BI-Bot: QA System in Healthcare Domain

A Project Semester Report

Submitted in Partial Fulfilment of the

Requirements for Award of the Degree of

Computer Science Engineering

*by*

APAR GARG

Roll No: E17CSE112

*under the supervision of*

Mr. Swetank Gupta

A close up of a sign

Description automatically generated

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

SCHOOL OF ENGINEERING AND APPLIED SCIENCES

BENNETT UNIVERSITY

GREATER NOIDA, UTTAR PRADESH, INDIA

MAY 2021

© APAR GARG, (2021)

Bennett University has the royalty-free permission to reproduce and distribute copies of this Report for teaching and research as well as for dissemination of Knowledge.

# DECLARATION

I hereby declare that except where specific reference is made to the work of others, the contents of this thesis are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This report is my work and does not contain any outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Apar Garg

Roll No: E17CSE112

Bennett University,

Greater Noida

09/05/2021

# ABSTRACT

Nowadays, QA systems are used as automated assistant systems for resolving queries in various messaging applications. Accessing a database to get specific information can be stressful and time-consuming if one does not have the much-needed technical skills. Designing a QA system/chatbot that can generate the result by accessing the database dynamically has linguistic and design challenges. In this work, I present a medical domain-specific QA system, which can provide real-time responses to any employee's standard business queries in a healthcare company by removing unwanted dependencies on the analytics team. The QA system uses NLP and DL techniques for NER and text pre-processing, etc.

*Keywords: QA System, NER, Data Engine*

# LIST OF FIGURES

|  |  |
| --- | --- |
| **Figure No.** | **Description** |
| Figure 1 | Architecture of proposed QA system. |
| Figure 2 | Black Box of proposed SpaCy model. |
| Figure 3 | Output for Business Type questions. |
| Figure 4 | Output for Technical Type questions. |

# LIST OF TABLES

|  |  |
| --- | --- |
| **Table No.** | **Description** |
| Table 1 | System Specifications. |
| Table 2 | Sample Business Type queries with corresponding functionality types. |
| Table 3 | Sample Technical Type queries with sample responses. |
| Table 4 | Synonyms and Acronyms used in the dataset for the QA system. |

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **Abbreviation** | **Explanation of the Abbreviation** |
| QA | Question Answering |
| NLP | Natural Language Processing |
| DL | Deep Learning |
| NER | Named Entity Recognition |

# CONTENTS

[DECLARATION iii](#_Toc71454340)

[ABSTRACT iv](#_Toc71454341)

[LIST OF FIGURES v](#_Toc71454342)

[LIST OF TABLES vi](#_Toc71454343)

[LIST OF ABBREVIATIONS vii](#_Toc71454344)

[CONTENTS viii](#_Toc71454345)

[1. ABOUT THE COMPANY 9](#_Toc71454346)

[2. SYSTEM SPECIFICATIONS 10](#_Toc71454347)

[3. INTRODUCTION 11](#_Toc71454348)

[4. RELATED WORK 12](#_Toc71454349)

[5. METHODOLOGY 13](#_Toc71454350)

[5.1. User Query 13](#_Toc71454351)

[5.2. Query pre-processing 14](#_Toc71454352)

[5.3. NER 15](#_Toc71454353)

[5.4. Spell Check 15](#_Toc71454354)

[5.5. Retrieve Data from Data Engine 15](#_Toc71454355)

[6. RESULTS 16](#_Toc71454356)

[7. CONCLUSION AND FUTURE WORK 18](#_Toc71454357)

[8. LEARNING OUTCOME 19](#_Toc71454358)

[8.1. Healthcare Domain Learnings 19](#_Toc71454359)

[8.2. Technical Learnings 19](#_Toc71454360)

[8.3. Other Learnings 19](#_Toc71454361)

[9. CHALLENGES 20](#_Toc71454362)

[9.1. Technical Challenges 20](#_Toc71454363)

[9.2. Other Challenges 20](#_Toc71454364)

[REFERENCES 21](#_Toc71454365)

# ABOUT THE COMPANY



D Cube Analytics is a US-based company with a branch located in Bangalore, India. D Cube helps the brand, sales, marketing, and market access teams at life sciences companies manage their brands effectively across stakeholders and markets as well as population health managers at payer & provider organizations drive optimal clinical and financial outcomes using insights.

They aim to deliver precise and predictive insights across the value chain that empowers customers to make confident and timely data-driven decisions at an asset and/or functional level.

* Industry - Pharmaceuticals
* Company size - 51-200 employees
* Headquarters - Schaumburg, Illinois, USA
* Type - Privately Held
* Founded - 2014
* Specialties -

1. Healthcare Analytics
2. Payer-Provider Analytics
3. Pharma Commercial Analytics

* Scope -

1. Helps brands make confident and timely data-driven decisions.
2. Deliver precise and predictive insights using Data Science.

# SYSTEM SPECIFICATIONS

**Table 1: System Specifications.**

|  |  |
| --- | --- |
| **Item** | **Value** |
| Processor | IntelI Core I i5-8250U CPU @ 1.60GHz,  1800 Mhz, 4 Core(s), 8 Logical Processor(s) |
| Total Physical Memory | 7.89 GB |
| Total Virtual Memory | 31.9 GB |
| System type | x64-based PC |
| OS Name | Microsoft Windows 10 Home Single  Language |
| OS Version | 10.0.18363 Build 18363 |

* Programming Language – Python
* Platform – Jupyter Notebook
* Python interface for Slack API – Slacker
* Python Frameworks -

1. Data Manipulation - numpy, pandas
2. Data Visualization - seaborn, matplotlib
3. NLP - re, nltk, spaCy, gensim

# INTRODUCTION

A new study suggests that technology is turning individuals more and more impatient each day. In this age of information technology, Google has become the de-facto place to get instant answers to generic queries. However, these days a typical user wants a solution to his specific question. Users might have to query a vast database to get the specific information they need. A typical user is generally not comfortable using a database to retrieve the information he/she needs because it either requires the knowledge of complex programming languages or an inter-mediator, which can use the databases and tell the result to the user. However, using an intermediary or learning a programming language is very tiresome and taxing. People having a technical background could also find this task a bit difficult sometimes.

For solving these kinds of problems, QA systems [1] were introduced. They are systems that take questions from the user in natural language as their input and send back answers.

This work introduces a medical domain-specific knowledge-based QA system, which can answer the queries related to the sales and marketing of drugs from a given database. The QA system provides real-time responses to the common business queries of any employee in a healthcare company by removing unwanted dependencies on the analytics team. This could bring significant time savings to both the analytics team and the employees having queries. I address factual questions expressed by WH pronouns. An answer can just be a table or a combination of table and bar plot(s). I focus on searching and extracting answers from a dummy database provided by D Cube. However, the suggested solution can easily be extended to other use cases as well.

The main contributions of this work are:

1. Design a Healthcare domain-specific QA system: A QA system is designed to understand healthcare queries. It extracts entities and establishes a correspondence between multiple instances of entities. Hence, I have proposed an end-to-end system to understand user queries and generate responses using a data engine.
2. Integrate QA system with Slack: Slacker [2], a Slack API client, was used to connect Python Backend with Slack. This provides the users with fast and easy access to the QA system.
3. Collection of domain-specific data: No sample query data is available for healthcare QA system research. I have collected data for different types of sample questions, each with unique functionality for training and testing.
4. Custom training for NER: I have trained a custom blank SpaCy [3] model using manual annotation for NER [4].

# RELATED WORK

Recent developments in DL and NLP have introduced domain-specific QA systems and chatbots as digital assisting technologies to answer user queries dynamically. In the following subsections, I summarize work done in the past related to medical domain-specific QA systems and chatbots.

Weizenbaum et. al. [5] designed the first natural language database system called ELIZA for simulating a psychotherapist. It worked on identifying patterns and simple structural and syntactic structures in the database from the user inputs. In another research, Green et. al. [6] proposed BASEBALL, which was another domain-specific question answering system for statistical queries regarding baseball games, played during the American league.

Woods et. al. [7] proposed LUNAR, a QA system that gave information on soil samples taken from lunar exploration by Apollo. The system turned user questions into database queries by means of simple pattern matching rules and produce answers at last. In a similar research, Androutsopoulos et. al. [8] proposed a NLIDB framework that facilitated the users to ask questions in their natural languages and later obtained data from databases.

Clarke et. al. [9] looked to the web as a question answering asset. The framework provided by them performed complex parsing and extraction of entities for both queries and best-matching web pages.

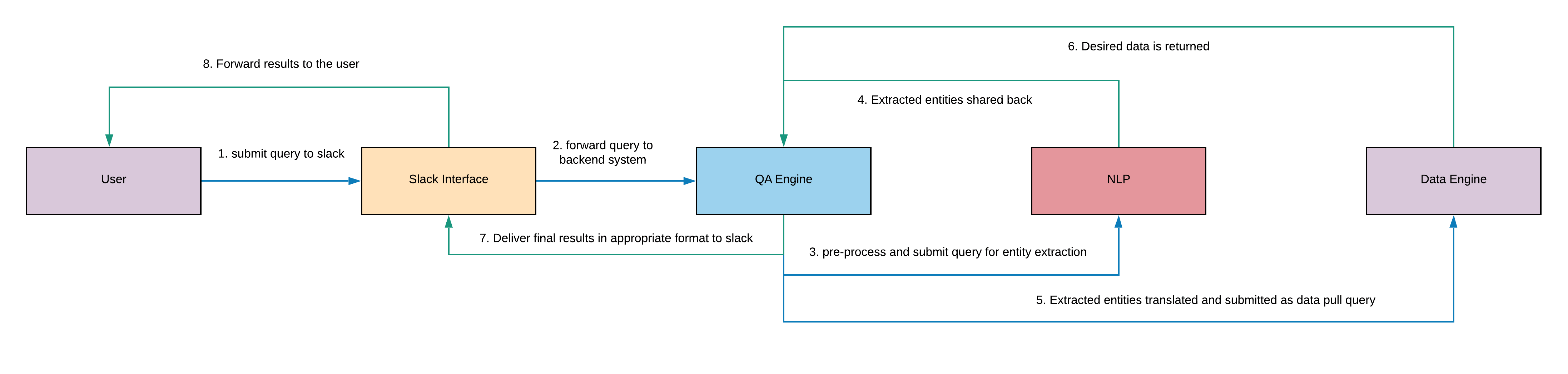
Brill et. al. [10] proposed the AskMSR QA system design and evaluated the contributions to accuracy from the various system components. They additionally discussed predictive approaches, where the answering question system would possibly throw an erroneous response. In a similar research, Zheng et. al. [11] developed AnswerBus, which is an open-domain QA framework focused on retrieval of information at the sentence level. It addressed natural-language questions from users in various languages such as English and German and extract potential Web answers.

Abacha et. al. [12] introduced MEANS, which combined NLP and semantic web technologies. The paper used Medical entity recognition, which could detect and eliminate the phrasal information referring to medical entities and classify the entities in predefined categories. The categories were problem, test, drug, etc.

Zhu et. al. [13] gave a hierarchical attention retrieval model for question answering in the healthcare domain. The proposed model used two bi-directional RNN encoders to encode the inter-documents dependencies for a given query and document words. It also used cross attention between the query and the document. The system could answer binary answers, who, etc. But it was mainly designed for “what” and “how” type of questions in the healthcare domain.

# METHODOLOGY

Fig. 1. presents the basic architecture of my QA system. Firstly, the user submits a query to the QA Engine via Slack Interface. The submitted query is pre-processed and is sent to the NLP Engine. The NLP Engine extracts keywords or Named Entities, from the query. Then the extracted entities are forwarded to the Data Engine for retrieval of results. The retrieved results are sent back to the Slack Interface and displayed to the user.



**Figure 1: Architecture of proposed QA system.**

## **5.1. User Query**

A user can submit multiple queries to the QA system via Slack, one at a time. Slacker library was used to connect Python to Slack. Table 2 below shows some of the possible business questions which a user can ask, along with the type of functionality they are testing. Similarly, Table 3 shows some of the technical questions and their predefined responses.

**Table 2: Sample Business Type queries with corresponding functionality types.**

|  |  |
| --- | --- |
| **Type of functionality tested** | **Sample Question** |
| Singe value sales question | What is the total sales of Prolia in the USA in quarter 1 of 2018? |
| Multiple value sales question | What is the total sales of Alendronate in Canada and France in Q2 2019? |
| Different KPIs | What is the QTD sales of Prolia and Alendronate? |
| Comparison by time period | What is the sales of Xgeva in Q4 2016 vs last year same time period? |
| Comparison – Actual vs Budget | How are our sales for the USA this year and how does it look against Budget? |
| Other singe value questions | Top-performing product in ICON in the current quarter by sales? |

**Table 3: Sample Technical Type queries with sample responses.**

|  |  |
| --- | --- |
| **Technical question** | **Sample Response** |
| How is forecast calculated? | Forecast is calculated based on the previous sales and current market scenario by the Forecasting Team. |
| Till what date is data available? | Data is available till Dec 2019; Data is generally refreshed in the first week of every month and comes with one month lag. |
| How is sales vs Budget calculated? | Sales vs Budget for selected time period = (Sales for the time period)/ (Budget for the time period) – 1. |
| When was the data last refreshed? | Latest refreshed date of data: Jan 2020. |

## **5.2. Query pre-processing**

Pre-processing of the user query is an important phase in the system. I have done pre-processing in three steps, as mentioned below:

1. Noise Removal

Punctuations and additional white spaces do not give significant information to the query. They are just noise in text. Therefore, I have removed all punctuations (except comma, a hyphen, and question-mark) and additional white spaces in the query.

1. Lowercasing

Lowercasing was done to simplify the task of normalization of the query. Hence, there is no need to map both 'United States' and 'united states' to 'US'. After lowercasing the query, ‘United States’ will also get converted to 'united states'. So, mapping ‘united states’ to ‘US’ is sufficient. This task has saved a lot of space in my table of synonyms and acronyms.

1. Normalization

I have compiled a table of synonyms and acronyms from various sources on the web. The table consists of 200 synonyms and acronyms, which are mapped to their actual representation in the Data Engine. Table 4 given below shows some examples. This is helpful when e.g., United Kingdom, England, Britain maps to only one country in the data engine i.e., UK.

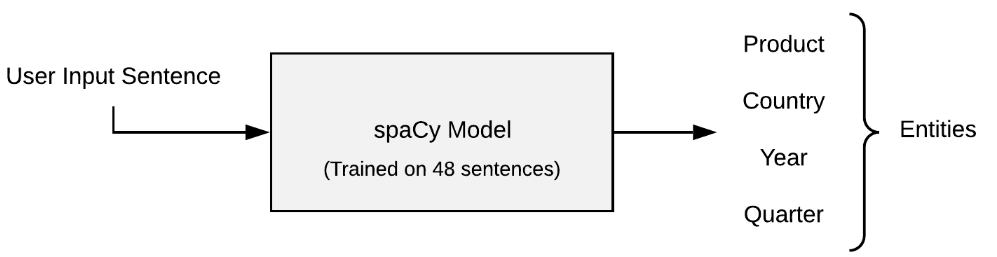
**Table 4: Synonyms and Acronyms used in the dataset for the QA system.**

|  |  |
| --- | --- |
| **Representation in Query** | **Representation in Data Engine** |
| united kingdom | UK |
| year to date | YTD |
| united states | US |
| quarter to date | QTD |
| england | UK |

## **5.3. NER**

I have trained a blank spaCy model using annotated training data to recognize four custom entities from the user query: product, country, quarter, and year. For annotating data, a custom-built web browser-based tool called spaCy NER Annotator was used. These custom entity tags are also the attributes in the Data Engine.

The model was trained on 92 sentences. After training it for 20 epochs with dropout 0.2 and SGD optimizer, the model gave a training loss of 2.06-8. Fig. 2. presents the black box of the model.



**Figure 2: Black Box of proposed SpaCy model.**

## **5.4. Spell Check**

Edit distance [14] is a method for finding how dissimilar two strings are. It can also be viewed as the minimum number of editing required to transform a word into another word. For example, the edit distance between the words "ran" and "run" is 1. Three operations are allowed on a character: insert a character, delete a character, and replace a character. This problem is solved using dynamic programming.

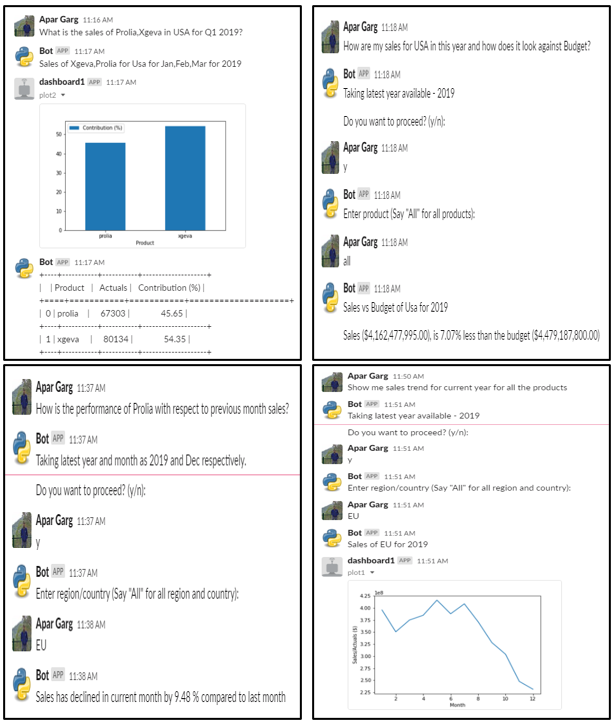
I have used the edit distance technique, to rectify spelling errors in extracted entities. The entry in the Data Engine column (with corresponding entity tag) having the minimum edit distance with the entity, is considered to be the right spelling of the entity. The minimum edit distance of an entity must be less than 2, for it to get transformed.

## **5.5. Retrieve Data from Data Engine**

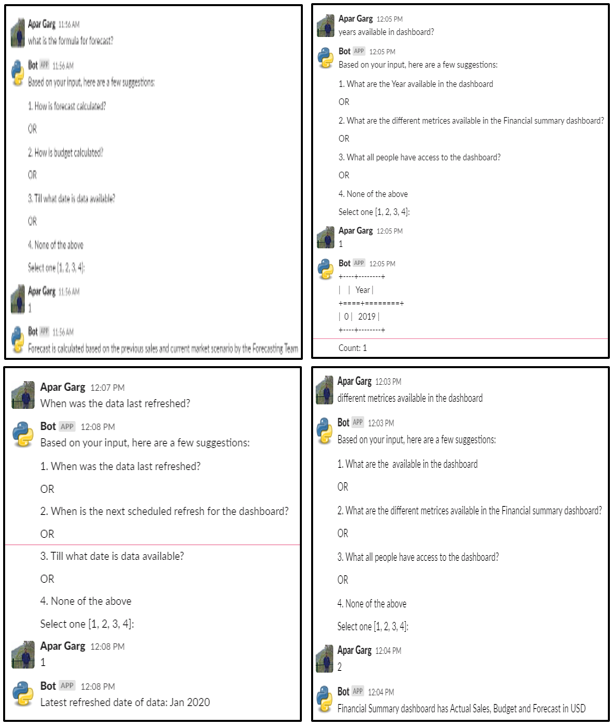
If the translated entities are not present in the Data Engine, then the QA system response is "Apologies... I don't understand your question”. If there is any lack of information within the query, the QA system will request more inputs from the user. After this, the translated entities are mapped to entries in the database to fetch “Total Sales” (represented by Actuals in Data Engine), Budget, or Forecast corresponding to those entries.

The retrieved results for sales could be a single row or it could be multiple rows. The QA system will accompany the result with a bar plot in case of multiple row output.

# RESULTS

Fig. 3. and Fig. 4. shows the output to some of the Business Type and Technical Type questions, respectively.

**Figure 3: Output for Business Type questions.**

**Figure 4: Output for Technical Type questions.**

# CONCLUSION AND FUTURE WORK

The goal of this work was to build a QA system in the medical domain which can provide real-time responses to the common business queries of any employee in a healthcare company by removing unwanted dependencies on the analytics teams. The QA system makes use of advanced NLP techniques like NER and text pre-processing, etc. to understand and answer complex user queries.

My future goal is to work on Boolean questions expecting a yes/no answer and other types of factual questions. Another future goal is to develop augmented intelligence that identifies the type of question and gives more insights around that question.

# LEARNING OUTCOME

## **8.1. Healthcare Domain Learnings**

Communicate effectively with respect to the following topics:

* Difference between Healthcare and Pharmaceutical sector.
* US Healthcare system.
* Major stakeholders in the healthcare system and their characteristics
* Drug Discovery and Development cycle with the timeline.
* Pharma Process Flow and stakeholders.
* Different types of datasets in the pharma domain.
* Pharmacology and subdivisions of pharmacology.
* Differences between branded and generic drugs.
* Common Therapeutic areas.

## **8.2. Technical Learnings**

1. Perform Pre-processing and EDA on different datasets in the pharma domain.
2. Select important features and engineer new features from existing datasets.
3. Apply Machine Learning theory and Computer Science fundamentals to produce computing-based solutions.
4. Custom NER for given data.
5. Text data manipulation and processing techniques.
6. Integration of chatbot with a messaging service like Slack.

## **8.3. Other Learnings**

1. Function effectively as a member of an organization.
2. Write Minutes of Meeting (MoM).
3. Write Status Update mail at the end of the day.
4. Develop a timeline for a project.

# CHALLENGES

## **9.1. Technical Challenges**

* Augmented Insights - Developing augmented intelligence that identifies the type of question and giving more insights around that question.
* Execution Time - Developing efficient code. Preprocessing the question and extracting entities from it is quite computationally expensive.

## **9.2. Other Challenges**

* KT sessions - Intensive KT sessions on fundamentals of pharma and healthcare were difficult to grasp initially due to limited prior knowledge in the field.
* NLP Course - Had to take up a course in Chatbot Systems. Managing the course with day-to-day internship tasks was pretty exhaustive.

# REFERENCES

1. Terol, Rafael M., Patricio Martínez-Barco, and Manuel Palomar. "A knowledge based method for the medical question answering problem." Computers in biology and medicine 37, no. 10 (2007): 1511-1521.
2. <https://pypi.org/project/slacker/>
3. <https://spacy.io/usage/training\#ner>
4. Mansouri, Alireza, Lilly Suriani Affendey, and Ali Mamat. "Named entity recognition approaches." International Journal of Computer Science and Network Security 8, no. 2 (2008): 339-344.
5. Weizenbaum, Joseph. "ELIZA—a computer program for the study of natural language communication between man and machine." Communications of the ACM 9, no. 1 (1966): 36-45.
6. Green Jr, Bert F., Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. "Baseball: an automatic question-answerer." In Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference, pp. 219-224. 1961.
7. Woods, William A. "Progress in natural language understanding: an application to lunar geology." In Proceedings of the June 4-8, 1973, national computer conference and exposition, pp. 441-450. 1973.
8. Androutsopoulos, Ion, Graeme D. Ritchie, and Peter Thanisch. "Natural language interfaces to databases–an introduction." Natural language engineering 1, no. 1 (1995): 29-81.
9. Clarke, Charles LA, Gordon V. Cormack, and Thomas R. Lynam. "Exploiting redundancy in question answering." In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 358-365. 2001.
10. Brill, Eric, Susan Dumais, and Michele Banko. "An analysis of the AskMSR question-answering system." In Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp. 257-264. Association for Computational Linguistics, 2002.
11. Zheng, Zhiping. "AnswerBus question answering system." In Human Language Technology Conference (HLT 2002), vol. 27. 2002.
12. Abacha, Asma Ben, and Pierre Zweigenbaum. "MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies." Information processing \& management 51, no. 5 (2015): 570-594. University of Pennsylvania School of Engineering and Applied Science Department of Computer and Information Science (2017).
13. Zhu, Ming, Aman Ahuja, Wei Wei, and Chandan K. Reddy. "A hierarchical attention retrieval model for healthcare question answering." In The World Wide Web Conference, pp. 2472-2482. 2019.
14. Ristad, Eric Sven, and Peter N. Yianilos. "Learning string-edit distance." IEEE Transactions on Pattern Analysis and Machine Intelligence 20, no. 5 (1998): 522-532.