents to search for relevant results.

We consider and address these information management system problems and demonstrate a system that allows users to contextually find information. We address this problem in a domain-specific setup where the documents in question are collections of corporate documents.

Closed domain information retrieval also benefits from query completeness when obtaining pertinent results for a query [10]. Often, queries do not capture a user's information needs very well; for example, a query may consist of only a few words, the query words may not match the words used in any of the relevant documents, or the user may not know how to express the information needs. Domain knowledge can be incorporated into the system to reduce the number of iterations required by the user to derive relevant information from a search system. This is accomplished by providing a conversational agent to the user, where the agent can ask the user questions to provide additional information.

Interest in conversational agents has increased in the research community and commercial systems over the past few years. Conversational agents can efficiently replace humans in goal-oriented settings in which the user wants to achieve a specific goal, such as booking travel tickets and ordering food [11], [12], [13]. Chatbots are commonly seen as intelligent agents who can converse with humans in the way humans interact; however, this is difficult. Earlier chatbot systems used scripts of questions and answers to respond to users. Each response was based only on the previous utterance of the user. Chatbots, in which the turn-taking of conversations was also implemented, were mostly used to retrieve information from a database of conversational scripts. Advances in natural language processing techniques with deep learning have led researchers to use recurrent neural networks to model human conversations [14], [15], [16], [17]. Finite-state machines have also successfully modeled human conversations for goal-oriented chatbots. Non-goal-oriented chatbots require a knowledge base from which they can extract information and converse with the user. Understanding user intent and maintaining context play a crucial role in such agents.

Finite-state machines (FSMs) offer a versatile framework for modeling machines, humans, and mixed-initiative agent architectures. Deep learning-based methods for designing non-goal-oriented chatbots encounter two primary challenges. First, end-to-end deep learning does not inherently equip the conversational agent to understand the context and meaning of the conversation. However, this understanding often requires an engineered knowledge base for information retrieval. Second, these deep learning models for conversation require extensive training data, which are often scarce for domain-specific applications.

Recent advances in Natural Language Processing, such as Generative Pre-Training (GPT) [18], [19], [20] and RLHF [21], [22], [23], [24], have paved the way for Large Language Models (LLMs). Architectures such as GPT3 and GPT4 are increasingly employed for open-domain question answering and information retrieval. However, these LLMs require significant computational resources and large training datasets to achieve optimal results. They are also susceptible to hallucinations, lack transparency, and can underperform compared to domain-specific fine-tuned models, such as BERT [25]. Commercial non-goal-oriented conversational agents typically rely on handcrafted features and knowledge-based data extraction. Although sequence-to-sequence and other neural network models have shown promise in emulating human conversations, they often lack provisions for external knowledge integration and face the challenge of data scarcity for domain-specific training. Traditional information retrieval systems employ non-contextual text representations, whereas neural network models offer contextualized text representations that enhance information relevance. In our research, we harnessed FSMs in a novel manner to model conversations by leveraging word embeddings from the BERT model to extract pertinent information from textual documents.

The contributions of this paper are summarized as follows:

1) A finite-state machine-based architecture for conversational agents to retrieve information from corporate documents (Microsoft Word and PDF) containing data stored in tabular format and from the intranet.
2) An information search model using a neural network architecture based on distributed word embeddings extracted from a BERT model fine-tuned on a classification task, making highly accurate predictions of document meta-data. We then used these predictions to filter documents to generate a set of candidate documents. We use the word movers distance to find the most relevant information from within documents.
3) We compared the relevance of the results from an information search model using distributed representations with cosine similarity and word movers distance. We use BM25 as a baseline and show that neural word embeddings with Word Movers Distance provide more relevant results than traditional IR algorithms using DCG metrics.

Our method enables an organization to build IR systems that do not require expensive hardware or a large labeled dataset for training, and domain knowledge can be built into a conversational agent to ensure query completeness. Since results are derived directly from the document through query-text matching, the system is also transparent and provides the source of its answers.

II. RELATED WORK

A vast amount of research literature is available on conversational agents and information retrieval. Various approaches for chatbot development include matching question-response pairs [26], [27], using neural networks [28], and modeling conversations with finite state machines [29]. One of the earliest works in chatbots was the ELIZA chatbot [30] developed by Joseph Weizenbaum. ELIZA used pattern matching and pre-written interactive scripts to engage users in conversation. The ALICE [31] chatbot is another early work on chatbots. It used the AIML markup language to create interactive templates to converse with the user.

The rise of machine learning and neural networks [16] in natural language processing gave way to natural language understanding. This allowed computer systems to understand the user's words rather than mindlessly using information retrieval models. Intent and entity detection [32], [33], [34], [35] in a user utterance is now used by most commercial chatbot systems [12].

Song et al. [36] demonstrated a retrieval-based conversational system that uses an ensemble of retrieval-based and generative-based approaches. They search fored user-based utterances in a large conversational repository, return a reply that best matches the query, and synthesize new replies. The generated and retrieved responses then perform a re-ranking process to determine the final answer to the output.

Instead of using Q-R pairs for chatbots, Yan et al. [37] extracted data from unstructured documents. A model was designed to directly measure the relevance between the utterances and responses. This chatbot is intended for short-text conversations, in which the response depends only on the last statement. Answers were obtained after selecting sentences from the given documents by ranking all possible sentences based on features designed at different granularity levels. They used the BM25 to generate a set of candidate sentences. The candidate sentences were re-ranked, and the relevance score of a sentence was computed as a weighted sum of different relevance functions. These relevance functions were word, phrase, sentence, document, relation, type, and topic-levels.

Conversational agents have been applied to IR problems across various domains. GeCoAgent [38] is a conversational agent modeled as a big data agent for clinicians and biologists. The agent helps users extract relevant data from the repositories and perform data analysis. SAACS [39] presents a framework and techniques that augment conversational search services with the capability to understand and reason about subjective user utterances. The authors discussed automatic subjective tag extraction from user utterances and online reviews using state-of-the-art machine learning techniques such as BERT, adversarial training, and data programming. HoPE [40] is an architecture for conversational agents that uses ontology-based modeling and Sentence-BERT [41] networks adjusted on pregnancy guidelines data to support pregnant women in obtaining more reliable information during the baby's thousand day period.

Neural networks are becoming increasingly popular for development of conversational agents. Vinyals and Le [14], used a sequence-to-sequence neural network [16] framework for end-to-end training of conversational agents. The model converses by predicting the following sentence given the previous sentence or sentences in a conversation. They specified that their model could perform simple reasoning and extract knowledge from domain-specific datasets. Wu et al. [42] used a sequential matching framework for context-sensitive chatbots. They match each utterance's response in the context of multiple granularity levels and distill important matching information from each pair as a vector by using convolution and pooling operations. The vectors are then accumulated chronologically through an RNN, which models the relationships among the utterances. The final matching score is calculated using the hidden states of the RNN. Yi and Jung [29], showed that the classical method of developing chatbots leads to better user satisfaction than end-to-end neural network-based models. The chatbot that they developed was modeled with a finite-state machine using a bot-initiative strategy. Traditional IR methods using TF-IDF and SVM have been used to extract information.

Mohammad Mahdi Abdollah Pour et al. [43] focused on self-supervised contrastive learning methods for late and early fusion in Reviewed-Item Retrieval (RIR). It introduces novel contrastive fine-tuning techniques for BERT embeddings tailored for both the late and early fusion approaches. This study demonstrates that late fusion methods significantly outperform early fusion methods, highlighting the importance of individual review nuances in query scoring.

Rachmawati and Yulianti [44] explored the efficacy of transfer learning in building a CDQA system for COVID-19 inquiries. It introduces the Sequential Dependence Model (SDM) as the retriever in a retriever-reader architecture, significantly enhancing system accuracy. The study found optimal performance with the top 20 documents, five-sentence passages, and BERT-large as the reader model, demonstrating the superiority of transfer learning over non-transfer learning approaches in CDQA systems.

Ezhilarasi et al. [45] presented a comprehensive framework for enhancing crop information retrieval. This framework integrates LDW-ontology to calculate word importance and SNM-BERT for data training, aiming to tackle challenges such as unstructured data, low relevancy, and high computation time. The methodology included data pre-processing, crop ontology construction using OWL, visualization, and storage in MongoDB, followed by information retrieval through clustering. The SNM-BERT model, which utilizes

a transformer and an attention mechanism, significantly improves the accuracy, precision, and query retrieval time for simple and complex queries, outperforming existing techniques in agricultural information retrieval.

In their study, Larionov et al. [46] presented the architecture of Tartan, a non-goal-oriented bot. It provides users with engaging and fluent casual conversations, emphasizing structured conversations based on flexible finite-state models. Instead of using only one chat and information retrieval model, they used a different state machine for each conversational topic. The FSMs were swapped in and out as and when the user intention suggested.

An end-to-end deep learning system has also been used for dialogue state tracking in conversational agents. Williams et al. [28] developed a Hybrid Code Networks. Hybrid Code Networks allow the training of conversational agents with RNN architecture. A crucial part of these networks is enabling the user to engineer domain-specific knowledge into a conversational agent. Action templates can be engineered into agents that allow the execution of specific actions. Hybrid Code Networks can learn state transitions in a conversation, which can be used instead of modeling conversational scripts using finite-state machines.

Neural networks have become a significant part of natural language processing. Neural network models are currently being researched for information retrieval as well. Mitra et al. [47] presented a dual embedding space model, in which Word2Vec word embedding was used for IR. These two sets of embeddings represent documents and user queries, respectively. The query is mapped using the input representation of Word2Vec, and the documents are mapped using the output representation of Word2Vec. Wu and Li [48] used document topics to measure document similarity. Each document is considered to be composed of predefined topics. A word cluster denotes a topic. A word cluster is generated from the word-word co-occurrence matrix using PIRM [49]. They used GloVe word embeddings.

Building upon the extensive corpus of research in conversational agents and information retrieval, our work distinctively positions itself by integrating a finite state machine (FSM) architecture for query completeness with the nuanced capabilities of neural network-based distributed word embeddings, specifically from the BERT model. Unlike conventional methods which rely heavily on traditional algorithms or neural networks, this study introduces a novel architecture that combines both improved accuracy and relevance. Emphasizing transparency and resource efficiency, this approach is particularly suited to organizations with limited computational resources, offering a scalable and interpretable solution for corporate document retrieval.

III. CHATBOT ARCHITECTURE

Figure 1 shows the chatbot architecture. Natural Language Understanding is crucial for determining conversational flow and information extraction.

A. NATURAL LANGUAGE UNDERSTANDING

We used the Snips NLU open-source package for intent and entity extraction from the user utterances. Three primary user intentions were extracted. Three data sources were used: tabular sources, such as CSV files and Excel Sheets, text documents, such as PDFs and Word Documents, and intranet websites. Each intent suggests the source from which the user wants information. Each type of data source provides information unique to the source. We need to determine whether the user intends to:

1) Extract information from tabular data sources
2) Extract information from and view the location of text documents
3) Extract information from and view the location of intranet websites.

B. STATE MACHINE MANAGEMENT

The chatbot was modeled using a hierarchy of finite-state machines. The agent has multiple state machines, one for each user intent. The state machine manager can answer the user's basic questions and guide them to engage in conversation. Depending on the user's intent, it also selects the state machine that should be employed. The state machine manager recognizes the user intent, that is, the user wants to search for information from tabular sources, documents, or intranet websites and selects the associated state machine. The related state machine then drives the conversation.

Each state machine has a start state, and triggers its own state changes. Each state machine also has its own model for entity extraction. The state machine manager only has an intent-detection model. It also has an object that holds the currently active sub-state machine. If a specified intent is detected from a user's utterance, it triggers a change in its state. It has the same number of states as specified user intents. In this case, there are three intents. The state's change sets the state machine associated with the intent as the current state and puts it in the start state. Next, all utterances are passed to the currently active state machine. As the user converses with the bot, the global state-machine manager parses each statement to check its intent. If the intent does not suggest a functionality change, it passes the utterance to the currently active state machine. The current active-state machine parses entities from a statement using its intent-entity model.

This is a machine initiative chatbot. A machine initiative means that once the chatbot knows the user's intent, it can ask the user questions to gain more information about the user's query. The following are some of the entities that the chatbot can understand:

1) **Category**: The category from which the user wants information
2) **Country**: The country from which the user wants information

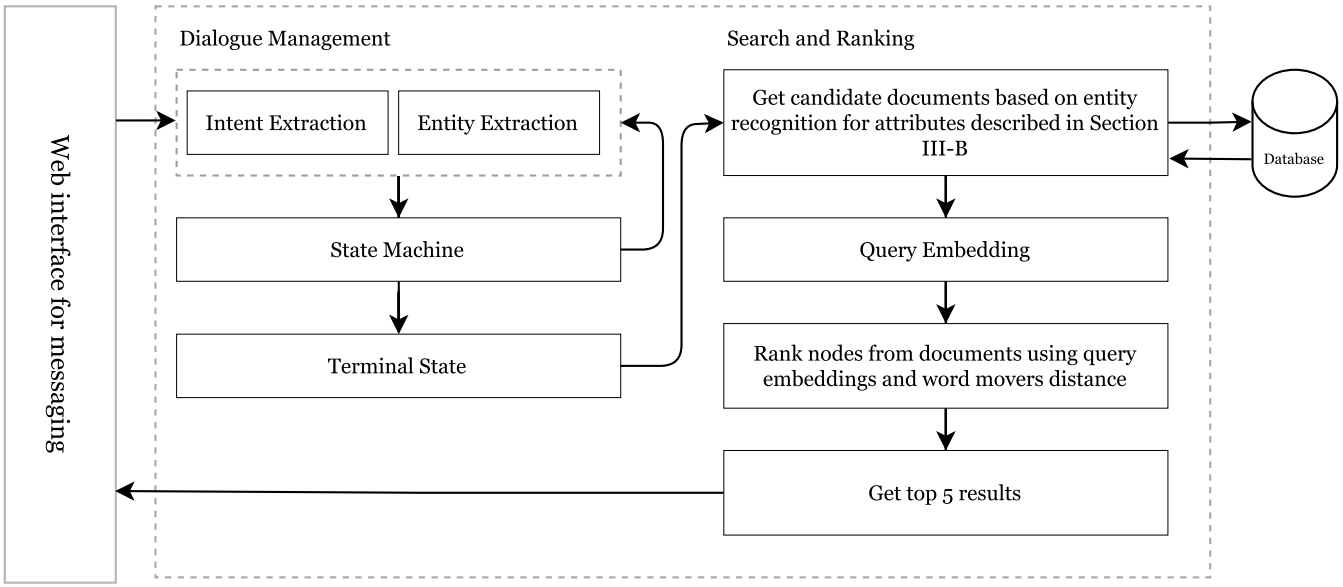

FIGURE 1. Architecture of the FSM Bot-initiative chatbot.

3) **Business Function**: The business function from which the user wants information. Hereby, the business function is referred to as just 'function.'
4) **Files edited or created on a specific date/month/year**: The user can ask the chatbot to display files or information only from those modified or created on a particular day, month, or year.
5) **Type of file**: Users can ask the bot to search for files or information from specific file types.
6) **Domain-specific keywords**: keywords that are specific to the organization. These may include abbreviations, words specific to the domain of the documents, etc.

The category, country, function, and date, if specified, were used to filter documents to extract information. If specified, any category, country, or function is used in the search model. Three situations arise when selecting what to use as a query for the search model. First, the keywords extracted from user utterances were used as a queries. Second, if none of the entities are detected in the utterance, the entire utterance is used as the query. Third, suppose any other entity is detected, and no keyword entity is present. The detected entity words are removed from the utterance, and the edited utterance is used as the query.

The context is maintained if a specific category, country, or function is specified. Any of these entities, if specified, is stored in a key-value pair by the state machine. Here, the key is the entity's name, and the value is the word extracted from the utterance corresponding to that entity. These key-value pairs are then used to filter the search results.

C. CONVERSATION

In each state machine's start state, the chatbot informs the user what it is capable of and suggests that the user inputs something for searching. Each step in the conversation is modeled as a state in a finite state machine. The chatbot decides when to move to another state, depending on the current state's conditions. The dialogue manager maps from the state to the function call in that state. This allows the dialogue manager to pass the user's utterance to the function associated with that state.

D. STATE TRANSITIONS

The next state is triggered by the current state as and when the purpose of the current state is fulfilled. When a state can transition to multiple states, the next state is chosen based on filtering out which meta-data will yield fewer results. We counted the results for each category, country, and function. For example, if the counts are:

$$CategoryA = 10 CategoryB = 40$$
$$CountryA = 20 CountryB = 30$$
$$FunctionA = 5 FunctionB = 45$$

The bot now selects which filter to ask from the user based on each meta-data's standard deviation. Typically, we computed the standard deviation from a given sequence using the formula $\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)}$. Using this for the counts above, we obtain:

$$\sigma(CategoryA, CategoryB) = 15.0$$
$$\sigma(CountryA, CountryB) = 5.0$$
$$\sigma(FunctionA, FunctionB) = 20.0$$

If we filter by country, there is a better chance of obtaining fewer final results. Thus, the bot transitions into a *country_response* state.

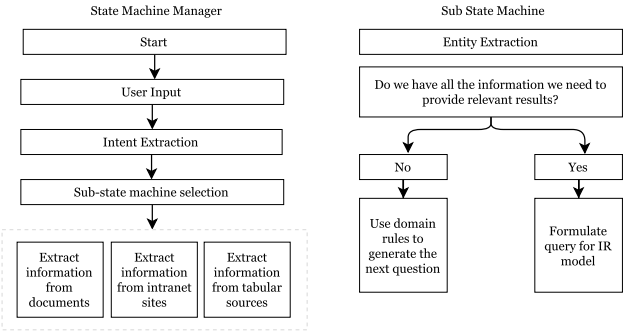

FIGURE 2. The figure above shows the global state machine manager and the selection of a sub-state machine. The flow of information in a sub-state machine is also shown.

If the user mentions meta-data to filter by, the second point for calculating the standard deviation is zero.

The search results are filtered each time a new entity is detected. The user can filter out search results without repeatedly searching the system.

E. STATE MACHINE SWITCHING

When a state machine manager processes an intent, it triggers the corresponding state machine. Subsequently, it directs all utterances to the active machine. If a different intent is received while the state machine is active, it switches to the relevant one. If no intent is detected in a user's utterance, it is directed to the active state machine. Figure 2 shows the state transitions and flow of information through the state machines.

IV. INFORMATION SEARCH MODEL

A. DATA

The data used to develop and test the models were documents collected from a private multinational bank. All documents were stored on internal cloud storage. The shared cloud storage has different folders that contain files for other categories. Each file in a category was created for a specific function and country. Highly domain-specific files contain words and abbreviations specific to the category, country, and function for which they contain information. Most documents contain a vast amount of numerical information, and the most descriptive parts are in document headers.

B. DATA COLLECTION, PARSING AND LABELLING

We developed a crawler to navigate various parts of the cloud storage, extract information from Microsoft Word and PDF documents, and generate metadata for each file. We employed a tree data structure to organize the data for efficient search. This choice was tailored to the intended use of our search model, which involves retrieving and displaying information from files or their paths that closely match the user queries.

Each document consists of sections labeled with headings, subheadings, and sub-subheadings, referred to as Level 1, Level 2, and Level 3 headings respectively. The root node of each document contains the full path and associated metadata of the file. Level 1 headings are children of the root node, level 2 headings under level 1 headings, and the same for level 3 headings. Each node stores a heading and the corresponding section text.

We used the documents collected by our crawler to prepare a dataset to train a model capable of categorizing user input queries by category, country, and function. As these documents were structured in a tree format, we created combinations of section headings with their root headings. Specifically, for each level 1 section, we generated combinations of lengths 1, 2, and 3, using headings from levels $(l+1)$ and $(l+2)$. Each combination served as a training data row.

The generated data were then normalized. Normalization of text for Natural Language Processing applications is a pre-processing step that modifies the text to standardize it for any downstream task. The text was normalized using the following modifications:

- Lowercasing all the characters
- Removing punctuations from the text

Some combinations were duplicated across the categories, countries, and functions. This was a multi-label classification problem for three different types of outputs, and the training dataset mapped each document to its category, country, and function. We addressed the nuances involving missing details by generating a ''no-data'' classification output. Figure 3 shows a representation of the document.

C. MODEL ARCHITECTURE

The model architecture is shown in Figure 4. This model uses BERT [6] for token encoding. BERT is a state-of-the-art encoder, which is a stack of Transformer Encoder layers. There are two variants of BERT, the BERT Base and BERT Large. The BERT Base has 12 stacked encoder layers, 12 attention heads, and a hidden state of dimension 768. In this model, we used the BERT architecture. The breakthrough for BERT is that it can be pre-trained and fine-tuned for downstream NLP tasks. All encoder layers in the BERT are trainable for downstream tasks. The last layer of the BERT Base was trained for the task.

BERT takes an input of sequence length n_l and outputs an embedding of $(n_l, 768)$. Figure 5 shows the encoder stack, which takes an input sequence length of 512 and outputs an embedding of dimension 768 for each token. The $(n_l, 768)$ embedding output is then passed through a 1D convolution layer. The 1D convolution over text can be considered by taking n-grams over words. Using a filter size of f is similar to taking f-grams over words. The filter size over volume becomes $(f, 768)$. For n_f number of filters, the output of this layer is thus of shape $(n_l - f + 1, n_l)$. The 1D max pooling layer uses a filter size of $(n_l - f + 1)$, the output of which is (n_f). This vector is then used as an input to three sub-networks, that perform multi-label classification for category, country, and function output. Our model uses:

$$n_l = 64, \quad f = 3, \quad n_f = 256.$$

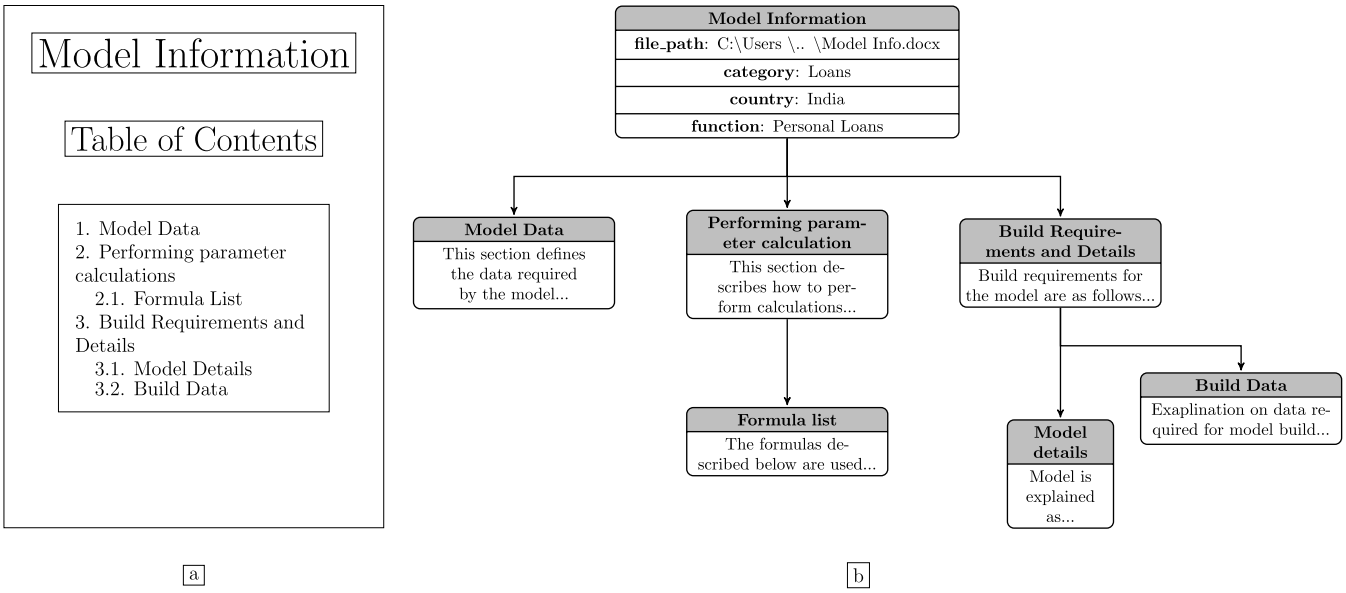

FIGURE 3. (a) Shows an example document with its name and the table of contents. Each topic in the table of contents is considered a node with a section of text information under that topic. (b) The document in (a) is parsed in a tree structure. The root node holds the file name along with the file meta-data. Each subsequent sub-topic forms a child node.

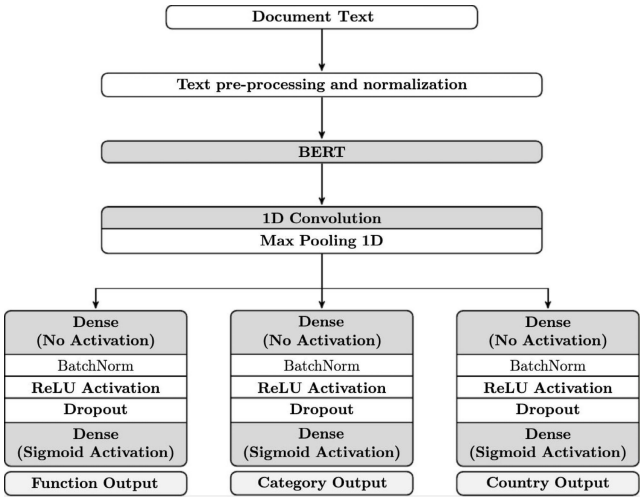

FIGURE 4. Proposed model: BERT + Convolutional + Branched Neural Network Model.

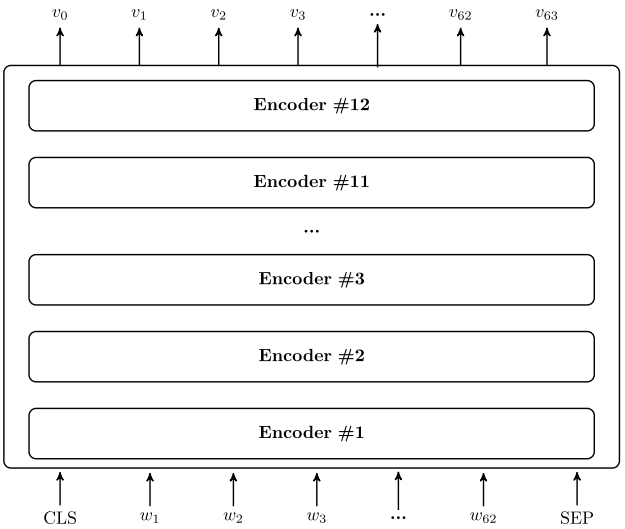

FIGURE 5. Stacked Encoder Layers. CLS and SEP are two special tokens specifying the sequence's beginning and end.

The output of the dense layer in each sub-network is passed through a sigmoid activation function. The sigmoid function models each logit as an independent probability distribution. Thus, the output of each sub-network is the probability of categories, countries, and portfolios. A learning rate of 1e-4 with a decay of 1e-6 was used. The Adam optimizer [50] was used for optimization. We used the sigmoid binary cross-entropy loss, which is defined as:

$$loss(x, z) = \frac{1}{n}\sum_{i=1}^{n} z_i \times -log(sigmoid(x_i)) + (1 - z_i) \times -log(1 - sigmoid(x_i)) \quad (1)$$

where, $x = logits$, $z = labels$.

D. WORD REPRESENTATION

We use word embeddings derived from the first sequence output of the last encoder layer (layer 11) of the BERT Base when only the word w_i is input as a sequence to the BERT model. Thus, each word i is represented by vector $\mathbf{w}^i \in \mathbb{R}^{768}$.

The learned distributed representations were used for similarity measurements in the representation space. We generated a vocabulary of words in the training set, and each word was assigned a unique ID. The training dataset contained a vocabulary of 20,772 words. Each word was passed through the BERT network, and its representation was recorded. Thus, a functional lookup table is obtained for each word in its representation.

E. CANDIDATE DOCUMENTS

The model described in Section IV-C was used to obtain the probability of a query belonging to a particular category, country, or function. The outputs are the probabilities that the query belongs to each label for each output type. We take the two highest probabilities for each category, country and function and use these probabilities to filter the set of available documents. These documents, filtered by category, country, and function, served as candidate documents for similarity measurements.

F. DOCUMENT REPRESENTATION

Each document is represented as a tree, as explained in Section IV-B. Each node other than the root node has a heading and text corresponding to information about the heading. Each heading h_i is a sequence of words in which each word has a unique id, $h_i = \{\psi(w_1), \psi(w_2), \psi(w_3), \ldots, \psi(w_n)\}$ where, $\psi : w \mapsto h$ is a mapping of the word to its unique id. This sequence is represented as a sequence of embeddings, $H_i = \{\phi(w_1), \phi(w_2), \phi(w_3), \ldots, \phi(w_n)\}$. For each root node, the representation is generated using the text sequence from the document's file name, where ϕ maps the unique word ID to word embedding. Thus, each node of every document is represented by $\mathbf{v} \in \mathbb{R}^{n\times 768}$, where n is the number of word tokens in the heading or name of the document.

V. SIMILARITY COMPUTATION

To rank the text or document most relevant to the user query, we need to measure the similarity between the query and each node in each candidate document. There are several methods for calculating the similarity between two sequences of word embeddings; we will experiment and work with two of them.

A. COSINE SIMILARITY

Given a query representation $q = \{q_1, q_2, q_3 \ldots q_n\}$ and a node representation $n = \{n_1, n_2, n_3 \ldots n_m\}$, the cosine similarity between the two is calculated as shown in Equation 2.

$$sim(q, n) = \frac{centroid(\vec{q}).centroid(\vec{n})}{||centroid(\vec{q})||||centroid(\vec{n})||} \tag{2}$$

where,

$$centroid(\mathbf{v}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{v_i} \tag{3}$$

Intuitively cosine similarity works because when word embeddings are learned in the embedding space, words of a similar category, country, and function are clustered closer to each other in the embedding space. The centroids of the query and node sequences are calculated. These two vectors are close to the clusters of their respective embeddings. Therefore, the higher the $sim(q, n)$ the more relevant are the query and node.

B. WORD MOVERS DISTANCE

Kusner et al. [51] developed the Word Movers Distance (WMD) as a metric to measure the similarity between two text documents. This method uses word embeddings generated by a neural network architecture rather than traditional approaches to calculate similarity. Although any word embedding can measure document similarity with the algorithm, the original paper uses Word2Vec embeddings. WMD measures the similarity or dissimilarity between two documents as the minimum distance traveled by the words in one document to reach the terms in other document's.

For the given two-word embeddings $\mathbf{x}_i$ and $\mathbf{x}_j$, the cost of moving from one word to another is given by

$$c(i, j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2 \tag{4}$$

Given two documents $\mathbf{d}$ and $\mathbf{d}'$, each word i in $\mathbf{d}$ can be transported to each word j in $\mathbf{d}'$ in total or in parts. Let $\mathbf{T} \in \mathbb{R}^{n\times n}$ be a sparse flow matrix where $\mathbf{T}_{ij} > 0$ denotes the amount of word $i \in \mathbf{d}$ flowing into word $j \in \mathbf{d}'$. The minimum cost of moving from $\mathbf{d}$ to $\mathbf{d}'$ is obtained by solving the following optimization problem:

$$\begin{aligned} \underset{\mathbf{T}\geq 0}{\text{minimize}} & \sum_{i,j=1}^{n} \mathbf{T}_{ij} c(i, j) \\ \text{subject to} & \sum_{j=1}^{n} \mathbf{T}_{ij} = d_i, \quad i = 1, \ldots, n \\ & \sum_{i=1}^{n} \mathbf{T}_{ij} = d'_j, \quad j = 1, \ldots, n \end{aligned} \tag{5}$$

where,

$$d_i = \frac{c_i}{\sum_{j=1}^{n} c_j} \tag{6}$$

where c_i is the term frequency of the ith word in document d. Our implementation uses word embeddings derived from BERT to measure WMD similarity.

Thus, the word mover distance is a constrained optimization problem that determines the optimum flow matrix T. The word movers distance is a particular case of the earth movers distance computation [52], [53].

VI. EXPERIMENTS

We compared the relevance of the generated search results and the efficiency of ranking the search results given a query for the BM25 model, BERT+Cosine similarity, and BERT+WMD models. The query uses pre-processing similar to document pre-processing, but we also removed the stopwords in this case.

EXPERIMENTAL SETUP

We created a dataset of distinct queries requested by users from user logs generated by the chatbot. Using 100 random queries from this set of queries, we generated the top 10 search results from the BM25, BERT+Cosine, and BERT+WMD ranking schemes. Human evaluators then

TABLE 1. Comparision of using BERT embeddings and Elmo embeddings with the branched neural network for multi-label classification of three different categories of classes. The highest metric values are shown in bold.

Classification Model	Classification Metric	Category Output	Country Output	Function Output
ElMO + Conv + Branched FC	True Positives	0.874	0.814	0.862
	F1	0.894	**0.837**	0.876
BERT + Conv + Branched FC	True Positives	**0.893**	**0.827**	**0.893**
	F1	**0.895**	0.834	**0.901**

TABLE 2. The DCG scores of the three search ranking models in comparison. DCG@X states that top X results were used for calculating the DCG score. * Each of these models ranks the files filtered using the predictions from the prediction model.

Search Ranking Model	DCG@3	DCG@5	DCG@10
BM25	51.349	66.689	87.407
BERT Embeddings + Cosine Similarity*	46.080	59.643	83.232
BERT Embeddings + Word Movers Distance*	**51.603**	**68.504**	**91.398**

assigned a relevance score to each search result from each query. The score is 0-5, where 5 is Perfect, 4 is Excellent, 3 is Fair, 2 is Poor, 1 is Bad and 0 is Irrelevant. These scores are then used to calculate each ranking model's Discounted Cumulative Gain (DCG). In the search result ranking from the BM25 algorithm, we consider all documents from the dataset as candidate documents. The candidate documents, which are to be ranked by BERT embeddings + cosine similarity and BERT embeddings + WMD, are generated by filtering the set of documents using the categories, country, and function of a document.

VII. RESULTS AND DISCUSSION

Table 1 lists the F1 and true positive scores for the multi-label classification of each metadata category of the document. The true positive score must be maximized because these predicted values filter the records for other similarity measurements. We also compared the classification model by using Elmo embeddings. Table 1 shows that BERT embeddings provide considerably better results.

Table 2 shows DCG scores for BM25, BERT embedding + cosine similarity, and BERT embeddings + WMD. We use BM25 as a baseline to measure the ranking performance with BERT embeddings.

The results in Table 2 show that BERT+WMD outperformed BM25 resulting in a significant margin of the DCG metric. However, cosine similarity shows inferior results for both DCG metrics, suggesting that they neither produce very relevant results nor rank them well.

We found that BERT embeddings with WMD provided better relevance by analyzing the queries and their ranked results. Although an exact word is not present in the result, a sequence with words with better semantic similarity to the query word is higher. The drawback of any ranking with BERT embedding is that if a word is absent in the training dataset or rarely occurs, the model needs to fine-tune the word's embeddings, thus showing poor search results. BM25, on the other hand, can score a text with a rarely occurring word higher than other texts in which it does not appear.

Therefore, we propose using a mixture model with BM25 and word movers distance. The mixture model ranks the filtered results from the predictive model.

$$SCORE(q, d) = \alpha WMD_{filtered}(q, d) + (1 - \alpha)BM25(q, d) \tag{7}$$

where α is a user-tuneable parameter.

VIII. CONCLUSION

This study evaluated the use of conversational agents and neural word embeddings to search for information in a closed-domain knowledge base. We show that BERT embeddings can produce good results when used with Word Movers Distance to find the similarity between a query and a document. We also present a method to telescope search results using document meta-data and entity recognition from user input. It has been shown that fine-tuning BERT is inexpensive and requires far less training data than training a model from scratch. In addition, our architecture matches the query to the document text instead of generating it, thus returning to the user the verbatim text from the document along with the document's source. This also allowed the system to be transparent. Future work can explore more advanced neural architectures such as retrieval-augmented generation models to improve the relevance and provide abstractive summaries. Investigating techniques for explainable information retrieval to provide a rationale for the retrieved documents is also a promising direction. Overall, our architecture enables organizations to build contextual and

transparent information retrieval systems in addition to large knowledge bases in a resource-efficient manner.

REFERENCES

[1] C. P. Medina and M. R. R. Ramon, "Using TF-IDF to determine word relevance in document queries," in *Proc. 1st Instructional Conf. Mach. Learn.*, vol. 242. Piscataway, NJ, USA, 2003, pp. 133–142.
[2] S. Robertson and H. Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends® Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.
[3] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
[4] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1532–1543.
[5] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. NAACL*, New Orleans, LA, USA, 2018, pp. 2227–2237. [Online]. Available: http://aclweb.org/anthology/N18-1202
[6] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol. NAACL-HLT*, vol. 1, Minneapolis, MN, USA, J. Burstein, C. Doran, and T. Solorio, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 4171–4186, doi: 10.18653/v1/n19-1423.
[7] B. Mitra and N. Craswell, "Neural models for information retrieval," 2017, *arXiv:1705.01509*.
[8] H. Zamani, "Neural models for information retrieval without labeled data," Ph.D. dissertation, College Inf. Comput. Sci., Univ. Massachusetts, Amherst, MA, USA, 2019.
[9] I. Matveeva, C. Burges, T. Burkard, A. Laucius, and L. Wong, "High accuracy retrieval with multiple nested ranker," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2006, pp. 437–444.
[10] R. Kruiper, I. Konstas, A. Gray, F. Sadeghineko, R. Watson, and B. Kumar, "Document and query expansion for information retrieval on building regulations," in *Proc. 30th EG-ICE Int. Conf. Intell. Comput. Eng.*, May 2023, pp. 1–12. [Online]. Available: https://www.ucl.ac.uk/bartlett/construction/research/virtual-research-centres/institute-digital-innovation-built-environment/30th-eg-ice
[11] S. Seneff and J. Polifroni, "Dialogue management in the mercury flight reservation system," in *Proc. ANLP/NAACL Workshop Conversational Syst.* Stroudsburg, PA, USA: Association for Computational Linguistics, 2000, pp. 11–16.
[12] M. K. Chinnakotla and P. Agrawal, "Lessons from building a large-scale commercial IR-based chatbot for an emerging market," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 1361–1362.
[13] J. Yu, M. Qiu, J. Jiang, J. Huang, S. Song, W. Chu, and H. Chen, "Modelling domain relationships for transfer learning on retrieval-based question answering systems in e-commerce," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*. New York, NY, USA: ACM, 2018, pp. 682–690.
[14] O. Vinyals and Q. Le, "A neural conversational model," 2015, *arXiv:1506.05869*.
[15] T. Wang, M. Cai, and J. Li, "A neural conversational model using MMI-WMD decoder based on the Seq2Seq with attention mechanism," in *Proc. Chin. Control And Decis. Conf. (CCDC)*, 2019, pp. 2696–2700.
[16] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 3104–3112.
[17] P. Liu, S. Chang, X. Huang, J. Tang, and J. C. K. Cheung, "Contextualized non-local neural networks for sequence learning," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 6762–6769.
[18] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. (2018). *Improving Language Understanding By Generative Pre-training*. Accessed: Dec. 16, 2023. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
[19] OpenAI, "GPT-4 technical report," 2023, *arXiv:2303.08774*.
[20] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst. Annu. Conf. Neural Inf. Process. Syst.*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. 2020, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
[21] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-tuning language models from human preferences," 2019, *arXiv:1909.08593*.
[22] J. MacGlashan, M. K. Ho, R. T. Loftin, B. Peng, G. Wang, D. L. Roberts, M. E. Taylor, and M. L. Littman, "Interactive learning from policy-dependent human feedback," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, Sydney, NSW, Australia, D. Precup and Y. W. Teh, Eds. 2017, pp. 2285–2294. [Online]. Available: http://proceedings.mlr.press/v70/macglashan17a.html
[23] Y. Bai et al., "Training a helpful and harmless assistant with reinforcement learning from human feedback," 2022, *arXiv:2204.05862*.
[24] G. Warnell, N. R. Waytowich, V. Lawhern, and P. Stone, "Deep TAMER: Interactive agent shaping in high-dimensional state spaces," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI-18)*, New Orleans, LA, USA, S. A. McIlraith and K. Q. Weinberger, Eds. Palo Alto, CA, USA: AAAI Press, Feb. 2018, pp. 1545–1554, doi: 10.1609/aaai.v32i1.11485.
[25] B. J. Gutierrez, N. McNeal, C. Washington, Y. Chen, L. Li, H. Sun, and Y. Su, "Thinking about GPT-3 in-context learning for biomedical IE? Think again," in *Proc. Findings Assoc. Comput. Linguistics (EMNLP)*, Abu Dhabi, United Arab Emirates, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, 2022, pp. 4497–4512, doi: 10.18653/v1/2022.findings-emnlp.329.
[26] K. Mrini, M. Laperrouza, and P. Dillenbourg, "Building a question-answering chatbot using forum data in the semantic space," in *Proc. 3rd Swiss Text Anal. Conf.-SwissText*, Winterthur, Switzerland, Jun. 2018, pp. 1–6. [Online]. Available: http://infoscience.epfl.ch/record/256467
[27] W. Yang and J. Wang., "Generating appropriate question-answer pairs for chatbots using data harvested from community-based QA sites," in *Proc. 9th Int. Joint Conf. Knowl. Discovery, Knowl. Eng. Knowl. Manage.*, Madeira, Portugal, 2017, pp. 342–349.
[28] J. D. Williams, K. Asadi, and G. Zweig, "Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Vancouver, BC, Canada, R. Barzilay and M. Kan, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, Jul. 2017, pp. 665–677, doi: 10.18653/v1/P17-1062.
[29] S. Yi and K. Jung, "A chatbot by combining finite state machine, information retrieval, and bot-initiative strategy," in *Proc. Alexa Prize*, 2017, pp. 1–10.
[30] J. Weizenbaum, "ELIZA—A computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966, doi: 10.1145/365153.365168.
[31] R. Wallace, "The elements of AIML style," 2003. Accessed: Nov. 20, 2023. [Online]. Available: https://files.ifi.uzh.ch/cl/hess/classes/seminare/chatbots/style.pdf
[32] D. Braun, A. Hernandez-Mendez, F. Matthes, and M. Langen, "Evaluating natural language understanding services for conversational question answering systems," in *Proc. 18th Annu. SIGdial Meeting Discourse Dialogue*, Saarbrucken, Germany, 2017, pp. 174–185.
[33] A. Saini, A. Verma, A. Arora, and C. Gupta, "Linguistic rule-based ontology-driven chatbot system," in *Advances in Computer Communication and Computational Sciences*, S. K. Bhatia, S. Tiwari, K. K. Mishra, and M. C. Trivedi, Eds. Singapore: Springer, 2019, pp. 47–57.
[34] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, Dec. 2016.
[35] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, San Diego, CA, USA, K. Knight, A. Nenkova, and O. Rambow, Eds. 2016, pp. 260–270, doi: 10.18653/v1/n16-1030.
[36] Y. Song, C.-T. Li, J.-Y. Nie, M. Zhang, D. Zhao, and R. Yan, "An ensemble of retrieval-based and generation-based human-computer conversation systems," in *Proc. 27th Int. Joint Conf. Artif. Intell. (IJCAI-18)*, Jul. 2018, pp. 4382–4388, doi: 10.24963/ijcai.2018/609.
[37] Z. Yan, N. Duan, J. Bao, P. Chen, M. Zhou, Z. Li, and J. Zhou, "DocChat: An information retrieval approach for chatbot engines using unstructured documents," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, Berlin, Germany, Aug. 2016, pp. 516–525.
[38] P. Crovari, S. Pidò, P. Pinoli, A. Bernasconi, A. Canakoglu, F. Garzotto, and S. Ceri, "GeCoAgent: A conversational agent for empowering genomic data extraction and analysis," *ACM Trans. Comput. for Healthcare*, vol. 3, no. 1, pp. 1–29, Jan. 2022, doi: 10.1145/3464383.

[39] Y. Gaci, J. Ramirez, B. Benatallah, F. Casati, and K. Benabdeslem. (2021). *Subjectivity Aware Conversational Search Services*. [Online]. Available: https://openproceedings.org/2021/conf/edbt/p70.pdf
[40] J. L. Z. Montenegro and C. A. da Costa, "The HoPE model architecture: A novel approach to pregnancy information retrieval based on conversational agents," *J. Healthcare Informat. Res.*, vol. 6, no. 3, pp. 253–294, Sep. 2022. [Online]. Available: https://link.springer.com/10.1007/s41666-022-00115-0
[41] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, Hong Kong, K. Inui, J. Jiang, V. Ng, and X. Wan, Eds. Stroudsburg, PA, USA: Association for Computational Linguistics, Nov. 2019, pp. 3980–3990, doi: 10.18653/v1/D19-1410.
[42] Y. Wu, W. Wu, C. Xing, C. Xu, Z. Li, and M. Zhou, "A sequential matching framework for multi-turn response selection in retrieval-based chatbots," *Comput. Linguistics*, vol. 45, no. 1, pp. 163–197, Mar. 2019, doi: 10.1162/coli_a_00345.
[43] M. M. A. Pour, P. Farinneya, A. Toroghi, A. Korikov, A. Pesaranghader, T. Sajed, M. Bharadwaj, B. Mavrin, and S. Sanner, "Self-supervised contrastive BERT fine-tuning for fusion-based reviewed-item retrieval," in *Proc. Adv. Inf. Retr. 45th Eur. Conf. Inf. Retr. (ECIR)*, in Lecture Notes in Computer Science, Dublin, Ireland, J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, and A. Caputo, Eds. Cham, Switzerland: Springer, vol. 13980, 2023, pp. 3–17, doi: 10.1007/978-3-031-28244-7_1.
[44] N. Rachmawati and E. Yulianti, "Transfer learning for closed domain question answering in COVID-19," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 12, 2022. [Online]. Available: http://thesai.org/Publications/ViewPaper?Volume=13&Issue=12&Code=IJACSA&SerialNo=34
[45] K. Ezhilarasi, D. Mansoor Hussain, M. Sowmiya, and N. Krishnamoorthy, "Crop information retrieval framework based on LDW-ontology and SNM-BERT techniques," *Inf. Technol. Control*, vol. 52, no. 3, pp. 731–743, Sep. 2023. [Online]. Available: https://itc.ktu.lt/index.php/ITC/article/view/31945
[46] G. Larionov, Z. Kaden, H. V. Dureddy, G. Bayomi T. Kalejaiye, M. Kale, S. P. Potharaju, A. P. Shah, and A. I Rudnicky, "Tartan: A retrieval-based socialbot powered by a dynamic finite-state machine architecture," 2018, *arXiv:1812.01260*.
[47] B. Mitra, E. Nalisnick, N. Craswell, and R. Caruana, "A dual embedding space model for document ranking," 2016, *arXiv:1602.01137*.
[48] X. Wu and H. Li, "Topic mover's distance based document classification," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2017, pp. 1998–2002.
[49] S. L. van der Pas and A. W. van der Vaart, "Bayesian community detection," *Bayesian Anal.*, vol. 13, no. 3, pp. 767–796, Sep. 2018, doi: 10.1214/17-BA1078.
[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[51] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger, "From word embeddings to document distances," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 957–966.
[52] O. Pele and M. Werman, "Fast and robust Earth mover's distances," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 460–467.
[53] O. Pele and M. Werman, "A linear time histogram metric for improved sift matching," in *Computer Vision–ECCV 2008*, D. Forsyth, P. Torr, and A. Zisserman, Eds. Berlin, Germany: Springer, 2008, pp. 495–508.

SHUBHAM RATERIA received the bachelor's degree in technology in computer and communication engineering from Manipal Institute of Technology, Manipal, India, in 2019. Then, he joined Hong Kong and Shanghai Bank Corporation, Bengaluru, as a Machine Learning Engineer to work on large-scale information search. Later, he joined Radical Health, Delhi, India, as a Deep Learning Researcher. He currently leads Custiv Manufacturing as the CTO and the Director. His current research interests include manufacturing engineering, CAD/CAM engineering, distributed systems, deep learning, and machine learning.

SANJAY SINGH (Senior Member, IEEE) received the degree from the Institution of Electronics and Telecommunications Engineers, New Delhi, India, in 2001, and the M.Tech. and Ph.D. degrees from Manipal Institute of Technology, Manipal, India, in 2003 and 2010, respectively. In 2004, he joined the Department of Information and Communication Technology, Manipal Institute of Technology, MAHE, where he is currently a Professor. His research interests include artificial intelligence, machine learning, neural networks, fuzzy logic, and natural language processing.