

Artificial Intelligence-Empowered Chatbot for Effective COVID-19 Information Delivery to Older Adults

Xin Wang, North Dakota State University, USA

Tianyi Liang, North Dakota State University, USA

Juan Li, North Dakota State University, USA*

Souradip Roy, North Dakota State University, USA

Vikram Pandey, North Dakota State University, USA

Yang Du, Valley City State University, USA

Jun Kong, North Dakota State University, USA

ABSTRACT

The coronavirus disease 2019 (COVID-19) epidemic poses a threat to the everyday life of people worldwide and brings challenges to the global health system. During this outbreak, it is critical to find creative ways to extend the reach of informatics into every person in society. Although there are many websites and mobile applications for this purpose, they are insufficient in reaching vulnerable populations like older adults who are not familiar with using new technologies to access information. In this paper, the authors propose an AI-enabled chatbot assistant that delivers real-time, useful, context-aware, and personalized information about COVID-19 to users, especially older adults. To use the assistant, a user simply speaks to it through a mobile phone or a smart speaker. This natural and interactive interface does not require the user to have any technical background. The virtual assistant was evaluated in the lab environment through various types of use cases. Preliminary qualitative test results demonstrate a reasonable precision and recall rate.

KEYWORDS

Artificial Intelligence, Chatbot, COVID-19, Information Delivery, Knowledge Graph, Natural Language Processing

INTRODUCTION

The world has been facing an unprecedented health crisis caused by the COVID-19. Government-coordinated efforts across the globe have focused on coping with the COVID-19 situation. Based on currently available information and clinical expertise, older adults might be at higher risk for severe illness from COVID-19 (Xie et al., 2020). As COVID-19 is a new virus, the public has lots of questions and confusion about this disease. Therefore, it is critical to provide timely, accurate, and relevant information to the public, which can be the difference between life and death. With lockdowns and other social distancing measures in effect in many countries, more and more people are relying on the

DOI: 10.4018/IJEHMC.293285

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Internet and digital tools for information and advice. Digital information solutions (e.g., (Chakraborty et al., 2020; Chakraborty & Rodrigues, 2020; Dash et al., 2021)) can be invaluable in the prevention and management of COVID-19. Over the past months, we have witnessed the increasing adoption of mobile apps to disseminate COVID-19 related information and resources. However, older adults' adoption and use of digital technology lag that of younger cohorts. According to Norman & Skinner (Norman & Skinner, 2006), older adults' eHealth literacy, or the ability to access, assess, and use health information to make informed healthcare decisions tends to be low and requires extensive assistance. Compared with younger people, older adults are less likely to obtain high-quality information or obtain food, supplies, and services online (Xie et al., 2020). The complexity of these digital tools makes them intimidating for older adults who struggle to use a computer or smartphone due to inexperience with the computer.

In this paper, we present the design of a chatbot assistant that offers real-time, useful, context-aware, and personalized tips and information to users, especially older adults through automatic question answering. The chatbot assistant provides a natural, interactive interface for users, and it does not require the user to have a technical background. To empower the chatbot with a "knowledgeable brain", we construct a comprehensive knowledge graph that connects multiple knowledge sources related to COVID-19. A knowledge graph (KG) represents a collection of interlinked entities. The entities (nodes) in a KG represent real-world objects, while the links (edges) between two entities present the relationship between them. KG is vital in our system for information search, analytics, and recommendations. Moreover, we use KGs to represent our knowledge for the following reasons: (1) KGs have formal semantics that allows both people and computers to process them efficiently and unambiguously. (2) Questions/queries can be naturally mapped with KGs. (3) KGs can be easily expanded with new knowledge/data. Based on the knowledge graph, we build a flexible dialogue system to support task-oriented conversation that can recognize user's speech, identify user's intention, and provide suitable responses to users, using AI technologies including natural language processing, semantic search, logic reasoning, and machine learning. It can answer various questions about COVID-19, such as general questions like "Is COVID-19 vaccine safe?", "What are the symptoms of COVID-19?", "Is COVID-19 airborne?", etc.; or specific personal questions, like "Do I need to be tested for COVID-19?", "Where I can get tested for COVID-19?", "How many new cases of COVID-19 appear in my grandson's school this week?", etc.

Overall, the contributions of the paper are summarized as follows:

- We identified people's information need during the COVID -19 pandemic and identified and prepared comprehensive information datasets that have been used as knowledge sources to get people informed of this disease.
- We developed a chatbot prototype system using AI technologies including natural language processing, knowledge graph, logic reasoning, and machine learning. This chatbot aims to assist people, especially older adults, to get timely information about COVID-19. Design decisions were made considering older adults' special needs.
- We evaluated the proposed chatbot with a set of qualitative evaluations.

The rest of the paper is organized as follows. Section 2 surveys related work on the voice-based agent or chatbot for healthcare and Covid-19 fighting. Section 3 describes in detail our proposed methodology. Section 4 presents our evaluation results. Finally, Section V provides conclusions and future work directions.

Background and Related Work

Inspired by the Turing test (Saygin et al., 2000), lots of chatbots or conversational agents (e.g., ELIZA (Weizenbaum, 1966), Parry (Cerf, 1973), and ALICE (Shawar & Atwell, 2015)) have been developed. We adopt many fundamental technologies of existing chatbot systems. Pattern matching

is important for chatbots. A pattern is a regularity in the world, which may repeat in a predictable manner. Systems then can take advantage of patterns to build repetitive behaviors the chatbots. Eliza and ALICE were the earliest chatbots using pattern recognition algorithms. Many recent chatbots also used pattern matching as a main technology. For example, Setiaji et al. (Setiaji & Wibowo, 2016) proposed a chatbot that stores the chat patterns modeled in the pattern-template in a relational database management system (RDBMS) table. The database has been employed as knowledge storage for pattern-matching requirement. Another chatbot for university related FAQs (Setiaji & Wibowo, 2016) performs template pattern matching between user entered query and knowledge (pattern).

The Artificial Intelligence Markup Language (AIML) is based on pattern matching technique. It has been applied to natural language modeling for the dialogue between humans. Latent Semantic Analysis (LSA) can be used together with AIML to discover likenesses between words as vector representation (J. W. Wei et al., 2020). Many chatbots, for example, the university FAQ bot (Ranoliya et al., 2017) and the e-business chatbot (Thomas, 2016) can answer template-based questions like greetings and general questions AIML and user LSA to answer other.

Natural Language Understanding (NLU) is a core technique to implement natural communication between human and a chatbot. It aims to extract context and meanings from unstructured natural language user inputs. The meaning will be mapped to (predefined) user intention (Jung, 2019). Intent detection normally processed using classification techniques in which intent labels are predicted for user input (Strobl et al., n.d.).

More recently, knowledge graphs have been utilized for assisting chatbot to acquire external knowledge. A knowledge graph can link concepts/words based on their relationships. It allows computers to store, navigate and reasoning over facts. A few chatbots have been designed using knowledge graph. For example, HHH (Bao et al., 2020) is an online healthcare question-and-answer (QA) chatbot that answers users' medical questions. HHH chatbot answer questions based on a knowledge graph constructed from medical data collected from the Internet. It uses a deep learning model, Hierarchical BiLSTM Attention Model (HBAM), to find the most similar question from a large QA dataset. Kbot (Ait-Mlouk & Jiang, 2020) is another chatbot that uses a knowledge graph as background knowledge source and convert a user's query to a SPARQL query to search the knowledge graph and get answers.

The potential of voice-enabled devices in healthcare has been noticed. Many voice technologies have been used to automate or simplify communication between patients and providers. Voice-enabled assistants are being used in appointment scheduling and reminders in an outpatient setting, or care team coordination in an inpatient setting. For example, Merit.ai (*Merit.Ai*, n.d.) provides automatic voice assistance for scheduling, rescheduling, and cancellation appointment for patients. VoiceFriend (*VoiceFriend*, n.d.) provides a notification service to family members and the medical team about the patient's information. Kiroku (*Kiroku*, n.d.) can pick up the context in a conversation between the doctor and the patient and automatically write clinical notes for the doctor.

Amazon has launched HIPAA compliant health care skills, which allow protected health information to be transmitted without violating HIPAA Rules. Express Scripts (*Express Scripts*, n.d.) is a HIPAA-compliant skill that can check the status of a home delivery prescription and send notifications when prescriptions have been shipped and when they arrive at the user's door. My Children's Enhanced Recovery After Surgery (ERAS) (*My Children's Enhanced Recovery After Surgery (ERAS)*, n.d.) developed by Boston Children's Hospital allows parents to update the care team in the hospital about the recovery progress after a child has gone through heart surgery. It can also remind parents of their child's scheduled appointment. 'Answers By Cigna' (*Answers By Cigna*, n.d.) is another Alexa skill developed to reduce the existing gap between the general people and healthcare professionals. It can explain the meaning of a vocabulary that one may find unfamiliar with. Alexa skill 'Livongo Blood Sugar Lookup' (*Livongo Blood Sugar Lookup*, n.d.) allows the user to ask their last blood sugar reading. Skills such as 'Express Script' (*Express Scripts*, n.d.) and 'Swedish Health

Connect' (*Swedish Health Connect*, n.d.) can help people to order their medicines and set the next appointment with the health service provider.

AI-powered voice agents have been employed in managing patients' needs during the pandemic. For example, CDC has created a COVID-19 assessment chatbot (*CDC COVID-19 Assessment Chatbot*, n.d.) to help people identify coronavirus symptoms and provide advice on whether they should seek medical assistance. This Self Checker bot asks a series of questions based on CDC guidelines about symptoms (such as coughing or shortness of breath) and risk factors (like age, underlying medical conditions, or potential exposure to the virus) and suggests next steps, including whether to self-isolate, consult with telehealth professional, or visit an emergency room. The CDC chatbot can only answer questions regarding COVID-19 symptoms.

During the pandemic, UCHealth used Livi (*Livi - UCHealth*, n.d.), a virtual assistant powered by AI to help patients send messages to their providers, find COVID test results or check in for upcoming appointments. Livi is good at getting patients' information like test results, and helping patients message their doctors; However, Livi cannot answer a specific question about COVID-19. Walgreens has introduced a voice virtual assistant to help people schedule COVID-19 vaccine appointments (*Walgreens Voicebot.Ai*, n.d.). The AI helps people set up their vaccine appointments, reducing Walgreens' overwhelming load of the huge number of incoming inquiries.

Most of the existing COVID-19 voice assistants are command-based simple agents. Based on a user's command, the voice assistant can perform certain operations like a real human assistant. Moreover, existing agents normally focus on one aspect of COVID-19 such as symptom, or test. Conversation-based assistants are more complex. It requires the agent to have a more comprehensive knowledge base about COVID-19 and provides more flexible and intelligent conversation management capability. Although some commercial conversational agents have been used in COVID-19 fighting, their technical details are not published in the scientific literature.

THE PROPOSED CHATBOT SYSTEM

Overview of the System

Figure 1 shows the overall architecture of the proposed chatbot system. The system gathers knowledge and information from multiple sources (ontologies, knowledge graphs, and data from web services). It is built using technologies of natural language processing, semantic web, knowledge graph, and machine learning. It consists of several major components, including speech and text conversion, intent classification, semantic entity and relation extraction, knowledge graph mapping, query results selection, and response generation. Users interact with the system through a chatting format using their voices. We employ Amazon Transcribe to do speech and text conversions. Amazon Transcribe uses a deep learning process called automatic speech recognition (ASR) to convert speech to text quickly and accurately. Other than ASR, we design and implement all the rest components of the system.

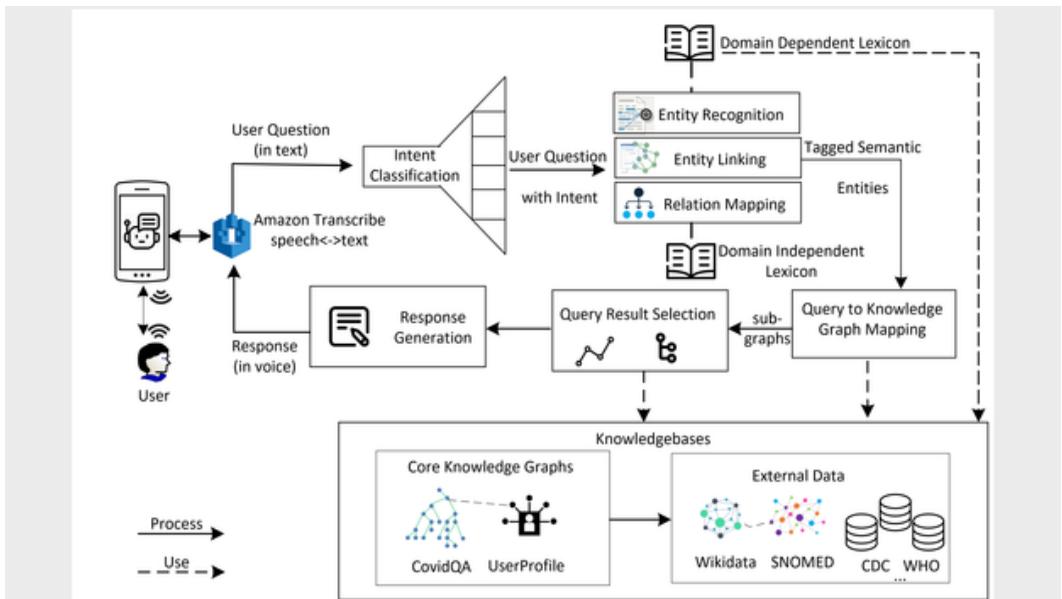
We made the design decisions based on older adults' need from the following aspects:

1. Knowledge preparation: when we constructed the knowledge graph and selected knowledge sources, we choose to use simple and easy to understand languages, because older adults normally have relatively lower health literacy.
2. Response generation: when we created the training dataset for natural language response generation, we used simple and easy to understand words (entities, relationships) to replace the triples in the knowledge graph. As a result, the responses generated by the chatbot would be easy to understand.
3. Mobile application interface: when we implemented the chatbot, we provided both voice and text input/output for convenient communication. Voice is especially valuable older adults who are

unable to use other forms of technology that may require fine motor skills, hand-eye coordination, and/or good vision. Our text input is configured with larger font size for older eyes.

More details about the design of the system are presented in the following subsections.

Figure 1. System architecture



Knowledge Preparation

The brain of the chatbot is a knowledge base that provides comprehensive and pragmatic answers to users' questions. Our knowledge base is constructed by linking several knowledge graphs and extend them with external datasets. KG can help the chatbot to understand the questions and find out answers by analyzing what the words in a query mean, rather than simply analyzing strings of characters. The core of this knowledge base is CovidQA, a KG that is created based on various frequently asked questions and answers (Q&A) about COVID-19 and existing ontology models related to COVID-19 (Dutta & DeBellis, 2020). CovidQA focuses on various information of Covid-19, such as cases across geolocations, symptoms, treatment, prevention, etc. This information is linked to a specific user by connecting it with the user's profile and context graph. The user profile graph records various aspects of a user's information such as the user's location, health status, age, preferences, and social network. The profile graph is constructed using a short survey given to the user when the user first using the system. The survey is conducted by the chatbot through chatting with the user. This user-specific KG can capture an individual user's profile in a structured manner. It is used for the personalization of the system, i.e., providing user-specific answers/recommendations to the user based on the user's characteristics.

Figure 2 and Figure 3 show part of the user-specific CovidQA knowledge graph used in our prototype system. Figure 2 is a schema-level graph that shows labels of each entity (i.e., Person, Disease, Vaccine, and Symptom) and the relationship between them (i.e., Person has_disease->Disease, Disease ->has symptom->Symptom). Figure 3 shows the detailed instance/individual level

St, Orlando, FL?” Given a detailed roadmap of America, it may be a computationally expensive exercise to compute the travel route with all possible roads. Therefore, instead of working at such a low level of detail, a human travel planner would prefer to do the route planning at the state-level first: i.e., finding a highway path from ND to FL first. We adopt a similar abstraction strategy for semantic entity linking. Instead of starting from numerous semantic entities and relationships at the instance graph (Figure 2 (b)), we link entity at the schema graph first (Figure 2(a)). Using the results from the schema graph, we can find related entities and paths at the instance level.

Given the set of entity mentions M , the goal of entity linking is to map each entity mention $m \in M$ to its corresponding entity $e \in E$ which is a set of entities defined in our KG. We adopt the name dictionary-based techniques to get candidate entities. Specifically, we construct a $\langle \text{Key}, \text{ValueList} \rangle$ pair dictionary with nodes' names in the KG as key and the corresponding URIs as value. Synonyms, *sameAs* nodes will be listed in the same *ValueList*. Therefore, one key may have multiple values. Based on the dictionary constructed in this way, string matching will be applied to choose candidate entities. Besides exact match, we also do proximity match of m and e . For example, e can be contained in or contains m . e matches the first few letters of all words in m . e shares several common words with m or e , and m has a strong string similarity.

The detection of predicates/relations from natural language is more difficult compared with the detection of nodes/entities. As the same predicate/relations can be expressed with totally different words. For example, “COVID-19 may cause headache” and “headache is a symptom of COVID-19” use different verbs although they imply the same meaning. To address this issue, we employ the pattern library extracted from the BOA framework (Gerber & Ngomo, 2012) to detect properties. The idea behind BOA is to extract natural-language patterns that represent predicates found on the Data Web from unstructured data by using background knowledge from the Data Web.

Knowledge Graph Mapping and Query Generation

After the text mentions have been mapped to the knowledge graph entities and properties, we can get the related subgraphs. Then the semantic representation of the query will be translated into a KG query. Again, the KG subgraphs can be obtained through a two-level hierarchical mapping, analogous to the route planning problem mentioned before. First, using entities obtained at the schema level to retrieve paths connecting them to form a subgraph at a higher level. This higher-level graph will act as the blueprint to match the lower-level instance graph. This two-level mapping approach may drastically reduce the size of a potentially huge search space in the lower level and remove (or reduce) the number of incorrect mappings.

We use a minimum spanning tree (MST) to model the subgraph mapping problem in the schema knowledge graph. Given a set of nodes and/or a few edges, to find a subset of the edges of a connected, edge-weighted undirected sub-graph that connects all the vertices (and/or edges) together, without any cycles and with the minimum possible total edge weight. As the size of the schema graph is far smaller than the instance graph, the cost of resolving the MST problem on the schema graph is lower than on the instance graph. After getting the MST from the schema graph, the selected subgraph/tree will be used as a pattern constraint to retrieve subgraphs from the instance graph. The constraints include each node's label and connecting edges between nodes at the instance graph. In our prototype system, we use Cypher (Francis et al., 2018), as the query language of KG which is stored in the Neo4j (Neo4j, 2012) graph database. The KG query generation process needs to generate three clauses including “match”, “where”, and “return” clauses. The connected paths linking entity and relationships are mapped to the match clause of the Cypher query. The attribute-value constraint will be used as the condition in the where clause. While the target entity is mapped to the return clause.

The following example illustrates this 2-level query resolution process. In this example, a user Bill asks the chatbot: “Where are the Covid test sites in my county?” Bill's profile has been stored in the KG. After mapping the question text with our KG, we could obtain entity mentions: “Bill” with label “Person”, “ND Testing Event” with label “Event”, and “County” as a label, and know

the question type is about “where (a location)”. Then, these entities’ labels will be brought into the schema graph as given conditions. By using the MST, we tend to find sets of paths. In each path, all the given vertices are fully connected with the minimum path. From this illustration, we will find the path shown in Figure 4. Based on the selected subgraph/tree, all the relationship information will be extracted from the path and utilized for querying the instance graph. Cypher Query Language provides many efficient procedures to expand the subgraph adhering to the label and relationship filters. Eventually, we obtain the instance path as shown in Figure 5. It shows that Sanford Health Athletic in North Dakota State University is a Covid test site in Bill’s County.

Figure 4. Paths found in Schema graph

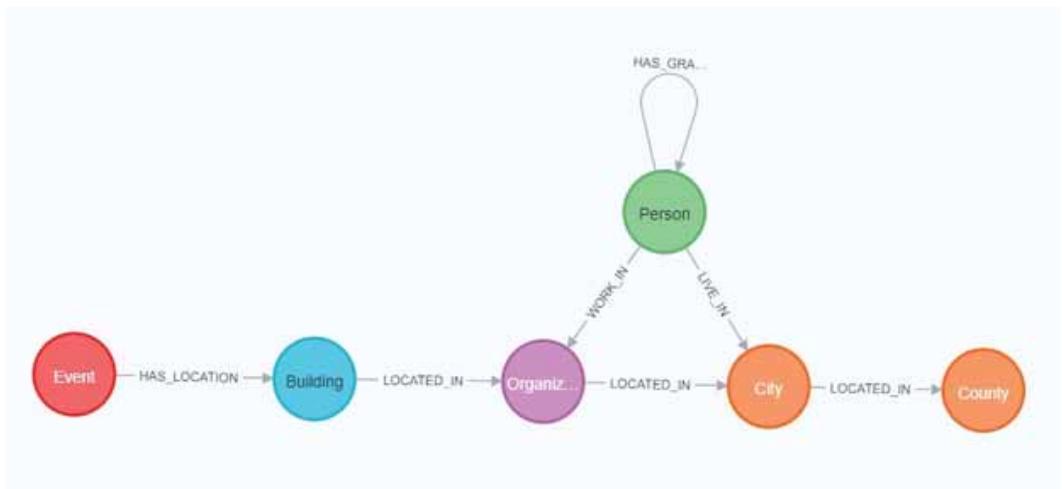
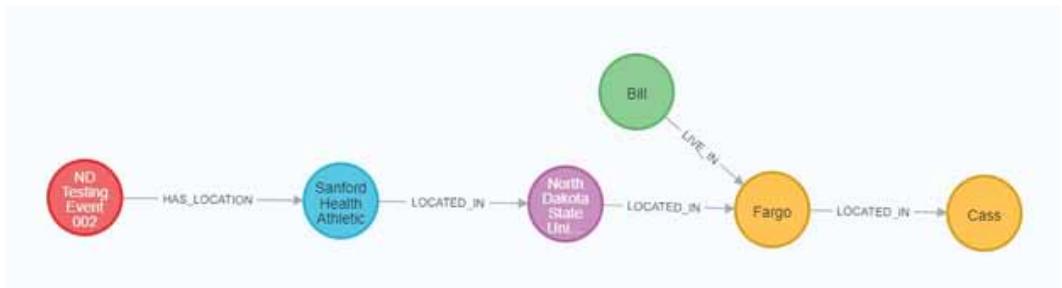


Figure 5. Path found in instance graph



Query Selection and Response Generation

A text question can be mapped to multiple KG queries. We need to determine which one among the multiple queries is the right one. For this purpose, we rank the KG queries based on their semantic distance to the original user’s natural language question and the category of the query intention. Given a user question and a set of potential target KG query candidates, the ranking of query candidates is based on the ranking of the relevance score of each of the candidates. The relevance score is computed

through the following steps: First, the preprocessed user question and all the candidate queries are encoded to a concept vector. A user question Q is modeled as a vector containing multiple concepts C_1, C_2, \dots, C_n .

$$Q = (C_1, C_2, \dots, C_n) \quad (1)$$

Next, the semantic distance between two concepts has been calculated based on common ontology defined in our knowledge graphs and WordNet (Li et al., 2020). We adopt a distance-based approach proposed in our previous work (Zaman & Li, 2014) to measure the semantic similarity between concepts. The basic idea is to identify the shortest path between two concepts in terms of the number of edges and then translate that distance into the semantic distance.

$$dis(C_a, C_b) = \frac{1}{2} \left(\frac{\sum_{i \in path(C_a, C_p)} w_i dis(C_i, C_{i+1})}{\sum_{i \in path(C_a, C_{root})} w_i dis(C_i, C_{i+1})} \right) + \quad (2)$$

$$\frac{1}{2} \left(\frac{\sum_{i \in path(C_b, C_p)} w_i dis(C_i, C_{i+1})}{\sum_{i \in path(C_b, C_{root})} w_i dis(C_i, C_{i+1})} \right)$$

Here, C_p is the nearest common parent of C_a and C_b . To calculate the distance between two concepts a weighting factor w has been used. To calculate this edge weight we used the following equation introduced by Jike Ge et al. (Jike & Yuhui, 2008b).

$$\frac{1}{2} \left(\frac{\sum_{i \in path(C_b, C_p)} w_i dis(C_i, C_{i+1})}{\sum_{i \in path(C_b, C_{root})} w_i dis(C_i, C_{i+1})} \right) \quad (3)$$

Here, $depth(C)$ denotes the depth of concept C from the root in ontology hierarchy and k is a predefined factor larger than 1 (we use $k=2$ for our model) which specifies how weight values decrease towards leaves. So the distance between nodes at the upper level is higher than those between nodes at the lower level. The semantic distance above defines the distance as a “relative” distance to the distance between nodes and their common ancestors, thus it integrates the edge weight with the depth and the length of the shortest path.

After calculating the semantic distance between two concepts C_a and C_b , we can calculate the relevance between them using the following equation:

$$Relev(C_a, C_b) = 1 - dis(C_a, C_b) \quad (4)$$

The relevance score between two concepts is defined as the complementary of their semantic distance. Now with the user question and candidate query vectors $Q=(C_{u1}, C_{u2}, \dots, C_{um})$ and $C=(C_{v1}, C_{v2}, \dots, C_{vn})$ respectively, the relevance score between them can be measured using the following equation:

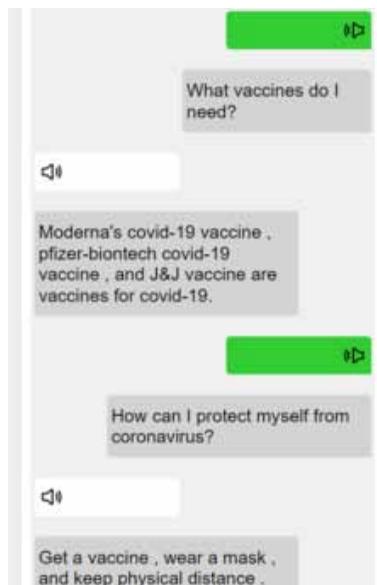
$$Relev(Q, C) = \frac{\sum_{j \in [1, m]}^n \max sim(Cu_i, Cv_j)}{n} \quad (5)$$

Once the candidate queries are ranked, we need to decide which of them should be returned as an answer. As some queries may not return a result, we may need to execute the queries before returning a result to the user. In the end, the system chooses non-empty results from the highest scored query. Then the results will be converted to the subject, verb, and objects format and send to SimpleNLG (Braun et al., 2019a), a Java API-based library, to generate a natural language sentence and return to the user.

Experimental Evaluation

We have implemented a prototype system, CovidBot, to study the performance and feasibility of the proposed system. Figure 6 shows the interaction between a user and CovidBot. The knowledge graph was presented with Neo4j running on a server in our private cloud through calling RESTful services. CovidBot client was implemented with Python and JavaScript with support of multiple tools and library including SpaCy (Goyal et al., 2018), spaczz (GitHub - Gandersen101/Spaczz: Fuzzy Matching and More Functionality for SpaCy., n.d.), PyTorch (Paszke et al., 2019), and anytree (GitHub - COfec0de/Anytree: Python Tree Data Library, n.d.). Our evaluation focuses on evaluating the chatbot's comprehension capability and question answering capability. We performed extensive experiments to test the performance of CovidBot. The testing data used in the experiments was from a manually created COVID-19 question set, COVID-Q (J. Wei et al., 2020), which is available for public use. The questions in the dataset were collected from multiple sources such as CDC, WHO, etc. The dataset comprises 1690 questions categorized in 15 different categories manually.

Figure 6. Communication between a user and the CovidBot



(Both the input and output are voice. Texts are converted from the voice)

In the first part of the experiments, we evaluated the performance of the intent identification. Intent classification is the initial step to identify a user's intent, therefore it can reduce a query's search space at the initial stage. We applied the Random Forest algorithm (Svetnik et al., 2003) to implement the intent classifier. We chose 500 trees(n) to stabilize the Out-of-Bag error (Mitchell, 2011). We use the Gini impurity (Strobl et al., 2007) as the criteria for node split. The number of features to consider when looking for the best split was set to "log2". The training and testing data were split to 80 to 20.

Table 1 shows the intent classification's performance in terms of accuracy, precision, recall, and F1-score, which are defined below in Equation (6)-(9) using the following abbreviations: True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (6)$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (7)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (8)$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (9)$$

As can be seen from Table 1, our intent classifier can achieve very good performance in identifying most of the user's intents (with macro and weighted average above 0.94). The only category with lower precision is the "Speculation" category, which is difficult to identify because of its fuzzy and subjective nature.

In the second part of the evaluation experiments, we evaluated the chatbot's question-answering performance inside a controlled setting. Specifically, out of the 15 categories related to Covid-19, we focused our implementation and testing on 5 categories, namely, Symptoms, Testing, Prevention, Risk, and Treatment. We tested 50 questions regarding precision, recall, and F1 score defined as follows:

$$\text{precision} = \frac{\text{number of relevant responses returned by system}}{\text{total number of responses returned by system}} \quad (10)$$

$$\text{recall} = \frac{\text{number of relevant responses returned by system}}{\text{total number of relevant responses}} \quad (11)$$

$$F1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{recision} + \text{recall}} \quad (12)$$

Table 1. Performance of Intent Classification

	Precision	Recall	F1-score	Support
Comparison	0.96	0.96	0.96	53
Economic Effects	1.00	0.98	0.99	55
Having COVID	1.00	0.90	0.95	60
Individual Response	1.00	0.89	0.94	62
Nomenclature	0.95	0.79	0.86	47
Origin	0.87	0.96	0.91	49
Prevention	0.98	0.95	0.96	55
Reporting	0.98	0.96	0.97	52
Societal Effects	0.98	0.96	0.97	67
Societal Response	0.98	0.95	0.97	59
Speculation	0.67	0.92	0.77	61
Symptoms	0.98	1.00	0.99	47
Testing	0.98	0.98	0.98	53
Transmission	0.98	1.00	0.99	47
Treatment	1.00	0.98	0.99	56
accuracy			0.95	823
macro avg	0.95	0.94	0.95	823
weighted avg	0.95	0.94	0.95	823

In definitions (10) and (11), a “relevant response” is a response returned by the chatbot, which is relevant to the user’s question, or (partially) answers the question. All responses from the chatbot to these 50 questions were manually checked and verified. The evaluation result is shown in Table 2. As can be seen from the table, the overall precision and recall rates are good: except for testing

Table 2. Performance of Question Answering

Query Category	Precision	Recall	F1
Symptoms	0.935	1.000	0.966
Testing	0.750	0.786	0.768
Prevention	0.955	1.000	0.977
Transmission	1.000	1.000	1.000
Treatment	0.896	1.000	0.945

queries that have a lower precision and recall rate (above 75%) most other queries get precision and recall above 90%.

After going through all the testing questions, we have identified several important factors that impact the query results: First, entity extraction from the user's questions has the most important impact on the query results. Mismatching may lead the search path in the wrong direction or stop the search too early. Moreover, identifying the starting point to search the knowledge graph is critical to the search result. The search result may differ if we choose a different starting point, even with the same search algorithm and the same entities we are looking for. Furthermore, the connectivity of the knowledge graph is another key factor affecting the query results. To some extent, the search effectiveness depends on the way how the entity nodes connect. The search may get trapped in a local area because of too few connections.

Conclusions

During this COVID-19 pandemic, people have been struggling with seeking accurate information about COVID-19. This has created an urgent need to employ effective mechanisms to disseminate information to citizens. For this aim, we propose a chatbot framework to assist people, especially older adults, to get up-to-date information on this disease thus assisting them to fight against it. The chatbot provides a natural interactive interface with users and it does not require any technical background which may be an obstacle for older adults to use new technologies. The system was built using technologies of natural language processing, semantic web, knowledge graph, and machine learning. It consists of several major components, including intent classification, semantic entity and relation extraction, knowledge graph mapping, query results selection, and response generation. Experiments have been performed on the prototype system, CovidBot. The initial qualitative evaluation demonstrates a reasonable recall and precision rate.

This study had some limitations. The qualitative evaluation is performed within a lab setting. It remains questionable whether real users in a natural setting outside the laboratory of the experiment would use CovidBot similar to researchers in the lab.

This project is ongoing. In the future, we plan to integrate data from user's social network sites and Electronic Health Records (EHR) in the knowledgebase. In this way, more dimensions of personalization can be performed. In addition, we plan to deploy the voice assistant in a real setting with older adults as the main users. More comprehensive user studies will be performed to evaluate the voice assistant's usability, the satisfaction rate of the users, and the quality-of-life improvement outcomes.

REFERENCES

- Ait-Mlouk, A., & Jiang, L. (2020). KBot: A Knowledge Graph Based ChatBot for Natural Language Understanding over Linked Data. *IEEE Access : Practical Innovations, Open Solutions*, 8. Advance online publication. doi:10.1109/ACCESS.2020.3016142
- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3). Advance online publication. doi:10.1080/00031305.1992.10475879
- Answers By Cigna. (n.d.). <https://www.amazon.com/Cigna-Answers-by>
- Bao, Q., Ni, L., & Liu, J. (2020). HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention. *ACM International Conference Proceeding Series*. <https://doi.org/doi:10.1145/3373017.3373049>
- Braun, D., Klimt, K., Schneider, D., & Matthes, F. (2019a). SimpleNLG-DE: Adapting SimpleNLG 4 to German. *INLG 2019 - 12th International Conference on Natural Language Generation, Proceedings of the Conference*. <https://doi.org/doi:10.18653/v1/w19-8651>
- Braun, D., Klimt, K., Schneider, D., & Matthes, F. (2019b). SimpleNLG-DE: Adapting SimpleNLG 4 to German. *Proceedings of the 12th International Conference on Natural Language Generation*, 415–420. <https://doi.org/doi:10.18653/v1/W19-8651>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1). Advance online publication. doi:10.1023/A:1010933404324
- CDC COVID-19 assessment chatbot. (n.d.). <https://covid19healthbot.cdc.gov/>
- Cerf, V. (1973). *RFC0439: PARRY encounters the DOCTOR*. RFC Editor.
- Chakraborty, C., Banerjee, A., Garg, L., & Rodrigues, J. J. (2020). *Internet of Medical Things for Smart Healthcare. Studies in Big Data*. Springer.
- Chakraborty, C., & Rodrigues, J. J. C. P. (2020). A comprehensive review on device-to-device communication paradigm: Trends, challenges and applications. *Wireless Personal Communications*, 114(1), 185–207.
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 785–794.
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*. <https://doi.org/doi:10.3115/v1/d14-1179>
- Dash, S., Chakraborty, C., Giri, S. K., & Pani, S. K. (2021). Intelligent computing on time-series data analysis and prediction of COVID-19 pandemics. *Pattern Recognition Letters*, 151, 69–75.
- Dutta, B., & DeBellis, M. (2020). CODO: An Ontology for Collection and Analysis of Covid-19 Data. *ArXiv Preprint ArXiv:2009.01210*.
- Express Scripts. (n.d.). <https://www.amazon.com/Express-Scripts/dp/B07QB7P6Y2>
- Fix, E., & Hodges, J. L. (1989). Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties. *International Statistical Review / Revue Internationale de Statistique*, 57(3). <https://doi.org/doi:10.2307/1403797>
- Francis, N., Green, A., Guagliardo, P., Libkin, L., Lindaaker, T., Marsault, V., Plantikow, S., Rydberg, M., Selmer, P., & Taylor, A. (2018). Cypher: An evolving query language for property graphs. *Proceedings of the ACM SIGMOD International Conference on Management of Data*. <https://doi.org/doi:10.1145/3183713.3190657>
- Gerber, D., & Ngomo, A. C. N. (2012). *Extracting multilingual natural-language patterns for RDF predicates*. Lecture Notes in Computer Science. Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics., doi:10.1007/978-3-642-33876-2_10
- GitHub - c0fec0de/anytree: Python tree data library. (n.d.).

GitHub - gandersen101/spacz: Fuzzy matching and more functionality for spaCy. (n.d.).

Goyal, P., Pandey, S., & Jain, K. (2018). SpaCy. In *Deep Learning for Natural Language Processing*. Creating Neural Networks with Python.

Jike, G., & Yuhui, Q. (2008a). Concept similarity matching based on semantic distance. *Proceedings of the 4th International Conference on Semantics, Knowledge, and Grid, SKG 2008*. <https://doi.org/doi:10.1109/SKG.2008.24>

Jike, G., & Yuhui, Q. (2008b). Concept similarity matching based on semantic distance. *Proceedings of the 4th International Conference on Semantics, Knowledge, and Grid, SKG 2008*. <https://doi.org/doi:10.1109/SKG.2008.24>

Jung, S. (2019). Semantic vector learning for natural language understanding. *Computer Speech & Language*, 56. Advance online publication. doi:10.1016/j.csl.2018.12.008

Kamiński, B., Jakubczyk, M., & Szufel, P. (2018). A framework for sensitivity analysis of decision trees. *Central European Journal of Operations Research*, 26(1). Advance online publication. doi:10.1007/s10100-017-0479-6

Kiroku. (n.d.). <https://trykiroku.com/>

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*. <https://doi.org/doi:10.18653/v1/n16-1030>

Li, F., Liao, L., Zhang, L., Zhu, X., Zhang, B., & Wang, Z. (2020). An efficient approach for measuring semantic similarity combining wordnet and wikipedia. *IEEE Access : Practical Innovations, Open Solutions*. Advance online publication. doi:10.1109/ACCESS.2020.3025611

Livi - UCHealth. (n.d.). <https://www.uchealth.org/today/tag/livi/>

Livongo Blood Sugar Lookup. (n.d.). <https://www.amazon.com/Livongo-Health-Blood-Sugar-Lookup/dp/B07QHF76RN>

Merit.ai. (n.d.). <https://www.merit.ai/>

Mesnil, G., He, X., Deng, L., & Bengio, Y. (2013). Investigation of recurrent-neural-network architectures and learning methods for spoken language understanding. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. <https://doi.org/doi:10.21437/interspeech.2013-596>

Mitchell, M. W. (2011). Bias of the Random Forest Out-of-Bag (OOB) Error for Certain Input Parameters. *Open Journal of Statistics*. Advance online publication. doi:10.4236/ojs.2011.13024

My Children's Enhanced Recovery After Surgery (ERAS). (n.d.).

Neo4j. (2012). *neo4j: World's Leading Graph Database*. Neo4J.Org.

Norman, C. D., & Skinner, H. A. (2006). eHealth literacy: Essential skills for consumer health in a networked world. In *Journal of Medical Internet Research*. <https://doi.org/doi:10.2196/jmir.8.2.e9>

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., & Fang, L. ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.

Ramos, J. (2003). Using TF-IDF to Determine Word Relevance in Document Queries. *Proceedings of the First Instructional Conference on Machine Learning*.

Ranoliya, B. R., Raghuvanshi, N., & Singh, S. (2017). Chatbot for university related FAQs. *2017 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2017, 2017-January*. <https://doi.org/doi:10.1109/ICACCI.2017.8126057>

Rish, I. (2001). An empirical study of the naive Bayes classifier. *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 22230. <https://doi.org/doi:10.1039/b104835j>

- Saygin, A. P., Cicekli, I., & Akman, V. (2000). Turing test: 50 years later. *Minds and Machines*, 10(4). Advance online publication. doi:10.1023/A:1011288000451
- Setiaji, B., & Wibowo, F. W. (2016). Chatbot Using a Knowledge in Database: Human-to-Machine Conversation Modeling. *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS, 0*. <https://doi.org/doi:10.1109/ISMS.2016.53>
- Setyawan, M. Y. H., Awangga, R. M., & Efendi, S. R. (2018). Comparison Of Multinomial Naive Bayes Algorithm And Logistic Regression For Intent Classification In Chatbot. *Proceedings of the 2018 International Conference on Applied Engineering, ICAE 2018*. <https://doi.org/doi:10.1109/INCAE.2018.8579372>
- Shawar, B. A., & Atwell, E. (2015). ALICE chatbot: Trials and outputs. *Computación y Sistemas*, 19(4). Advance online publication. doi:10.13053/CyS-19-4-2326
- Strobl, C., & Boulesteix, A. Data, T. A.-C. S. &, & 2007, undefined. (n.d.). Unbiased split selection for classification trees based on the Gini index. *Elsevier*. Retrieved October 24, 2021, from <https://www.sciencedirect.com/science/article>
- Strobl, C., Boulesteix, A. L., & Augustin, T. (2007). Unbiased split selection for classification trees based on the Gini Index. *Computational Statistics & Data Analysis*. Advance online publication. doi:10.1016/j.csda.2006.12.030
- Svetnik, V., Liaw, A., Tong, C., Christopher Culberson, J., Sheridan, R. P., & Feuston, B. P. (2003). Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *Journal of Chemical Information and Computer Sciences*. Advance online publication. doi:10.1021/ci034160g
- Swedish Health Connect*. (n.d.). https://www.amazon.com/DIT-Providence-St-Joseph-Health/dp/B07PGJYYF6/ref=sr_1_1?keywords=Swedish+Health+Connect&qid=1583186851&s=digital-skills&sr=1-1
- Thomas, N. T. (2016). An e-business chatbot using AIML and LSA. *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*. <https://doi.org/doi:10.1109/ICACCI.2016.7732476>
- VoiceFriend Integrated Messaging and Engagement solution*. (n.d.).
- VoiceFriend*. (n.d.). <https://www.voicefriend.net/>
- Walgreens Voicebot.ai*. (n.d.). <https://voicebot.ai/tag/walgreens/>
- Wei, J., Huang, C., Vosoughi, S., & Wei, J. (2020). What are people asking about COVID-19? A question classification dataset. In arXiv.
- Wei, J. W., Huang, C., Vosoughi, S., & Wei, J. (2020). What Are People Asking About COVID-19? A Question Classification Dataset. *CoRR, abs/2005.12522*. <https://arxiv.org/abs/2005.12522>
- Weizenbaum, J. (1966). ELIZA-A computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9(1). Advance online publication. doi:10.1145/365153.365168
- Wray, M., Csurka, G., Larlus, Di., & Damen, Di. (2019). Fine-grained action retrieval through multiple parts-of-speech embeddings. *Proceedings of the IEEE International Conference on Computer Vision*. <https://doi.org/doi:10.1109/ICCV.2019.00054>
- Xie, B., Charness, N., Fingerma, K., Kaye, J., Kim, M. T., & Khurshid, A. (2020). When Going Digital Becomes a Necessity: Ensuring Older Adults' Needs for Information, Services, and Social Inclusion During COVID-19. *Journal of Aging & Social Policy*. Advance online publication. doi:10.1080/08959420.2020.1771237
- Zaman, N., & Li, J. (2014). Semantics-enhanced recommendation system for social healthcare. *Proceedings - International Conference on Advanced Information Networking and Applications, AINA*. <https://doi.org/doi:10.1109/AINA.2014.93>

Xin Wang is a Ph.D. student in the Department of Computer Science of North Dakota State University. Her research interests focus on Database, Ontology, Natural Language Processing, and Machine Learning. Tianyi Liang received the B.S. degree in computer science from North Dakota State University (NDSU) in 2018, where he is currently pursuing the Ph.D. degree. He is currently a research assistant at NDSU. His research interests include human-computer interaction, user experience, artificial intelligence, and software engineering.

Vikram Pandey received his B.E. degree in Electronics and Communication Engineering from Savitribai Phule Pune University, Pune, India, in 2015. He is currently pursuing his Ph.D. degree in Computer Science at the North Dakota State University, Fargo, ND, USA. His research interests include Semantic Web, Ontology, Blockchain, and IoT.

Yang Du is an assistant professor at Valley City State University. His current research interest focuses on Artificial Intelligence, Machine Learning, and Bioinformatics.