Using Graph-Based Learning to Create Conversational Intelligence Platforms

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Sadhika Dhanasekar

Spring, 2024

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Rosanne Vrugtman, Department of Computer Science

Using Graph-Based Learning to Create Conversational Intelligence Platforms

CS4991 Capstone Report, 2024

Sadhika Dhanasekar
Computer Science
The University of Virginia

School of Engineering and Applied Science

Charlottesville, Virginia USA

qkx9an@virginia.edu

ABSTRACT

Noblis, a non-profit corporation delivering technical and advisory solutions to federal government clients, required a Conversational Intelligence Platform, or ChatBot, for the purpose of connecting employees in a company to other sources of help internally. I utilized graph-based relational database learning to augment the underlying machine learning basis. The relational database deploys RDFLib Triples to connect employees via their skillsets and experiences in the company by noun-verb-adjective triplets. The data is collected via employee surveys and the company's online system, InTheKnow, detailing previous projects and news articles regarding the company. The data is analyzed using Python and the SpaCy library in conjunction with an underlying machine learning base powered by Microsoft AzureBot Framework. Preliminary results indicate success in finding employees who have worked on similar projects to supply aid and advice. However, the ChatBot fails to consider many supplementary skillsets that may also be useful to search with and may mistake names with occupations, projects, etc. Future work may expand on the relational database to encompass more traits, such as how much free time the employee has or their team role in the project.

1. INTRODUCTION

Noblis is an independent, nonprofit organization serving clients in the civil and defense fields to ensure our nation's security. However, the broad range of clientèle and projects Noblis covers makes rapid communication difficult, hindering employees from leveraging existing frameworks or projects to enhance their own assignments. While Noblis provides their employees with a routinely updated database of previous projects and news articles regarding the company called InTheKnow, the lack of an advanced search system and overflow of information can make it difficult to find the right resource or garner advice from an unknown coworker.

To facilitate communication between employees and promote interteam collaboration, I was tasked with creating a Conversational Intelligence Platform, specifically called a Chatbot. A Chatbot is a computer program designed to simulate conversation with human users that is trained with natural language processing (NLP) techniques. This Chatbot would be tailored to connect employees through the use of graph-based relational database learning, augmenting the underlying machine learning basis that handles the NLP processing.

There are many use cases of the designed Chatbot. One application is to connect employees to someone more experienced in their domain, such as a mentor or peer who has developed programs in a previously unknown software. Another utilization involves employees exploring projects in their field that they can pull inspiration from when helping a client, delivering work more efficiently. In this way, employees can be more informed about their own skillsets and available to help their coworkers.

2. RELATED WORKS

The usage of Chatbots to assist employees is not new, but as technology and other acpabilities have been growing, they are gaining a significant importance as a virtual assistant. Vilela (2021) explores the use of a chatbot model to improve remote employee experiences by managing growing company logistics. He concludes that his Chatbot prototype was positively recieved by mobile employees who enjoyed the connection it provided to their coworkers. It also helped to keep managers and their subordinates more on track. My approach takes advantage of the base NLP processing model Vilela used for the Chatbot, Diaflow, but will expand on the model by using a dynamic database relational system that does not require manual updating.

Graph-based learning is also frequently used to supplement machine learning models. Farhadi et al. (2012) looked at deploying a graph that correlates employee skills to that of a previously determined expert group, and found that this grouping prevented redundanct communication costs. My approach borrows this graph-based grouping, but instead of grouping by a subjective score of skill, the approach standardizes their skills by looking at project complexity and experience to document connections between employees.

Lastly, the benefits of collaboration and partnership in the workplace is important to note. Brigid et al. (2002) discovers that having a mix of expert areas professionals creates a major shift in action on behalf of the team, and allows them to be more directive in accomplishing the goals and services they provide. The ultimate goal is to improve the collaboration among Noblis employees, and my approach utilizes techniques, such as suggestions for peer reviews.

3. PROJECT DESIGN

The goal of this project is to design a functional chatbot that can respond to user queries with an underlying, graph-based relational database of employee information.

3.1 Review of System Architecture

The system architecture consists of three main components. The first component is the database backend containing a relational, graph-based designed software that connects employee information to each other. This is done using Python and the RDFLib Triples library. The next component is the NLP and querying software that links the backend with the frontend, i.e. the questions asked by the user of the ChatBot. This is trained using the SpaCy NLP library and dataset. Lastly, the third component deals with the frontend ChatBot machine that acts like an app within the Microsoft System Offices. This component also formats plausible responses for the question the user has asked using machine learning principles to continuously format better and more accurate answers. The app is displayed using the Microsoft AzureBot Framework and a built-in machine learning model trained on example queries.

3.2 Requirements

3.2.1 Client Needs

The system is designed to work for Noblis and their internal employees and facilitate communication between teams. One requirement of the system is to document and encode employee projects inside Noblis. To gain reliable information about project members, data must be correctly pulled from InTheKnow using popular and tagged keywords and store useful information, such as specific project elements worked on, client deployed for and team member roles. In addition, the system should be constantly updating the information it recieves as new projects are added into the database. The system can also gain information from the questions that are asked by connecting the user with others who have asked about Python and add those connections to its database.

Another requirement of the system is to interact with a wide array of queries. Users should be able to ask for someone who can assist them with a topic, similar projects or previous projects of a clientele; and find employees curious to learn about an area, knowledge domain experts, and other areas related to completing their task. It is important that the ChatBot is able to respond to these queries, and also build upon its previous knowledge to answer them correctly. If a previous answer was not satisfactory, the ChatBot should not give the same answer and should adapt its algorithm and understanding of the equation accordingly.

3.2.2 System Limitations

The system is limited by the data that it is given as it is limited to the interaction with InTheKnow and human errors in employee surveys. There is potential that the system will miss out on appropriate keywords. As project entries in InTheKnow are written by employees, there can be technical terms or discrepencies between similar projects that the system will not be able to pick up on. In addition, the system could be obstructed by bad user input that it may use to build its backend, or loose connections that may be reinforced by an incorrect understanding of employee information. Thus, the main limitation of the system is the lack of understanding project descriptions and connecting that to the nuances in user queries.

3.3 Key Components

3.3.1 Specifications

The first specification of the system is in designing a backend with easily retrievable and well-connected entries. Using Python, the data from InTheKnow was sorted into three major categories: people, project keywords, and verbs. Verbs would account for positions and roles, such as finding coders, writers, analysts, managers, etc. based on the action they were conducting. Project keywords were found based on the most common, unique words that were evident in project submission entries, as well as a manually inputted database comprised of common querying words and employee surveys. This data was encoded into RDFLib Triples, which connects people to verbs to keywords, forming the graph-based aspect of the database. Each triple can then be connected via any of their three categories to other triples, incorporating the relational aspect of the database.

The next specification of the system lies in the NLP linking layer. This layer was pretrained using the SpaCy library in Python, which consists of a large language model that is able to identify most common parts of speech. This was tailored to interact with the backend database by sorting parts of speech by proper nouns, nouns and verbs, which aligned with the aspects of the RDFLib Triples. From there, a hashing dictionary was used to see if these words were in the database and to send back the search results based on the highest number of connections that word had.

The last two specifications were the machine learning model and app framework. The machine learning model was trained to determine the context of the query. This was done using many example queries that fulfiled the broad range of services required. Examples include “Can you connect me with someone experienced in using SQL in databases?” and “Who has done projects for this company?”. The model is able to process the connections returned back from the NLP layer and return the correct response, formatted in an understandable way that highlights why this was the final recommendation made. A sample response to the first example question would be, “Jane Doe is recommended to be reached out to based on her experience in Project X and her skillset in SQL”. The model was also able to send feedback through the NLP layer to the backend based on if the user liked the response they were given, or wanted to gain a new response.

For the app framework, the Microsoft AzureBot Framework contains an interactive app layout that was adjusted to meet the ChatBot’s needs. The app took up a bottom quarter of a screen and allowed for the user to type a question, and recieve a response with the person or other information they requested.

3.3.2 Challenges

One challenge I ran into was in finding the best graph-based theory to organize the data in, before eventually settling on RDFLib triples. Many suggestions tried resulted in poorly connected data, and a lack of good querying abilities as they chose poor words to hash and search on. The graph would also grow too quickly, and take too long to properly search.

Another challenge was connecting the machine learning model to the NLP layer results. The context of the question could be unclear due to confusions in overlapping names and words. For example, phrases like “Is Chris Baker experienced in Python?” would be confused with baker as a profession rather than as a proper noun and name.

3.3.3 Solutions

RDFLib triples was chosen based on the related works that were looked at before aligning with that of Farhedi et al. (2012), but taking out their connection with a direct object, and replacing it with the project's tagged keywords. This resulted in an effective connection approach and was easy to sort out from the InTheKnow articles.
The challenge regarding the model's inability to deal with confusing language restrictions or understanding is one that is still being worked on. As the model encounters more unique queries, it will be able to train itself to differentiate between words and tasks given and become more successful.

4. RESULTS

The ChatBot is able to return a response with almost every query. It is also able to recognize queries beyond its reach, such as an user asking what the weather is today, and state that this is not a permissable question. In the times that the query returns a useful answer, coworkers who have tried it have expressed positive remarks about the recommendations it has made and the ease of finding an expert in the queried area or previous projects they can build on, saving them days of time on projects that they were stuck on. It also helps them recognize the convenience of the InTheKnow website for finding past, helpful experiences.
However, the queries do not always return a satisfactory result. Sometimes it may return a person who is not experienced in the area, but had their name written in a connected paper. Othertimes, it misunderstands a question and returns an answer on an unrealted topic. During these situations, the user is able to request other answers, which may return a more satisfactory answer after a few tries. This also serves to train and improve the model for future iterations.

5. CONCLUSION

To facilitate a more efficient working encironment and lead to more workplace collaboration, a Chatbot was created. The Chatbot was made using Azure Bot Framework, NLP techinques, and a relational database of employee data. The graph-based approach proved to be effective and accurately connected employees with minimized space, augmenting the underlying machine learning basis that handles the NLP processing. While trained on relatively low data, the Chatbot surpassed expectations and was able to deliver promising results in recommending employee information, previous projects, and other data scraped from the company's internal news site.

However, there are limitations to consider about the current edition of the Chatbot. One would be the use of privacy in the Chatbot as data of employees who leave the company or work with more classified information will be put into the Chatbot without their knowledge. There is also a limitation in the data collection, as employees can choose not to fill out a survey or fail to report their projects correctly. Lastly, the broadness of the company results in different vocab terms or missions that the Chatbot would fail to match on between articles even if the projects were similar enough, leading to missed opportunities for the user.

FUTURE WORK

Future work may expand on the relational databse of the Chatbot to encompass more qualities such as the free time an employee has, their team role in a project, their realm of experiences, and other helpful attributes. This would be crucial to creating a comprehensive Chatbot assistance as it would be able to determine more accurate connections and provide more data to assist the user in their queries. In addition, the underlying queries could be further tested and explored as the Chatbot should remain robust to unique phrasings and unconventional queries.

This Chatbot could grow to encompass a greater role, such as generating high-level code, project plans, clientele data, and other work helpful to the workplace. Another helpful application would be the ability to deploy the Chatbot on multiple environments, especially on a mobile device. This opens the possibility of using the Chatbot during meetings to take notes, look up mentioned people or projects, or as a general assist in the workplace.

ACKNOWLEDGMENTS

I'd like to acknowledge Jennifer Miller and Caleb Aryee for their roles as great mentors and team members, and helped connect me to many resources and facilitate my transition into the company. I'd also like to thank my manager, Cody Rutherford, for pitching the idea and his weekly check-ins. Lastly, I'd like to thank my peer mentor, Randy Bailey, for guiding the idea from a humanitarian perspective.

REFERENCES

Farhadi, F., Sorkhi, M., Hashemi, S., & Hamzeh, A. (2012, May 19). An effective framework for fast expert mining in collaboration networks: A group-oriented and cost-based method - journal of computer science and technology. SpringerLink. https://link.springer.com/article/10.1007/s11390-012-1245-9

Flannery, K. B., Slovic, R., Treasure, T., Ackley, D., & Lucas, F. (2002, January 1). Collaboration and partnership to improve employment outcomes. Journal of Vocational Rehabilitation. https://content.iospress.com/articles/journal-of-vocational-rehabilitation/jvr00160

Valela, V. (2021, April 16). Run: Global mobility chatbot: Chatbot model to improve mobile ... NOVA Information Management Sector. https://run.unl.pt/handle/10362/116183