

https://dspace.org/

Automatic Comprehension of Customer Queries for Feedback Generation

Okwunma, Nnamdi Ekene

2018

https://hdl.handle.net/10539/26482

Okwunma, Nnamdi Ekenedilichukwu,. (2019). Automatic comprehension of customer queries for feedback generation, University of the Witwatersrand, https://hdl.handle.net/10539/26482

Downloaded from a DSpace repository. https://dspace.org/

Automatic Comprehension of Customer Queries for Feedback Generation

Nnamdi Ekene Okwunma
(Student Number: 818576)



School of Computer Science and Applied Mathematics,
University of the Witwatersrand, Johannesburg.

A dissertation submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg in fulfillment of the requirements for the degree of Master of Science.

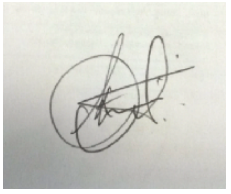
Supervised by
Dr. Abejide Ade-Ibijola

June 2018

Declaration

I, Okwunma Nnamdi .E, hereby declare the content of this dissertation to be my own work unless otherwise explicitly referenced. This dissertation is submitted for the degree of Master of Science at the University of the Witwatersrand, Johannesburg. This work has not been submitted to any other university, nor for any other degree.

Signed:



Date: 25 - 06 - 2018

Abstract

One major challenge in customer-driven industries is the response to large volumes of queries. In response to this business need, Frequently Asked Questions (FAQs) have been used for over four decades to provide customers with a repository of questions and associated answers. However, FAQs require some efforts on the part of the customers to search, especially when the FAQ repository is large and poorly indexed or structured. This even gets difficult when an organisation has hundreds of queries in its repository of FAQs. One way of dealing with this rigorous task is to allow customers to ask their questions in a Natural Language, extract the meaning of the input text and automatically provide feedback from a pool of FAQs. This is an Information Retrieval (IR) problem, in Natural Language Processing (NLP). This research work, presents the first application of Jumping Finite Automata (JFA) — an abstract computing machine — in performing this IR task. This methodology involves the abstraction of all FAQs to a JFA and applying algorithms to map customer queries to the underlying JFA of all possible queries. A data set of FAQs from a university's Computer and Network Service (CNS) was used as test case. A prototype chat-bot application was developed that takes customer queries in a chat, automatically maps them to a FAQ, and presents the corresponding answer to the user. This research is expected to be the first of such applications of JFA in comprehending customer queries.

Keywords — Jumping Finite Automata (JFA), Natural Language Processing (NLP), Frequently Asked Questions (FAQs), Matching.

Acknowledgements

I give all glory to my Creator, the Lord, God almighty for the gift of life; and for making today a reality.

A special gratitude to my amazing supervisor, Dr. Abejide Ade-Ibijola for his effortless inputs, counselings and guidance all through the course of this project. I could not have done it without your selfless academic contributions sir, you are indeed God sent.

To my amiable wife Barr. (Mrs) Chioma Ekene-Okwunma, God bless you immensely for your patience, understanding, prayers and support (mentally, emotionally, spiritually) I love you.

To my Parents Sir G.O.C and Lady Chinyere Okwunma for bringing me into this world and ensuring I never lacked anything till this stage, thank you for your prayers and advises.

To my mother and father in-law Sir Chinedu and Lady Carol Oburuoga, I am proud to have you in my life, thank you for your prayers and support.

To Prof. Clifford Odimegwu and family for their endless support in all ramifications, I am really humbled sir.

To my spiritual father, Pastor Prince Nwadike for his endless and sleepless nights of intervening prayers for me, thank you sir.

To my siblings, Onyinye Chamberlain Udeze, Amarachukwu Onyenuforo, Uchechukwu Oburuoga, Amarachi Oburuoga, Onyekachi Njoku, Akudo Igwemezie Ugonna Oburuoga, Chimaobi Oburuoga and Chidinma Okwunma for all your supports and prayers.

Not forgetting Fernando for his efforts and also my senior colleagues, brothers and academic Dr's in the making: Micheal Ayewe, Kehinde Aruleba, Blessing Okpokiri, George Obaido, Jude Ewemade and Mostafa(Mosty) for their constant support through out the course of this work.

A brother and friend Eze Sixtus Amefuna and family, your constant prayers kept me going.

Eyichukwu Odimegwu and Nelly Mvandaba, God bless you for your prayers and support.

Chimezie Anthony Okafor and family, your supports will forever be appreciated.
God bless you all.

Contents

Declaration i

Abstract ii

Acknowledgements iii

Contents iv

List of Figures vi

List of Tables vii

Publications viii

1 Introduction 1
- 1.1 Background . . . 1
- 1.2 Problem Statement . . . 3
- 1.3 Motivation . . . 3
- 1.4 Research Context . . . 4
  - 1.4.1 Aim and Objectives . . . 4
  - 1.4.2 Research Questions . . . 5
- 1.5 Methodology . . . 5
- 1.6 Research Contribution . . . 6
- 1.7 Document Outline . . . 7

2 Background and Related Work 8
- 2.1 Introduction . . . 8
- 2.2 Question Answering Systems . . . 8
- 2.3 Techniques and Components . . . 13
- 2.4 JFA as an Abstract Machine . . . 22
  - 2.4.1 Why JFA? . . . 23
  - 2.4.2 Comparative Analysis of Classical FA and JFA . . . 25
- 2.5 Algorithms used in Natural Language Question Answering Systems . . . 26
  - 2.5.1 Cocke–Younger–Kasami Algorithm (CYK) . . . 27
  - 2.5.2 Earley's Algorithm . . . 27
  - 2.5.3 Shift-Reduce Parsing Algorithm . . . 28

2.6 Conclusion 29

3 Design, Implementation and Results **30**
3.1 Introduction 30
3.2 JFA Design 30
3.2.1 Algorithms 32
3.3 Implementation 33
3.3.1 Abstracting FAQs to JFA 33
3.3.2 Generating Feedback from FAQs using JFA 44
3.4 Text Normalization 45
3.5 Results 47
3.6 Conclusion 49

4 Evaluation **50**
4.1 Introduction 50
4.2 Data Set 50
4.3 Ground Truth Data 52
4.4 Evaluation Metrics 52
4.5 System Performance Metrics 53
4.6 Conclusion 57

5 Conclusion and Future Work **58**
5.1 Conclusion 58
5.2 Limitations of the Study 58
5.3 Contributions and Future Work 59

A Ground Truth Work **66**

B Evaluation Results **73**

List of Figures

2.1 Yahoo! Answers. . . . 9
2.2 Uber FAQs page . . . 14
2.3 Broad classification of NLP: Source: Khurana et al. [2017]. . . . 17
2.4 Phases of NLP architecture: Source: Khurana et al. [2017]. . . . 18
2.6 JFA Computation . . . 26

3.1 JFA Design . . . 31
3.2 Transition process . . . 35
3.3 Top Mail 1st Query . . . 38
3.4 Top Mail 1st and 2nd Queries . . . 40
3.5 Complete Top Mail Queries . . . 43
3.6 Match Process . . . 44
3.7 Real time queries mapped to JFAs . . . 48

B.1 Test 1 . . . 73
B.2 Test 2 . . . 73
B.3 Test 3 . . . 74
B.4 Test 4 . . . 74
B.5 Test 5 . . . 74
B.6 Test 6 . . . 75
B.7 Test 7 . . . 75
B.8 Test 4 . . . 76

List of Tables

3.1 JFA symbols 1 - 24 36
3.2 JFA symbols 25 - 52 37

4.1 Archiving and Deleting Messages with rephrased queries 50
4.2 Top Mail FAQs with rephrased queries 51
4.3 Spam FAQs with rephrased queries 51
4.4 Mobile App FAQs with rephrased queries 51
4.5 Labels FAQs with rephrased queries 51
4.6 File Attachment FAQs with rephrased queries 51
4.7 Email Features FAQs with rephrased queries 52
4.8 Conversations and Messages FAQs with rephrased queries 52
4.9 Original Queries 55
4.10 Synthetically Generated Queries 55

Publications

Portions of this research was presented at the following conference;

- Nnamdi Ekene-Okwunma, Abejide Ade-Ibijola, and Turgay Celik (2017). Automatic Comprehension of Customer Queries in Online Instant Messaging Systems for Feedback Generation (Abstract and Poster). In the proceedings of the 2nd International Conference on the Internet, Cyber-security, and Information Systems (ICICIS), pages 168–169, Sandton, Johannesburg, South Africa.

Chapter 1

Introduction

1.1 Background

Question Answering (QA), is one of the most natural forms of human-computer interaction, with the task of automatically generating answers to natural language questions from humans [Sosnin, 2012; Yao, 2014]. Furthermore, QA is a form of Information Retrieval (IR), characterised by information needs that is partially expressed as natural language statements or questions [Kolomiyets and Moens, 2011].

QA systems, make use of Frequently Asked Questions (FAQs) files as its knowledge base [Burke et al., 1997]. These FAQs are usually files developed from information obtained from different sources, such as structured interviews with individuals or small groups, existing sources like publications, archival records, or previous questionnaires and tests [Weller, 2007]. FAQs are usually found on the world wide web, or corporate websites. FAQs pages on the web provide resources that address user information needs [Jijkoun and de Rijke, 2005].

The ultimate goal of FAQs repository is to address questions and answers posed by customers [Schwalb, 2004]. Traditional QA systems relied on question and answer templates, which were mostly manually constructed [Yao, 2014]. Customers were expected to go on the FAQs page of a website and manually search for their desired questions and its related answer, resulting to more search-time than supposed. This is an Information Retrieval (IR) problem.

The need for effective methods of automated IR has grown in importance because of the tremendous explosion in the amount of unstructured data, both internal corporate document collections, and the immense growing number of document sources on the internet [Greengrass, 2000].

Successes have been reported in a series of question-answering evaluations that started in 1999 as part of the Text Retrieval Conference (TREC) [Voorhees and Tice, 1999], to ease the search for information from pool of documents. Some of these projects were by Magnini et al. [2003]; van Delden and Gomez [2004]; Fan et al. [2012]; Wang et al. [2016].

This work is an Information Retrieval (IR) problem in Natural Language Processing (NLP) domain. NLP has been around for more than four decades [Ade-Ibijola, 2016]. The domain, investigates techniques for tasks such as automatic natural language comprehension, summarisation, machine translation, optical character recognition, information retrieval and parsing [Jacobs, 2014]. Machines may be considered as intelligent to some reasonable extent, if machines can perform these tasks on representations of natural languages (textual or verbal). IR systems rank documents by their estimation of the usefulness of a document for a user query, and most IR systems assign a numeric score to every document and rank documents by this score [Singhal, 2001].

In other to solve this IR problem, we proposed to use Finite Automata (FA) to model the work. Considering that FA reads input strings in a linear logic manner, we decided to use a non-linear logic model of FA known as Jumping Finite Automata (JFA), considering we are dealing with semi-structured data.

JFA, is a recent model of FA, that reads input discontinuously [Meduna and Zemek, 2014a]. These automata work just like classical finite automata except that they do not read their input strings in a symbol-by-symbol left-to-right way (linear logic). It enables a jump at any position of the input while processing input strings. In a nutshell, after reading a symbol, they can jump over some symbols within the words and continue their computation from there, but with the aim of reaching the accepting state (non-linear logic). JFA can start reading strings from any where, be it at the start state or the final state. JFA is discussed broadly in Section 2.4.

We tested the system on real user queries. According to Walker et al. [2014], to evaluate a QA system for commercial use, it would be preferable to test it on real user queries. That is, questions that have been posed by potential or real end-users rather than system developers or testers. The most important aspect for a functional QA system is to answer those put by real users. The results of our tests are shown and discussed in Section 3.5 and Appendix B.

1.2 Problem Statement

Access to relevant information is one of the major problems faced by users in the information circus nowadays [Bekhti et al., 2011]. Mostly, a user lacks time to find a short and precise answer to his/her query among the range of available documents. Hence, precision in retrieving the accurate information is crucial and a challenging task for Information Retrieval systems developers.

This work, is concerned with solving IR problems in NLP using JFA as a model to improve two main modules of FAQs, *question processing and answer processing*. The evaluation of this system's operations affects both modules.

1.3 Motivation

People have questions and they need answers, not documents. Automatic question answering will definitely be a significant advance in the state-of-art information retrieval technology [Srihari and Li, 2000]. Most classical computer science methods in the previous century, were developed for continuous information processing [Meduna and Zemek, 2014b]. Their formal models such as finite automata, work on words and represents information in a strictly continuous left-to-right symbol-by-symbol way accordingly. Modern information methods, however are often designed for a different way of information processing. In our present days, computational methods frequently process information in a discontinuous way [Büttcher et al., 2016; Christopher et al., 2008]. Within a particular running process, a typical computational step may be performed somewhere in the middle of information, while the very next computational step is executed far away from it; In other words, before the next step is carried out, the process has to jump over a large portion of the information to the desired position of execution [Meduna and Zemek, 2012].

The World Wide Web is not just an information system, it is a social space where people interact. One major form of interaction consists of asking and answering questions. It has become the first stop for finding answers to questions.

Answers provided at question asking sites are a form of public good, as they are made available freely to the general public for unlimited consumption [Raban and Harper, 2008]. A certain contributor critical mass and contribution quality level are vital for the production and persistence of public goods.

To find information, customers search and browse [Rice et al., 2001]. Search engines are used to locate web pages conforming to the set of search terms specified by customers.

Sets of information database or web links are surfed by customers to understand the types of information available and to find topically-relevant content.

In order to make information retrieval easy for customers, I proposed to build an IR based system, that will automatically generate feedback to their queries, using JFA. The principal motivation of this project, is to build a functioning knowledge-based information retrieval system, which will rely on the knowledge engineering inbuilt of FAQs files, distributed on the internet in natural language using JFA to enable automated feedback to queries. i.e. building the next generation of question answering system.

1.4 Research Context

1.4.1 Aim and Objectives

This work is aimed to apply JFA in IR natural language question answering system, that will be an information service available to customers. Where a user can pose a query in natural language, and if it happens to be related to a QA pair in our training FAQs file, the system will be able to automatically generate the matching answer to the query.

This work also aims to advance customer relationship with industries, using natural language customer-support system that understands and interacts with users in a satisfying manner, especially when dealing with FAQs. Customer queries feedback process has been improved by this research work.

One of the key uses of feedback is to provide robustness of uncertainty [Aström and Murray, 2010]. The term $Feedback$ refers to the process in which two or more dynamical systems (a system whose behaviour changes overtime, often in response to external stimulation or forcing) are connected together, such that each system influences the other and their dynamics are thus strongly coupled [Aström and Murray, 2010].

In order for the aim of this research to be fulfilled, corpus of FAQs from Wits CNS FAQ page on the web were extracted to use as our training data. Furthermore manually categorized them into nodes for effective recognition by the automata. JFA as discussed in the background and related work of this research, is to be implemented as a model. This is expected to be the first of such application of JFA in NLP.

At the end of this research work, customer's search-time on question answer platforms regarding FAQs will reduce. In other words customers can be able to pose their

queries in natural language and receive automated feedback with related answers to their queries.

1.4.2 Research Questions

Specifically, the following questions have been answered during the course of undertaking this research.

1. *How do we abstract FAQs to Jumping Finite Automata?* This question is addressed in Subsection 3.3.1 of this work.
2. *How do we generate automatic feedback for customers using JFA?* When a question is posed by a customer, the JFAs' as seen in Table (3.2) are mapped to the set of FAQ files, and a list of files ranked by relevance to the question is returned.

1.5 Methodology

In order for the objectives of this research to be accomplished, all possible FAQs are abstracted to a JFA, and algorithms applied to map customer queries to the underlying JFA of all possible queries. In order to test how well these queries are recognized, a prototype chat-bot application was developed. This chat-bot takes customer queries in a chat, automatically maps them to a FAQ, and presents the corresponding answer to the user. The methodology of this research was conducted in two phases:

1. Already existing FAQs were abstracted to JFA, and the states were shown. Details of what the transition process is and how it functions were also shown, and lastly explained how the data in this work is represented.
2. Automatic feedback for customers were generated using JFA model. In that course, we showed how related answers were matched with the questions in the database.

Although the tools discussed in the literature of this work, have been effective in designing question answering systems, it is understood that non has been done using JFA. The methodology of this work is conducted and completed by giving thorough insight of the following:

1. *the states of this work's JFA design*,
2. *the transition process of the JFA design*,

3. *how we represented data in this work*, and
4. *how the answers are matched with the questions in the database*.

The states of this work's JFA design. A state is a member of a set. Furthermore JFA states are the elementary building blocks or units of the automaton, they are historical abstractions. Usually, an automata is considered as a modeling framework for discrete event system. As these systems are dynamic in nature, their variables evolve with time and thus a configuration of the values of these variables is a state of the system. In the context of this research, our semantics-oriented explanation of the above entails a somewhat less radical abstraction, because our system might still allow for more than one attribute per state. States could as well be vectors in such a manner that any two of them could still have some components in common if only at least one component is different — e.g.: S = {a,b,c} whereas S'= {a,b,d} — thus allowing for some elements of "historic continuity" across an "epochal break". More details of this phase are provided in Chapter 3.

The transition process of the JFA design. A transition function is a move from one automata state to another. JFA, processes its input strings by jumping either from the beginning or at the end of the symbols. It can also jump alternately from the left or right, so the input is accepted from both ends of the word. More explanation of this phase is also provided in Chapter 3.

How data is represented in this work. Data representation can be done in a lot of ways. But in this research work, we represented data with $\sum$. More explanation is given in Chapter 3

How the answers are matched with the questions in the database. A proper explanation is given in Subsection 3.3.2

1.6 Research Contribution

This study describes our effort, to ensure the ease of customers stress towards online FAQs document retrieval, also search-time reduction, in finding related answers to desired questions;

1. Jumping Finite Automata (JFA) was used; a formal model of computation to recognised user queries and map them to FAQs,
2. A tool called CNSBot was developed; an intelligent chat bot that chats with a user and responds to their queries,

3. The CNSBot was tested with the 50 training FAQs and the performance metrics is shown in Section 4.5

1.7 Document Outline

The rest of this research is organized as follows:

- Chapter 2 *Background and related work*: This chapter presents an overview of introduction to the concept of automatic comprehension of customer queries for feedback generation. Furthermore, similar projects that covered questions answering are discussed in detail. Including the NASA project whose methods are compared to the algorithms and design of this work. JFA as a model is properly introduced and discussed
- Chapter 3 *Design and implementation*: This chapter presents the methods used in this work to build an efficient question answering system.
- Chapter 4 *Evaluation*: This chapter provides the evaluation metrics used and also details on the results of the experiment carried out in this research.
- Chapter 5 *Conclusion, Limitations, Contribution and Future Work*: This chapter concludes the research, provides the limitations of the study, presents future work, and outlines contribution of the research study.

Chapter 2

Background and Related Work

2.1 Introduction

This chapter discusses work related to the background of question answering systems and presents the approach for carrying out this research. In particular, Section 2.2 introduces the conceptual idea of question answering systems; its origin and the different modifications made by researchers. Section 2.3 Highlights some techniques and components seen in this work. Section 2.5 discusses few algorithms used in question answering systems. Section 2.6 concludes the chapter.

2.2 Question Answering Systems

Several surveys on question answering technologies have been made in the past. The very first approach to question answering in English was reviewed by Simmons [1965]. Background, motivation and general approaches to open domain question answering highly promoted by the Text Retrieval Conference (TREC) were discussed by Hirschman and Gaizauskas [2001].

Question Answering series evaluations started in 1999 as part of the Text Retrieval Conference (TREC) [Voorhees and Tice, 1999]. The TREC-8 Q/A track was the first large-scale evaluation of domain-independent Q/A system. Its goal was to retrieve small snippets of text that contain the actual answer to a question rather than the document lists traditionally returned by text retrieval system.

An update on the approaches used in open domain question answering was also presented in [Mollá-Aliod and Vicedo, 2010] and [Webber and Webb, 2010] accordingly.

In the recent times, the question asking process has been formalized into knowledge markets/question asking sites. Example is Yahoo! Answers as seen in Figure 2.1. These sites vary in membership, goals and technologies, but all provide interfaces for members to broadcast questions to the community, as well as methods for finding and answering other members' questions.

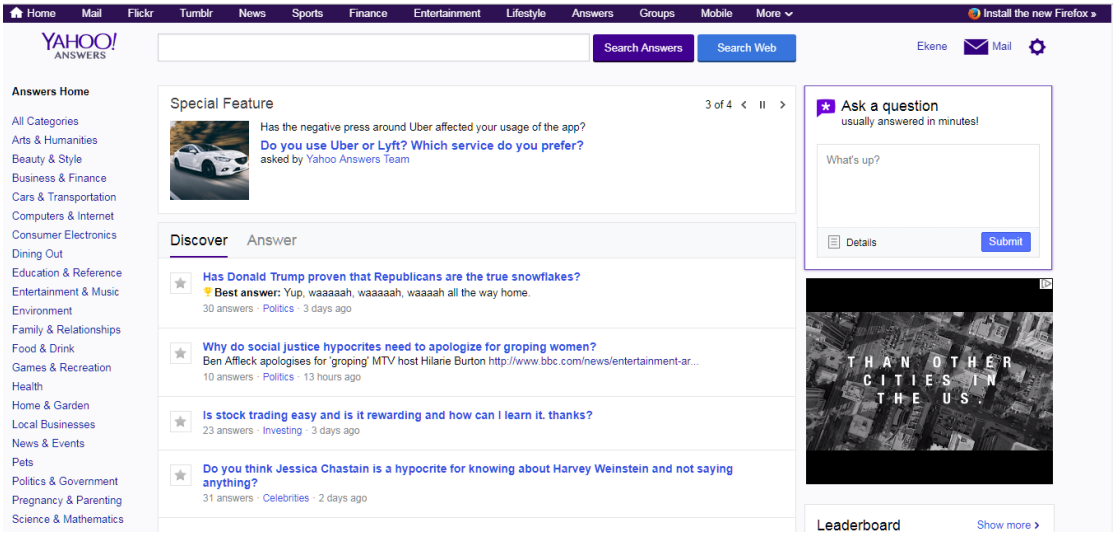


FIGURE 2.1: Yahoo! Answer.

The survey presented in this work is different from the previous ones in several ways. Firstly, distinction between open domain and restricted domain question answering are not made, and it is not restricted to the approaches reported in the Q/A field only. It presents the Q/A task from an information retrieval perspective and emphasizes the importance of the retrieval models. i.e., representation of queries and information documents, and the retrieval functions which are used for estimating the relevance between a query and its answer.

Other related projects also exist in NLP domain for question answering purposes, but using different approaches as presented below.

BASEBALL; An Automatic Question-Answerer [Green Jr et al., 1961]:

> BASEBALL is one of the earliest question answering systems that supports a finite amount of questions on corpora containing a fixed set of documents. It is a computer program that answers questions phrased in ordinary English about stored data. The program reads the question from punched cards. After the words and idioms are looked up in a dictionary, the phrase structure and other syntactic facts are determined for a content analysis, which lists attribute-value pairs specifying the information given and the information requested. The requested information is then extracted from the data matching the specifications, and any necessary

processing is done. The program's focus is on baseball games. The program is organized into several successive and essentially independent routines, each operating on the output of its predecessor and producing an input for the routine that follows. The linguistic routines include question read-in, dictionary look-up, syntactic analysis, and content analysis. The processing routines include the processor and the responder. The specification list indicates to the processor what part of the stored data is relevant for answering the input question. The processor extracts the matching information from the data and produces for the responder the answer to the question in the form of a list structure. The core of the processor is a search routine that attempts to find a match on each path of a given data structure, for all the attribute-value pairs on the spec list; when a match for the whole spec list is found on a given path, these pairs relevant to the spec list are entered on a found list. Finally, the answer is printed.

Retrieving NASA problem reports [van Delden and Gomez, 2004]: A case study in natural language information retrieval. This is a Finite Automata (FA) system that retrieves problem reports (PR) from the National Aeronautics Space Administration (NASA) PR database (it searches the database and returns relevant PR to questions which are asked in natural language by NASA engineers). Its database is queried with natural language questions (hence a natural language information retrieval problem). The system uses rule-based part-of-speech tagger to first assign to each word in the question, also a partial-parse of the question is produced with independent sets of deterministic finite state automata (by using partial-parse information, a look up strategy searches the database for problem reports relevant to the question). It relies on enhancing noun and verb phrase matching for its search strategies (A bi-gram stemmer and irregular verb conjugates techniques is incorporated into the system to improve matching accuracy). And finally, the system is evaluated by a set of 55 questions posed by NASA engineers.

AQUASYS; A Question-Answering System for Arabic [Bekhti et al., 2011]: AQUASYS is designed to answer fact-based, questions seeking answers related to different types of entities: person, location, organization, time, quantity, etc. The system is composed of three modules: [1] A question analysis module, [2] A sentence filtering module and [3] An answer extraction module. The question analysis phase is crucial and plays an essential role in the answer finding phase. This system gives more attention to the question analysis in order to extract, from it, valuable and informative features unlike most of the existing Arabic Q-A. These features are decisive for the answer filtering process and consequently have a strong impact on answer finding accuracy performance. The answers relatedness scoring

phase serves to find the most accurate answer. The global AQUASYS architecture is built on four main modules, namely: Question analysis, sentences filtering, candidate answers finding and candidate answers scoring and ranking modules. Each module is developed based on a number of sub-modules and/or processes. At the end of the test an overall recall rate of 97.5% (the recall rate is computed as the number of relevant answers retrieved by the number of relevant answers in the documents) were obtained and 66.25% as a precision rate (the number of question answered correctly by the total number of question asked).

Natural Language Question Answering over RDF — A Graph Data Driven Approach [Zou et al., 2014]: In this project they proposed a semantic query graph to model the query intention in the natural language question in a structural way, based on which, RDF Q/A is reduced to sub graph matching problem. More importantly, they resolve the ambiguity of natural language questions at the time when matches of queries are found. They believed that the cost of disambiguation is saved if there are no matching found. RDF question/answering (Q/A) allows users to ask questions in natural languages over a knowledge base represented by RDF. To answer a natural language question, the existing work takes a two stage approach: question understanding and query evaluation. Their focus is on question understanding to deal with the disambiguation of the natural language phrases. The technique used, is the most common technique called the joint disambiguation, which has the exponential search space. They compared their method with some state-of-the art RDF Q/A systems in the benchmark data set. Extensive experiments confirm that their method not only improves the precision but also speeds up query performance greatly.

Question Answering System Using Natural Language Processing With NLIDB Approach [Malhotra, 2017]: This project focuses on creating semantic analyzer for automatic Question-answering system for domain specific database. It provides user with the relevant answers to the user questions using Natural Language processing (NLP) and Natural language interface for database (NLIDB). NLIDB are the systems that translate a natural language sentence into a database query. It contains the stepwise description on conversion of question to simple SQL query without using any clauses. According to these researchers, it portrays completely automatic, reliable, fast way to query a database. They applied Natural language processing techniques on English text and converted to SQL query using series of steps like lowercase conversion, tokenization, chunking, generation of SQL query and mapping the query to the database. It uses semantic matching technique to translate the Natural Langauge question to the relative SQL query. Various steps such as lower case conversion, tokenization and ambiguity remover are used to

convert to SQL query which is mapped to the database to get the required information.

The Multi-source Integrated Platform for Answering Clinical Questions (MiPACQ) [Cairns et al., 2011]: This is a QA pipeline, that integrates a variety of information retrieval and natural language processing systems, into an extensible question answering system. MiPACQ is an integrated framework for semantic-based question processing and information extraction. Using NLP and information retrieval (IR) techniques, MiPACQ accepts free-text clinical questions and presents the user with sufficient answers, from a variety of sources such as general medical encyclopedic resources, and the patients data residing in the Electronic Medical Record (EMR). This system integrates numerous NLP components to enable deeper semantic understanding of medical questions and resources, and it is designed to allow integration with a wide range of information sources and NLP systems. A baseline information retrieval system that operates at the document level is created based on the $Lucene$ full-text search index, which by default uses the vector space model with normalized $tf - idf$ parameters to rank the documents being queried. MiPACQ document-level baseline makes extensive use of stemming and stop-listing to improve both the precision and recall of the results.

The Interactive Multimodal Information Extraction (IMIX) Program, a research program by Hofs et al. [2011]: that took place between 2004 and 2009, and was funded by the Netherlands Organization for Scientific Research (NOW). IMIX had the design of an interactive multimodal question answering(QA) system, that is able to answer general encyclopedic medical questions in natural language as one of its area of concentration. Its demonstrator (Dennis Hofs, Boris van Schooten, and Rieks op den Akker) threw open a fully functional system to users and allowed them to ask questions using text, speech and gestures and received their feedback also in the form of text, speech or images and could also be used in follow-up-questions.

A US patent on method of handling FAQs in a natural language dialog service [Di Fabbrizio et al., 2014]: The interest of this project, was an invention to improve customer relationships with companies using a natural language help desk that understands and interacts with users, in a well organized and pleasant manner, especially when handling FAQs. The invention relates to dialog systems and more specifically to a system and method of providing a natural voice user interface between a human and a computing device, the natural voice user interface having a module for handling frequently asked questions.

Quora [D'angelo et al., 2013], which was co-founded by two former facebook employees, for the sole purpose of questions and answers has also had its success in providing satisfactory feedback to queries. In an article by Wang et al. [2013], we understand that Quora makes use of three internal graphs that serve complementary roles in improving its effective content discovery namely: a user-topic follow graph, a user to user social graph, and a related question graph. Quora integrates an effective social network into a tradition Question and Answer site, it has the ability to locate questions related to a given question and effectively creates a related question graph, where nodes represent questions, and links represent a measure of similarity as determined by Quora. The related question graph provides an easy way for users to browse through Quora repository of questions with similarity as a distance metric.

FINCHAN [Ade-Ibijola, 2016], a research article which looked into the automatic comprehension and summarisation of Instant Messages (IMs) exchanged on Instant Bloomberg (IB) application which is popular finance IM software. FINCHAN was implemented as a windows application, developed with Microsoft's $.Net$ framework version 4.6, and stores data in Microsoft's SQL Server database file using the Server 2014 management studio API. This designed tool, takes a raw chat and performs lexical analysis on the text using a lexer, in order to parse the chat text and lexical analysis. It identifies the syntax of groups of lexemes using some production rules of grammar and attempts to understand the chats using semantic rules.

2.3 Techniques and Components

Frequently Asked Questions (FAQs)

With the knowledge that information is an objective commodity defined by the dependency relations between distinct events [Dretske, 1981], we describe FAQs as reoccurring queries from customers to service providers in the quest to gain needed information for user knowledge.

In most cases as seen in Sarle [1995], these queries are segmented into topics for customers to find their genre of queries with ease. Example is the Uber FAQs page as seen in Figure 2.2

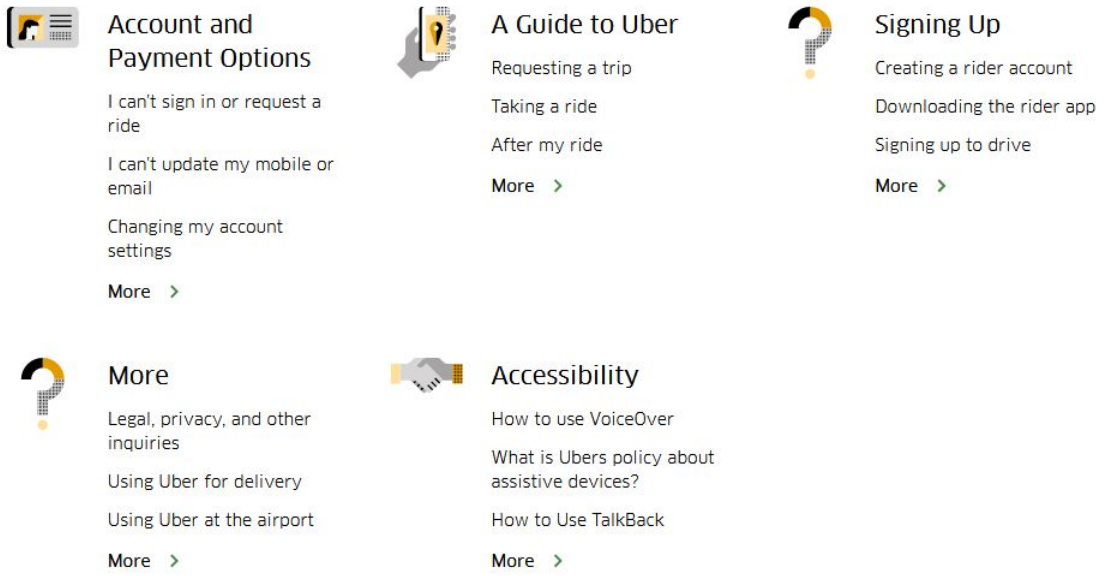


FIGURE 2.2: Uber FAQs page

Example 2.1.

In this research work, Questions Answers (QA) found in already existing FAQs database were retrieved. I believe that the most natural kind of interface to database of answers is the question stated in natural language.

Below are FAQs from *University of the Witwatersrand Computer and Network Services (CNS)* used as training data in this research work. There is a large number of FAQs on the site, but 50 FAQs in eight topics as training data were extracted. **Namely;**

1. Top Mail FAQs,
2. Archiving and Deleting Messages FAQs,
3. Spam FAQs,
4. Email Features FAQs,
5. Conversation and Messages FAQs,
6. File Attachments FAQs,
7. Labels FAQs,
8. Moble App FAQs.

- Top Mail FAQs:

Question: How do I configure my mobile device?

Answer: "Help files are available on the Google Documentation site"
Question: How do I configure my mobile device?

Answer: "Help files are available on the Google Documentation site"
Question: How do I go about forwarding my mail

Answer: "Help files are available on the Google Documentation site"
Question: How do I configure my mobile device?

Answer: "Help files are available on the Google Documentation site"
Question: How do I go about forwarding my mail

Answer: "Help files are available on the Google Documentation site"
Question: How much space do I have for mail messages? Can I run out of space?

Answer: "Your account has 25 GB of storage. You'll probably never run out of space. In fact, a heavy email user who sends and receives lots of file attachments and archives all messages might use up to 5% of this space a year (usually less), so it would take decades to use up all of the available space. Also note that Google Apps restricts the size of file attachments to 20 MB, so you don't have to worry about a few large files using up your storage space."
Question: Can I access my mail offline?

Answer: "Yes, you can set up Gmail for offline access to your most recent email messages. To enable this feature, click Settings in the upper-right corner of your Mail window, click the Offline tab, and then click Enable Offline Mail for this computer. (Note, however, that if your Google Apps administrator has disabled offline access for your domain, the Offline tab won't be available and you won't be able to use this feature.)"
Question: Can I stop messages from being grouped into conversations?

Answer: "Yes. Click Settings in the upper-right corner of your Gmail window and, on the General tab, scroll down to Conversation View. If Conversation View is off, new messages won't be grouped into conversations, and any existing conversations are ungrouped into separate messages. If Conversation View is on, you can't separate the messages in a conversation. However, if you want to send a reply but don't want it to be added to the conversation, you can simply change the subject line in your reply."

Question: How do I mark a message as "unread" in my Inbox after I open it?

Answer: "Select the message. Then, in the More drop-down list, select Mark as unread."

Question: Can I recall a message I already sent?

Answer: "Yes, Gmail Labs has an early version of a new "message undo" feature that lets you recall a message within a few seconds after you send it. To enable the feature, open your Gmail Settings, go to the Labs tab, and enable the Undo Send lab. Note that your Google Apps administrator must enable Labs for your domain for this feature to be available."

Question: I've heard Gmail search is really powerful. How does it work?

Answer: "To search for messages, type a word that the messages contain. Note, however, that Search matches "whole words" only—that is, it doesn't recognize partial or similar matches. For example, if you search for benefits, Search won't find benefit or benef. Also, Search doesn't recognize special search characters, such as square brackets, parentheses, currency symbols, the ampersand, the pound sign, and asterisks. By default, search doesn't look in your Trash or Spam folders. To search those folders also, click-show search options next to the Search field, and then, in the Search drop-down list, select Mail , Spam and Trash. You can find more information about using Search in the Gmail Help Center.A list of the advanced search operators is available in the Gmail Help Center. You can also print out this reference sheet."

Question: Can I make Gmail the default email program when I click email links?

Answer: "Yes, if Google Talk is enabled for your domain. In that case, specify Gmail as your default email program as follows: *Open Google Talk. Click Settings* in the upper-right corner of your contacts list. In the General dialog box, select Open Gmail when I click on email links. *Click OK*. Note however, that this setting doesn't work for all email links."

In the above FAQ file with topic "Top Mail", one can see that all the FAQs are organized in Q/A format, and all the information needed to determine the relevance of a Q/A pair can be found within the Q/A pair. Other FAQ files are shown in Appendix A

The process of retrieving Q/A's found in already existing FAQs files, involves ingesting large amount of semi-structured data for proper results. The ability to gain knowledge and understanding from the data ingested, and use them subsequently to answer with a level of confidence to various question. In this research work, a variety of Natural Language Processing (NLP) techniques will be used. Such system generally operate by generating a large number of hypotheses [Eggebraaten et al., 2014].

Natural Language Processing (NLP)

A language can be defined as a set of rules or set of symbols [Parkes, 2008]. Symbols are put together to convey information. NLP is primarily classified into two parts as seen in Figure 2.3 i.e. *Natural Language Understanding and Natural Language Generation* which evolves the task to understand and generate text.

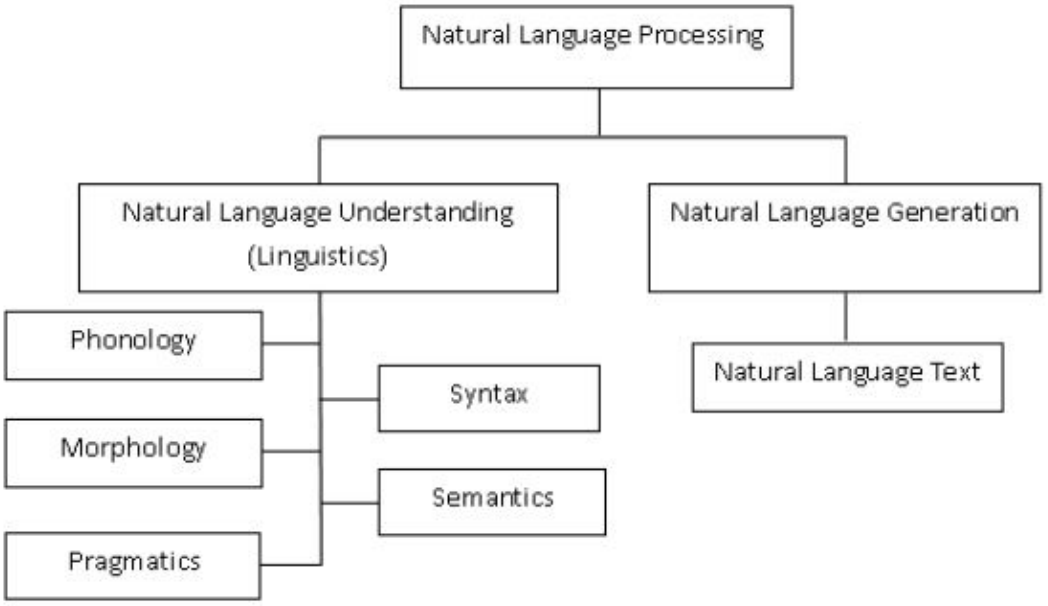


FIGURE 2.3: Broad classification of NLP: Source: Khurana et al. [2017].

NLP is a collection of theoretically motivated computational techniques, used to analyse and represent naturally occurring texts at one or more levels of linguistic analysis, for the purpose of achieving human-like language processing for a range of tasks or applications [Liddy, 2001]. A language developed naturally without modeling and planning is considered a Natural Language. A well known example is the English language. A Natural Language coequals computing code [Bordignon, 2016].

The major objective of NLP is to value and produce languages that humans use naturally. Jurafsky and Martin [2014] defined the objective of NLP as *"to get computers to perform useful tasks involving human language. Tasks like enabling human-machine communication, improving human to human communication, or simply doing useful processing of text or speech"*

As defined by Jackson and Moulinier [2007], the $natural$ label is meant to differentiate human speech and writing, from other formal languages such as mathematical notations, or programming languages where the vocabulary and syntax are comparatively restricted. NLP system's focal point is to efficiently create a structure that will process texts and make their information accessible to computer applications.

Different methods are frequently applied to specific kinds of applications, such as natural language text processing and summarisation using weighting schemes, machine translation applying statistical machine translation etc [Chowdhury, 2003].

LEVELS OF NLP:

The *levels of language*, is said to be one of the most explanatory method for representing the NLP, which helps to generate the NLP text by realizing *Content Planning*, *Sentence Planning* and *Surface Realization phases* [Reshamwala et al., 2013]. Refer to Figure 2.4.

Linguistics is the science of language which includes *Phonology* that refers to sound, *Morphology* as word formation, *Syntax* as sentence structure, *Semantics* as syntax and *Pragmatics* which refers to understanding [Khurana et al., 2017]. Refer to Figure 2.3

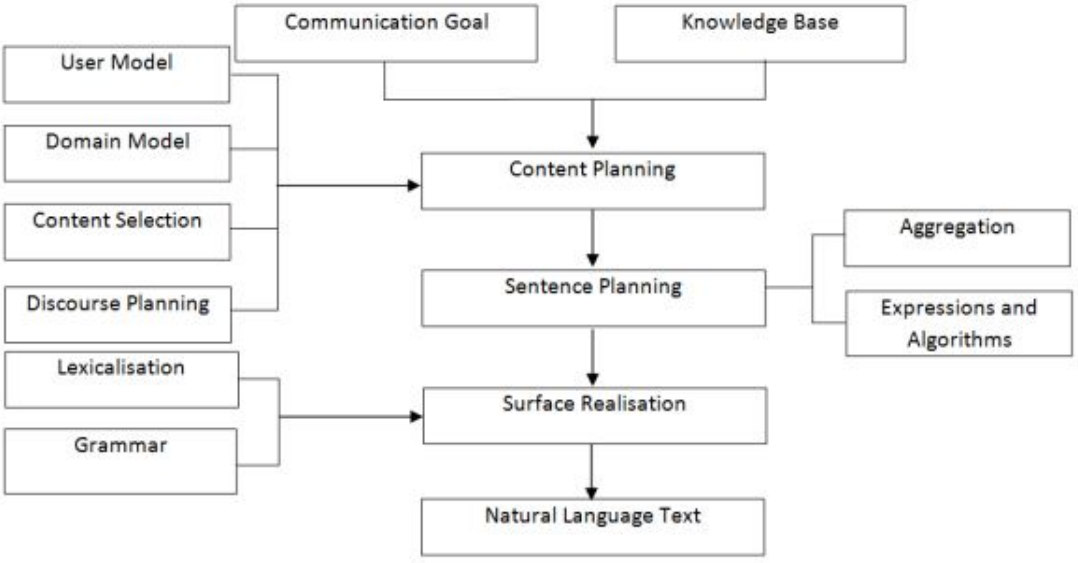


FIGURE 2.4: Phases of NLP architecture: Source: Khurana et al. [2017].

1. **Phonology:** In Linguistics, Phonology refers to the systematic arrangement of sound. The term phonology comes from *Ancient Greek*. The term *phono-* means voice or sound, and the suffix *–logy* refers to word or speech [Khurana et al., 2017]. The interpretation of speech sounds within and across words are handled by phonology. Three types of rules are used in phonological analysis:
   (a) Prosodic rules: It is used to check for fluctuation in stress and intonation across a sentence.
   (b) Phonemic rules: It is used for variations of pronunciation when words are spoken together.

(c) Phonetic rules: It is used for sound within words.

2. **Morphology:** The different parts of a word represent the smallest units of meaning known as *Morphemes*. Morphology comprises of Nature of words, and are initiated by morphemes. Reshamwala et al. [2013], described Morphology as the first stage of analysis, once input has been received. It controls the ways in which words break down into their components and the affects of their grammatical status. Morphology is used mainly to identify the parts of speech in a sentence and words that interact together. Morphology is also seen as the systematic description of words in a natural language. It describes a set of relations between words' surface forms and lexical forms.
3. **Semantics:** Semantics processing, determines the possible meanings of a sentence by pivoting on the interactions among word-level meanings in the sentence [Khurana et al., 2017]. It builds up a representation of the objects and actions that a sentence is describing and includes the details provided by adjectives, adverbs and propositions. This level of processing can incorporate the semantic disambiguation of words with multiple senses; gathers information vital to the pragmatic analysis in order to determine which meaning was intended by the user.
4. **Pragmatics:** Pragmatics is "the analysis of the real meaning of an utterance in a human language, by disambiguating and contextualizing the utterance" [Reshamwala et al., 2013]. Pragmatic is concerned with the strong use of language in situations and utilizes nub over and above the nub of the text in order to understand the goal and to explain how extra meaning is read into texts without literally being encoded in them Liddy [2001].
5. **Syntactic:** This level ensures to scrutinize the words in a sentence so as to uncover the grammatical structure of the sentence. Syntax involves applying the rules of the target language's grammar. Its task is to determine the role of each word in a sentence and organize this data into a structure that is more easily manipulated for further analysis.

Stanford CoreNLP by Manning et al. [2014], is an NLP core example. It is capable of providing base forms of words, their parts of speech and put them into categories. It is a system that provides set of natural language analysis tools i.e. company names, communities etc. Also it gives normalized dates, time and numeric qualities and makes the structure of sentences terms of phrases and word dependencies. With the rapid growth of full text databases, alongside developments in NLP technology, NLP researchers have suggested that it could be useful to apply text retrieval, primarily for indexing purposes but perhaps also for more

or less related tasks such as document retrieval. Hence Information Retrieval [Jones, 1999].

Information Retrieval (IR)

Information Retrieval (IR) is the discipline that deals with retrieval of unstructured data, especially textual documents, in response to a query or topic statement, which may itself be unstructured, e.g., a sentence or even another document, or which may be structured, e.g., a boolean expression [Greengrass, 2000]. IR systems rank documents by their estimation of the usefulness of a document for a user query, and most IR systems assign a numeric score to every document and rank documents by this score [Singhal, 2001]. The need for effective methods of automated IR has grown in importance because of the tremendous explosion in the amount of unstructured data, both internal, corporate document collections, and the immense and growing number of document sources on the internet [Greengrass, 2000].

In this procedure standard information-retrieval technology is used, the public-domain $SMART$ (System for the Mechanical Analysis and Retrieval of Text) information retrieval package. It is a set of programs composing a fully automatic document retrieval system [Buckley, 1985]. Since the early 1960's the $SMART$ project has tested out new ideas in information science aimed at fully automatic document retrieval [Fox, 1983]. Beginning in 1980, development of an enhanced and generalized version of $SMART$ has progresses at Cornell with the goal of investigating the effectiveness and efficiency of automatic methods of retrieval of text [Buckley et al., 1993].

With the above knowledge, IR is very relevant in response to queries, as it is a computer application with a primary task of processing those texts where NLP plays essential role. In this work, it is tasked to make a choice from a set of documents, relevant to every customer query. In other words, it is concerned with building this system to accept a natural language query and return a list of documents ranked according to their estimated relevance to the customer's information needs.

Text Categorisation

Due to the increased availability of documents in the digital form and ensuing need to organize them, the automated categorisation of texts into predefined categories has witnessed a booming interest in the last decades [Sebastiani, 2002]. In this work, text categorisation is important as it is given the task of automatically assigning FAQs into their respective categories for proper matching by the FA for effective response.

WORDNET

WORDNET is an online lexical database designed for use under program control [Miller, 1995]. [1]

WORDNET is a database of English words that are linked together by their semantic relationships . It provides a system of relations between words and synonym sets and between synonym sets themselves. *WORDNET* is like a supercharged dictionary/thesaurus with a graph structure. As our proposed system hopes to process natural language text files, information about words and their meanings are needed. It provides a more effective combination of traditional lexicographic information and modern computing. Using a marker-passing algorithm [Quillian and Memory, 1968], this system uses *WORDNET* database to accept variations. Marker-passing is performed, to compare each word in the user's question with each word in the FAQ file. Our system obtains its knowledge of shallow lexical semantics from *WORDNET*, a semantic network of English words [Miller, 1995].

Formal Languages and Automata

Since Formal Language Theory was developed in the mid 1950's, in an attempt to develop theories of natural language acquisition, it has been realized that the theory is quite relevant to the artificial languages that had originated in computer science [Harrison, 1978].

This study, constitutes an important sub-area of Computer Science. The development after Chomsky [1956], gave a mathematical model of a grammar in connection with his study of natural languages, that made the concept of grammar relevant to the programmer when syntax of the programming language AGOL was defined by a context free grammar. This naturally led to syntax-directed compiling and the concept of compiler compiler [Hopcroft and Ullman, 1969]. The theory of formal language has been developed comprehensively, with several discemible trends, including applications to the syntactic analysis of programming languages, programming schemes, models of biological systems and relationships with natural languages [Harrison, 1978].

This work is in the domain of computational linguistics, and in the theory of computation, the simpler abstract machine is Finite Automata (FA). The Automata Theory was applied to abstract FAQs and came up with a graph algorithm to match the QAs. FLA will deal with the formal notion of strings i.e. the basic element of a language. The term "Automata" means "self-acting". Automata

[1] www.stevenloria.com/tutorial-wordnet-textblob/

Theory is a branch of computer science that deals with designing abstract self-propelled computing devices that follow a predetermined sequence of operations automatically. An automaton with a finite number of states is called a Finite Automaton(FA).

FA is the five-tuples combination focusing on states and transition through input characters Ezhilarasu et al. [2015] as seen below:

1. Q is a finite set of states.
2. Σ is a finite set of symbols, called the alphabet of the automaton.
3. δ is the transition function.
4. q_0 is the initial state from where any input is processed ($q_0 \in Q$).
5. F is a set of final state of Q ($F \subseteq Q$).

But considering we are dealing with unstructured data, Jumping Finite Automata (JFA) was used to model this research work. More insight is given about JFA in Section 2.4

2.4 JFA as an Abstract Machine

JFA, is a recent model of Finite Automata (FA), that reads input discontinuously [Meduna and Zemek, 2014a]. This automata, work just like classical finite automata except that they do not read their input strings in a symbol-by-symbol left-to-right way. It enables a jump at any position of the input while processing strings.

Definition 2.1. Finite Automata (FA) is a recognizer for regular languages. Informally, a state machine that comprehensively captures all possible states and transitions that a machine can take while responding to a stream of input symbols. FA has Jumping Finite Automata (JFA) which was introduced by Meduna and Zemek [2014a], as one of its processing model.

JFA reads input words discontinuously, e.g. [2, 5, 8, 10, 11, 18]. It enables a jump at any position of the input while processing strings [Meduna and Zemek, 2014a]. In other words, after reading a symbol, they can jump over some symbols within the words and continue their computation from there. This also means they can start from the beginning, start from even the end but with goal of reaching the accepting state.

JFA is a quintuple automaton. Following Meduna and Zemek [2012], a general finite machine is denoted as $M = (Q, \Sigma, R, s, F)$ consisting:

1. Q is a finite set of states,
2. Σ is the input alphabet of the automaton,
3. R is the finite set of rules of the form $py \rightarrow q$ ($p, q \in Q, y \in \Sigma^*$),
4. $s \in Q$ is the start state, and
5. $(F \subseteq Q)$ is a set of final state.

If all rules $py \rightarrow q \in$ R satisfy $\mid y \mid \leq 1$, then M is a finite machine.

M can be interpreted in two ways:

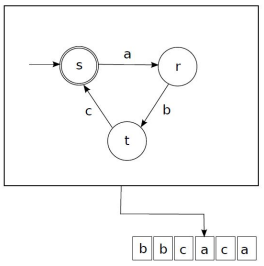
1. As a (general) finite automaton: a configuration of M is any string in Q Σ^*, the binary move relation on Q Σ^*, written as $\Rightarrow$, is defined as follows:
   $pw \Rightarrow qz \Leftrightarrow \exists\, py \rightarrow q \in R : w = yz$.
2. As a (general) jumping finite automaton: a configuration of M is any string in Σ^* Q Σ^*, the binary jumping relation on Σ^* Q Σ^*, written as $\curvearrowright$, satisfies:
   $vpw \curvearrowright v\prime qz\prime \Leftrightarrow \exists\, py \rightarrow q \in$ R $\exists$ z $\in \Sigma^*$: w = yz $\wedge vz = v\prime z\prime$.

   Hence the following languages are obtain from a (general) finite machine M:
   L_{FA} (M) = {w $\in \Sigma^*$: $\exists\, f \in$ F : $sw \Rightarrow^* f$ },
   L_{JFA} = {w $\in \Sigma^*$: $\exists\, u, v \in \Sigma^*$ $\exists\, f \in$ F : w = $uv \wedge usv \curvearrowright^* f$ }.

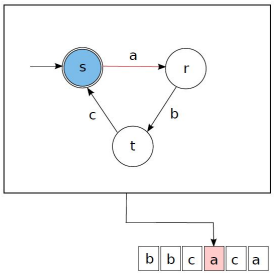
   The above defines the language classes *REG* (accepted by finite automata), *JFA* (accepted by JFAs) and *GJFA* (accepted by GJFAs). Moreover, *CFL* denotes the class of context-free languages [Fernau et al., 2017].

2.4.1 Why JFA?

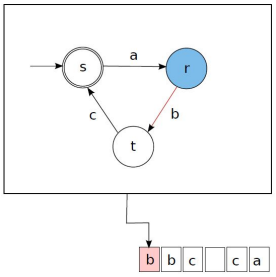
In the previous century, most classical computer science methods were developed for continuous information processing [Meduna and Zemek, 2012]. Accordingly, their formal models, such as finite automata, work on words, representing information, in a strictly continuous left to-right symbol-by-symbol way. Modern information methods, however, are often designed for a different way of information processing. In recent times, computational methods such as JFA frequently process information in a discontinuous way [3, 5, 6, 10, 19, 27] as mentioned earlier in Definition 2.1 see Figure 2.5a to Figure 2.5h below, unlike FA that reads its input strings in a linear logic manner [1, 2, 3, 4, 5, 6].



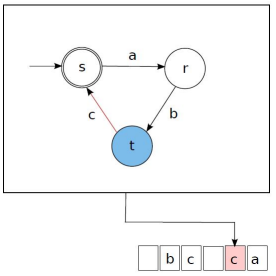
(A) Start state



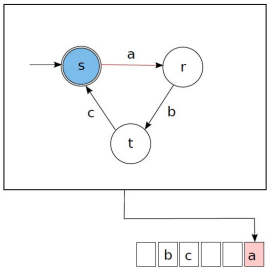
(B) 2nd state



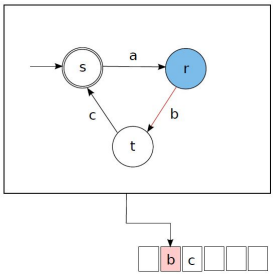
(C) 3rd state



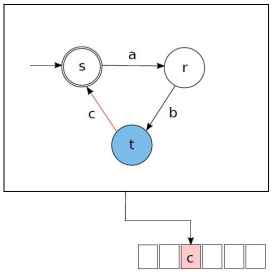
(D) 4th state



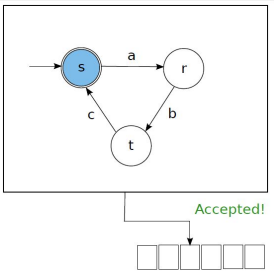
(E) 5th state



(F) 6th state



(G) 7th state



(H) Accepting state

A typical computational step, may be performed somewhere in the middle of information within a particular running process, while the very next computational step is executed far away from it. This implies, that before the next step is carried out, the process has to jump over a large portion of the information to the desired position of execution. Of course, classical formal models, which work on words strictly continuously, inadequately and inappropriately reflect discontinuous information processing of this kind. Formalizing discontinuous information processing adequately gave rise to the idea of adapting classical formal models in a discontinuous way.

2.4.2 Comparative Analysis of Classical FA and JFA

As it has already established earlier that classical FA reads input strings in a linear logic manner while JFA reads in a non-linear logic manner, a statistical comparison is shown.

Referencing one of the queries in the FAQs training data set; How do I configure my mobile device?

There are four entities in this query;

a. How, b. configure, c. mobile, d. device.

Therefore;
we have a set of alphabets $\{a, b, c, d\}$

In the event where a query such as the following is posed; *Hi, I am having a challenge with my* $mobile_c$ $device_d$, *please* how_a *can I* $configure_b$ *it?*

Hence;
The set of alphabets from the query posed are $\{c, d, a, b\}$

In the above query, there are four entities matching the entities presented but not sequentially. The manner at which classical FA and JFA will read these input alphabets will differ.

During computation, classical FA will struggle to read these input alphabets at the shortest time possible considering they are not linearly arranged in the query considering below conditions;

Let w = a_i, a_r....a_j be a string over an alphabet $\sum$. The automaton M accepts the string w if a sequence of states, q_0, q_1,... q_i exists in Q with the following conditions:

1. r_0 = q_0
2. r_i + 1 = δ (r_i , a_i + 1) for $i = 0, \ldots,$ n-1
3. $rn \in F$

1. The machine starts in the start state.
2. The machine goes from state to state according to the transition function.
3. The machine accepts its input if it ends up in an accepting state.

Classical FA cannot function outside these rules.

This is where JFA makes computation easy, because it will read the input alphabets just the way they are in the query and compute the output immediately.

Hence;

Where L (M) is the language accepted by the automata M.

$abcda \curvearrowright abcd \; [S_0c \rightarrow S_1]$

$\curvearrowright abd \quad [S_1d \rightarrow S_2]$

$\curvearrowright ab \quad [S_2a \rightarrow S_3]$

$\curvearrowright b \quad [S_3b \rightarrow S_4]$

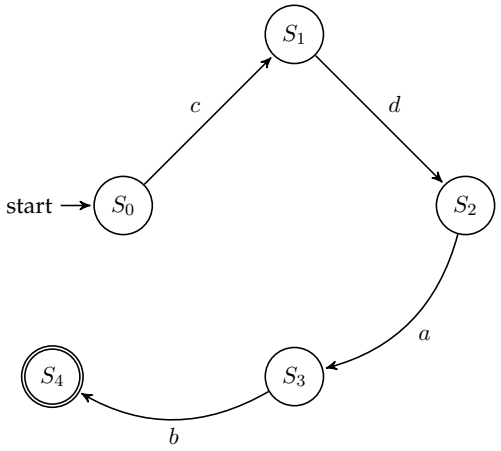


FIGURE 2.6: JFA Computation

2.5 Algorithms used in Natural Language Question Answering Systems

An algorithm is a sequence of computational steps that transform an input into an output [Cormen et al., 2001]. Furthermore, an algorithm informally is any well-defined computational method that takes some value, or set of values, as input and produces some value, or set of values, as output. An algorithm can be detailed in English, as a computer program, or even as a hardware design. Its only condition, is that the specification must provide a precise description of the computational procedure to be followed. In this section, we discuss some algorithms used in natural language question answering system.

2.5.1 Cocke–Younger–Kasami Algorithm (CYK)

The Cocke–Younger–Kasami algorithm (alternatively called CYK, or CKY), is one of the earliest recognition and parsing algorithms. It is a parsing algorithm for context-free grammars, named after its inventors, John Cocke, Daniel Younger and Tadao Kasami [Hambir and Srivastav, 2012]. This algorithm, employs bottom-up (data-driven from the symbols up) parsing and dynamic programming. Its standard version, can only recognize languages defined by context-free grammars in Chomsky Normal Form (CNF). In CKY, there is a possibility that its algorithm can be extended to handle some grammars which are not in CNF.

CKY agorithms considers every possible consecutive sub-sequence of letters and sets $K \in T[i, j]$ if the sequence of letters starting from i to j can be generated from the non-terminal K.

- Once sequences of length 1 is considered, it transits on to sequences of length 2, and so on.
- It considers every possible partition of the sub-sequence into two halves for sub-sequences of length 2 and greater, and checks to see if there is some production $A \rightarrow BC$ such that B matches the first half and C matches the second half.
- If so, it records A as matching the whole sub-sequence.
- Once this process is completed, the sentence is recognized by the grammar if the entire string is matched by the start symbol.

2.5.2 Earley's Algorithm

Earley's algorithm, named after its inventor, Jay Earley [Earley, 1970], is a chart parser that uses dynamic programming; it is mainly used for parsing in computational linguistics. This algorithm is an efficient top-down parsing algorithm that avoids some of the inefficiency associated with purely naive search with the same top-down strategy.

1. Intermediate solutions are created only once and stored in a chart (dynamic programming).
2. Left-recursion problem is solved by examining the input.
3. Earley is not picky about what type of grammar it accepts, i.e., it accepts arbitrary CFGs (cf. CKY)

Earley's algorithm is a context free top-down parsing algorithm. Which makes it a goal driven algorithm. It offers a dynamic programming approach and it is a chart parsing algorithms.

2.5.3 Shift-Reduce Parsing Algorithm

Shift-reduce parsing [Aho and Johnson, 1974] attempts to construct a parse tree for an input string beginning at the leaves and working up towards the root. Meaning, it is a process of "reducing" (opposite of deriving a symbol using a production rule) a string w to the start symbol of a grammar. At every (reduction) step, a particular substring matching the Right Hand Side(RHS) of a production rule is replaced by the symbol on the Left Hand Side(LHS) of the production. A general form of shift-reduce parsing is LR (scanning from Left to right and using Right-most derivation in reverse) parsing, which is used in a number of automatic parser generators like *Yacc, Bison,* etc.

One advantage of shift-reduce parsers, is that the scoring model can be defined over actions, allowing highly efficient parsing by using a greedy algorithm, in which the highest scoring action (or a small number of possible actions) is taken at each step [Zhang and Clark, 2011].

Shift/Reduce Parsing Algorithm

Input:

input string, w, and an LR parsing table, T, for grammar G, with functions ACTION and GOTO. The Parse Table has one row for each "state", an ACTION column for each terminal symbol and a GOTO column for each symbol which is the left-hand side of a production in G. (Presumably, this will be equivalent to the non-terminal symbols.)

Output:

if w is in L(G), a bottom-up parse for w; otherwise, an error indication

Data Structures:

a stack whose elements are terminal symbols and/or state numbers, a pointer, ip, to the next input symbol, AND an array of productions in G

Initial State:

the stack consists of the single state, s0; ip points to the first character in w.

The algorithm:

loop forever:

for top-of-stack symbol, s, and next input symbol, a case action of T[s,a]

shift x: (x is a STATE number) push a, then x on the top of the stack and advance ip to point to the next input symbol.

reduce y: (y is a PRODUCTION number) Assume that the production is of the form
A ⇒ beta
pop 2 * —beta— symbols of the stack. At this point the top of the stack should be a state number, say s'. push A, then goto of T[s',A] (a state number) on the top of the stack.
Output the production A ⇒ beta.

accept:

return — a successful parse.

default:

error — the input string is not in the language.

2.6 Conclusion

In this chapter, the background related to this research was presented. The idea behind question answering was also presented. A number of techniques used in designing natural language question answering systems were discussed. The techniques and components of this system were presented. Lastly, few number of algorithms used in natural language question answering systems were highlighted.

Chapter 3

Design, Implementation and Results

3.1 Introduction

The previous chapter outlined the aims, questions to be answered and methods of this research. This chapter is focused on the design and implementation of the Question Answering system using JFA. Section 3.2 deals with the JFA design. Section 3.3 is focused on answering the research questions formulated in Subsection 1.4.2. Section 3.5 shows the research results and discussions from the experiment. Section 3.6 concludes the chapter.

The operation of this work will be relatively simple to the user. First is to narrow the search to a single FAQ file, which is likely to contain an answer to the user query. The answer to the search will automatically be confirmed by mapping the JFAs to the FAQs. The FAQs file is considered to be a set of natural language QA pairs.

An algorithm is set to convert these improved automata to the matching JFA, that will be used to process strings.

3.2 JFA Design

In this section, we depict our JFA design using a flow chart in Figure 3.1, showing clearly the processing concept of this system. Briefly, each of the processing stages in the model is described and capped with an algorithm.

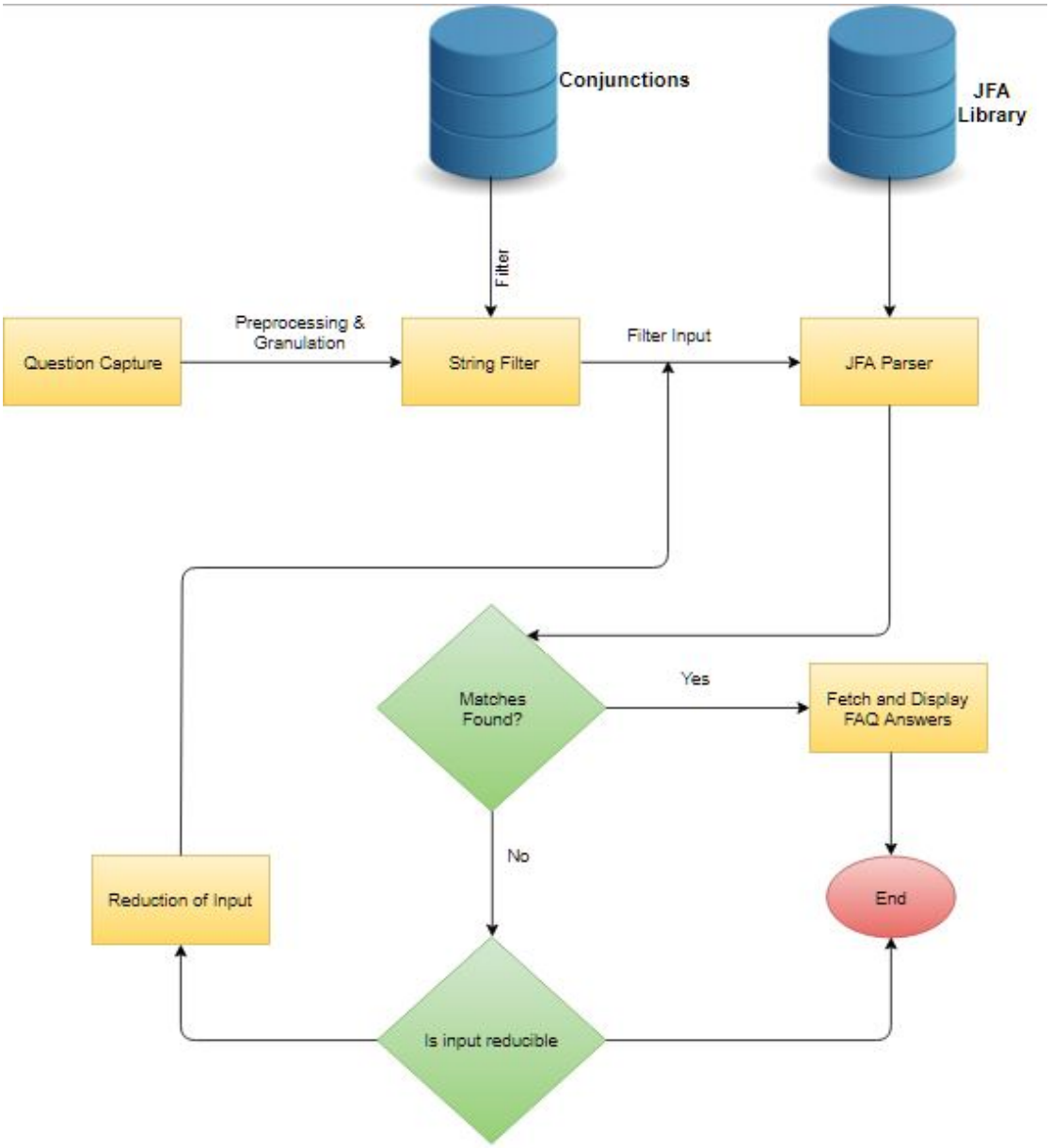


FIGURE 3.1: JFA Design

1. *Question Capture*: The natural language question input by the user needs to be captured and analysed into whatever form/forms needed by subsequent parts of the system. The question may be interpreted in the context of an on-going dialogue, and in the light of a model which the system has of the user. In this stage, the user could be asked to clarify his or her question before proceeding. Considering that our system has access to a corpus of document collection as a knowledge resource for answering questions, this collection, needs to be processed(Preprocessing and Granulation) before querying, in order to transform it into a form which is appropriate for real-time question answering.

2. *String Filter*: A subset of documents(filtered) from the total document collection

(conjunctions) is selected, comprising those documents deemed most likely to contain an answer to the user query.

3. *JFA Parser*: This subset of documents(filtered), from the data set goes through our JFA library in order to analyze (sentences) in terms of their grammatical constituents. Refer to Table (3.2)

4. *Fetch and Display FAQ Answer*: After analyzing the sentences in our JFA parser and matches found in the JFA library, the system fetches and displays FAQ answers to the user and ends the process.

5. *Reduction of Input*: In the case where a match is not found, the system will check if the input is reducible assuming the query is still ambiguous. If it is reducible, it backtracks to reduce input $\rightarrow$ JFA parser in order to find a match. If in any case, it is not reducible, it ends the process.

3.2.1 Algorithms

Algorithm 1 JFA algorithm

```
1:  function JFA_PARSER(text, jfa_list[ ], threshold) returns status
2:      status ⟵ {failed, 0.0} // {parsing status, percentage matched}
3:      set input_text ⟵ normalise_input (input_text)
4:      set matched_jfas to 0
5:      convert text to text_array
6:      for each jfa in jfa_list do
7:          for each user_word in text_array do
8:              if (jfa contains user_word) OR (jfa contains synonym (user_word)) then
9:                  POP user_word from jfa
10:             end if
11:         end for
12:         if jfa is empty then
13:             jfa recognises user text
14:             increment matched_jfas by 1
15:         end if
16:     end for
17:     set percentageMatched to matched_jfas / LEN(jfa_list[ ])*100
18:     if percentageMatched >= threshold then
19:         status ⟵ {success, percentageMatched}
20:     else
21:         status ⟵ {failed, percentageMatched}
22:     end if
23:     return status
24: end function
```

Algorithm 2 Text Normalisation

```
1:  function NORMALISE_INPUT(user_input, acceptance_rate)
2:      for each user_text in user_input do
3:          if user_text.Length is greater than 5 then
4:              set dictionary ⟵ load_english_dictionary()
5:              if dictionary.contains(user_text) is false then
6:                  for each dict_word in dictionary do
7:                      if calculateLevenshtein(word, user_text) >= acceptance_rate then
8:                          return word
9:                          ExitFor
10:                     end if
11:                 end for
12:             end if
13:         else
14:             return user_text
15:         end if
16:     end for
17: end function
```

3.3 Implementation

In this section, we answer the research questions as posed in the previous Chapter 1.4.2.

The following will be discussed.

1. Abstracting FAQs to JFA
2. Generating feedback from FAQs using JFA

3.3.1 Abstracting FAQs to JFA

In Kramer [2007], abstraction is described as the process of removing characteristics from a data set, in order to reduce it to a set of essential characteristics. In other words, through the process of abstraction, a programmer hides all but the relevant data about an object in order to reduce complexity and increase efficiency. An NLP based IR system has more goals of giving accurate and complete information in response to a user's actual information need [Chatterjee et al., 2015]. The task of answering natural language questions by abstracting FAQs to JFA to build an algorithm for QA matching is addressed. Part of our intent for initiating the categorisation process in [fig: FAQs categorisation] below, is to make heading information available to the machine.

The following is addressed to help understand the abstraction process:

1. The states in this work
2. The transition process
3. How data is represented

i. **The states in this work**

The alphabets in the FAQs are used as the states. In this work there are three entities that make up the alphabets in each query i.e. the recognizer for the JFA.

i Query Type (Green)

ii Action Type (Red)

iii Object Type (Blue)

Let's say:

1. $\sum_{\text{query}} = i$
2. $\sum_{\text{action}} = j$
3. $\sum_{\text{object}} = k$

The state is therefore the combination of the symbols/alphabets, whether single or multiple which is $_i, _j, _k = a_0, b_1, c_2, \ldots q_i$

1. How do I configure my mobile device?
2. How do I go about forwarding my mail?
3. How much space do I have for mail messages? Can I run out of space?
4. Can I access my mail offline ?
5. Can I stop messages from being grouped into conversations?
6. How do I mark a message as unread in my inbox after I open it?
7. Can I recall a message I already sent?
8. I've heard Gmail search is really powerful. How does it work?

9. Can I make Gmail the default email program when I click email links?

Each query from *Top Mail FAQs* category above, contains Σ. In this system, the transition process is modelled using a JFA. Generally JFA is used for content sensitive languages. A running process within a typical computational step may be performed somewhere in the middle of information being processed, while the very computational step is executed far away from it. In other words, in JFA before the next step is carried out, the process has to jump over a large portion of the information to the desired position of execution.

ii. The transition process

In this research work, $Graph\ traversal$ is used to represent δ (transition process), considering this work is dealing with automata.

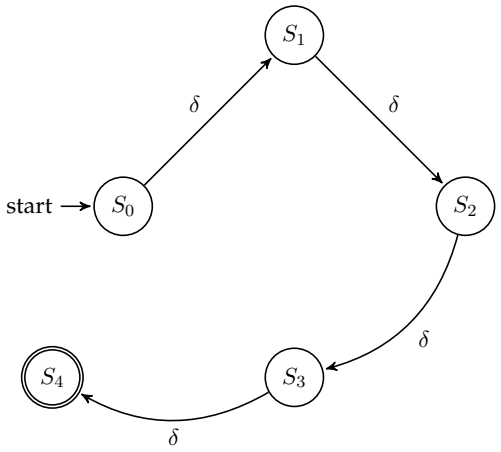


FIGURE 3.2: Transition process

Example 3.1.

JFA symbols		
a_i	b_j	c_k
a_0 *Where*	b_0 *Find*	c_0 *WitsMapp*
a_1 *How*	b_1 *Remove*	c_1 *App*
a_2 *Which*	b_2 *Access*	c_2 *Features*
a_3 *What*	b_3 *See*	c_3 *Comms*
a_4 *Can I*	b_4 *Change*	c_4 *Dashboard*
a_5 *When*	b_5 *Delete*	c_5 *Device*
a_6 *Why*	b_6 *Stay*	c_6 *Resources*
a_7 *Should*	b_7 *Move*	c_7 *Home-screen*
a_8 *Does*	b_8 *Archived*	c_8 *Message*
a_9 *Is there*	b_9 *Archive*	c_9 *Trash*
–	b_{10} *Deleted*	c_{10} *Inbox*
–	b_{11} *Show up*	c_{11} *Sent Folder*
–	b_{12} *Reply*	c_{12} *Conversation*
–	b_{13} *Forward*	c_{13} *Window*
–	b_{14} *Open*	c_{14} *Gmail*
–	b_{15} *Spell-check*	c_{15} *Office*
–	b_{16} *Have*	c_{16} *Feature*
–	b_{17} *Share*	c_{17} *Email*
–	b_{18} *Support*	c_{18} *Employee*
–	b_{19} *Add*	c_{19} *Keyboard*
–	b_{20} *Drag*	c_{20} *Mailboxes*
–	b_{21} *Drop*	c_{21} *Meal*
–	b_{22} *Copy*	c_{22} *Program*
–	b_{23} *Attach*	c_{23} *Mailbox*
–	b_{24} *Organize*	c_{24} *Tasks*

TABLE 3.1: JFA symbols 1 - 24

Example 3.2.

JFA symbols		
a_i	b_j	c_k
–	b_{25} *Create*	c_{25} *List*
–	b_{26} *Apply*	c_{26} *File*
–	b_{27} *Nest*	c_{27} *Attachments*
–	b_{28} *Nested*	c_{28} *Attachment*
–	b_{29} *Deleting*	c_{29} *Folders*
–	b_{30} *View*	c_{30} *Labels*
–	b_{31} *Book*	c_{31} *Label*
–	b_{32} *Log*	c_{32} *Program*
–	b_{33} *Forgot*	c_{33} *Mail*
–	b_{34} *Do*	c_{34} *Program*
–	b_{35} *Remember*	c_{35} *Details*
–	b_{36} *Signout*	c_{36} *Multiple*
–	b_{37} *Remain*	c_{37} *Devices*
–	b_{38} *Prevent*	c_{38} *Person*
–	b_{39} *Tagged*	c_{39} *Spam*
–	b_{40} *Configure*	c_{40} *Folder*
–	b_{41} *Forwarding*	c_{41} *Specific*
–	b_{42} *Stop*	c_{42} *Senders*
–	b_{43} *Grouped*	c_{43} *Mobile*
–	b_{44} *Mark*	c_{44} *Space*
–	b_{45} *Recall*	c_{45} *Messages*
–	b_{46} *Work*	c_{46} *Offline*
–	b_{47} *Make*	c_{47} *Links*
–	b_{48} *Click*	–
–	b_{49} *Sent*	–
–	b_{50} *Limitation*	–
–	b_{51} *Configuring*	–
–	b_{52} *go*	–

TABLE 3.2: JFA symbols 25 - 52

Table (3.1) and Table (3.2) contains the JFAs symbols of this work (recognizer for the machine). It's not just about parsing text only, the context to provide better answers needs to be understood. Thus the reason the JFAs from the FAQs we have were extracted.

Given: 1st query from *Top Mail FAQs* category

Let's say we have the query:

1. How do I configure my mobile device?

M = $(\{S_0; S_1; S_2; S_3; S_4\}, \{a_i, b_j, c_k\}, R, s; \{S_4\})$
$\{S_0; S_1; S_2; S_3; S_4\}$ are the states,
$\{a_i, b_j, c_k\}$, are the input alphabets,
R is the finite set of the rules.
s is the start state,
$\{S_4\}$ is the accepting state.
With

R = $\{S_0 a_i \rightarrow S_1, S_1 b_j \rightarrow S_2, S_2 c_k \rightarrow S_3, S_3 c_k \rightarrow S_4\}$
Accepts

L (M) = $\{w \in \{a_i, b_j, c_k\}^*: |a_i| = 1 \leq |b_j| \leq 3 = 1 \leq |c_k| \leq 3\}$
i.e. $b_j = \{b_{40}\}$; $c_k = \{c_{43}, c_5\}$
Where L (M) is the language accepted by the automata M.
$b_j a_i c_k b_j c_k S_0 a_i \curvearrowright b_j a_i c_k S_1 b_j c_k \; [S_0 a_i \rightarrow S_1]$
$\curvearrowright b_j a_i c_k S_1 c_k \qquad [S_1 b_j \rightarrow S_2]$
$\curvearrowright b_j a_i S_2 c_k \qquad [S_2 c_k \rightarrow S_3]$
$\curvearrowright b_j a_i S_3 \qquad [S_3 c_k \rightarrow S_4]$

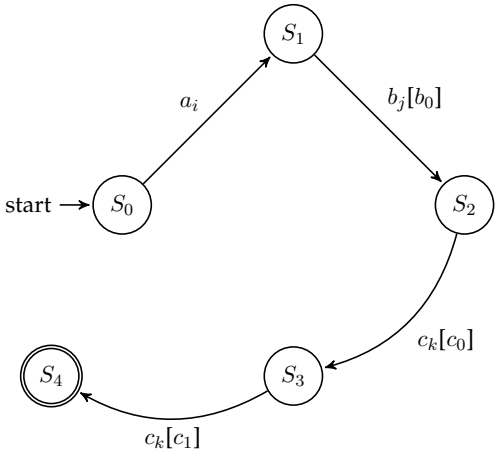


FIGURE 3.3: Top Mail 1st Query

Given: 1st and 2nd queries from *Top Mail FAQs* category:

1. How do I configure my mobile device?
2. How do I go about forwarding my mail?

M = ($\{S_0; S_1; S_2; S_3; S_4; S_5; S_6\}, \{a_i, b_j, c_k\}, R, s$;

$\{S_4\}\{S_6\}$)

With

R = $\{S_0a_i \rightarrow S_2, S_2b_j \rightarrow S_1, S_1c_k \rightarrow S_3, S_3c_k \rightarrow S_4, S_0a_i \rightarrow S_1, S_1b_j \rightarrow S_5, S_5c_k \rightarrow S_6\}$

Accepts

L (M) = $\{w \in \{a_i, b_j, c_k\}^*: |a_i| = 1 \leq |b_j| \leq 3 = 1 \leq |c_k| \leq 3\}$

i.e. $b_j = \{b_{40}, b_{52}, b_{41}\}$; $c_k = \{c_{43}, c_5, c_{33}\}$

Where L (M) is the language accepted by the automata M.

$b_jb_ja_ia_ic_kc_kb_jb_jc_kc_kS_0a_ia_i$

$\curvearrowright b_jb_ja_ia_ic_kc_kS_1b_jb_jc_kc_ka_i$ $[S_0a_i \rightarrow S_1]$

$\curvearrowright b_jb_ja_ia_iS_2c_kc_kb_jc_kc_ka_i$ $[S_1b_j \rightarrow S_2]$

$\curvearrowright b_jb_ja_ia_iS_3c_kb_jc_kc_ka_i$ $[S_2c_k \rightarrow S_3]$

$\curvearrowright b_jb_ja_ia_ic_kb_jS_4c_ka_i$ $[S_3a_1 \rightarrow S_4]$

$\curvearrowright b_jb_jS_0a_ic_kb_jc_ka_i$ $[S_0a_i \rightarrow S_1]$

$\curvearrowright S_1a_ic_kb_jc_ka_i$ $[S_1b_j \rightarrow S_5]$

$\curvearrowright a_ic_kb_jS_5a_i$ $[S_5c_k \rightarrow S_6]$

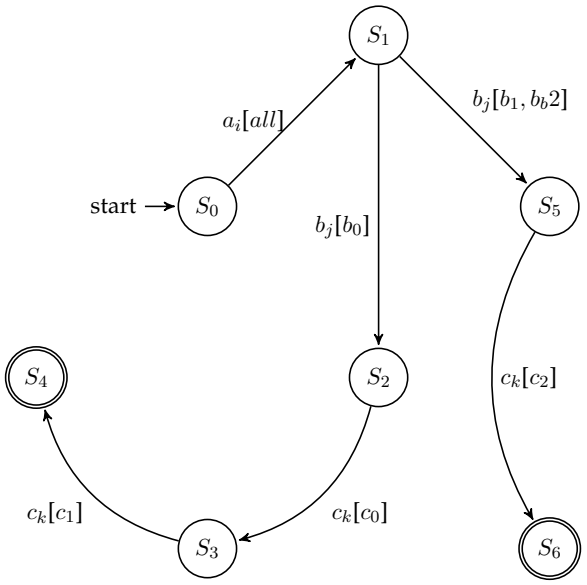


FIGURE 3.4: Top Mail 1st and 2nd Queries

Given: Complete queries from *Top Mail FAQs* category

1. How do I configure my mobile device?
2. How do I go about forwarding my mail?
3. How much space do I have for mail messages? Can I run out of space?
4. Can I access my mail offline ?
5. Can I stop messages from being grouped into conversations?
6. How do I mark a message as unread in my inbox after I open it?
7. Can I recall a message I already sent?
8. I've heard Gmail search is really powerful. How does it work?
9. Can I make Gmail the default email program when I click email links?

M = $(\{S_0; S_1; \ldots S_{38}\}, \{a_i, b_j, c_k\}, R, s; \{S_4\}\{S_6\}\{S_{15}\}$

$\{S_{18}\}\{S_{23}\}\{S_{26}\}\{S_{30}\}\{S_{34}\}\{S_{38}\})$

With

R = $\{S_0a_i \rightarrow S_2, S_2b_j \rightarrow S_1, S_1c_k \rightarrow S_3, S_3c_k \rightarrow S_4, S_0a_i \rightarrow S_1, S_1b_j \rightarrow S_5, S_5c_k \rightarrow S_6\}S_0a_i \rightarrow S_8, S_8b_j \rightarrow S_9, S_9c_k \rightarrow S_{10}, S_{10}b_j \rightarrow S_{11}, S_{11}c_k \rightarrow S_0, S_0a_i \rightarrow S_{14}, S_{14}b_j \rightarrow$

$S_{15}, S_{15}c_k \rightarrow S_{16}, S_{16}a_i \rightarrow S_{17}, S_{17}b_j \rightarrow S_{18}, S_{18}c_k \rightarrow S_{16}, S_{16}a_i \rightarrow S_{20}, S_{20}b_j \rightarrow S_{21}, S_{21}c_k \rightarrow S_{22}, S_{22}b_j \rightarrow S_{23}, S_{23}c_k \rightarrow S_0, S_0a_i \rightarrow S_{25}, S_{25}b_j \rightarrow S_{26}, S_{26}c_k \rightarrow S_{27}, S_{27}b_j \rightarrow S_{16}, S_{16}a_i \rightarrow S_{29}, S_{29}b_j \rightarrow S_{30}, S_{30}c_k \rightarrow S_{31}, S_{31}b_j \rightarrow S_{32}, S_{32}c_k \rightarrow S_0, S_0a_i \rightarrow S_{34}, S_{34}b_j \rightarrow S_{16}, 16a_i \rightarrow S_{36}, S_{36}c_k \rightarrow S_{37}, S_{37}b_j \rightarrow S_{38}$

Accepts

L (M) = $\{w \in \{a_i, b_j, c_k\}^*: |a_i| = 1 \leq |b_j| \leq 3 = 1 \leq |c_k| \leq 3\}$

i.e. $b_j = \{b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}\ , b_{14}\}$;

$c_k = \{c_0, c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9, c_{10}, c_{11}, c_{12}, c_{13}, c_{14}\}$

Where L (M) is the language accepted by the automata M.

$b_jb_jb_jb_jb_jb_jb_ja_ia_ia_ia_ic_kc_kc_kc_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_kc_kS_0a_ia_ia_ia_i$

$\curvearrowright b_jb_jS_1b_jb_jb_jb_ja_ia_ia_ia_ic_kc_kc_kc_kc_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_1b_j \rightarrow S_2]$

$\curvearrowright b_jb_jb_jb_jb_jb_jb_ja_ia_ia_ia_ic_kc_kc_kc_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kS_3c_kc_kc_ka_ia_ia_ia_i$ $[S_2c_k \rightarrow S_3]$

$\curvearrowright b_jb_jb_jb_jb_jb_jb_ja_iS_0a_ia_ic_kc_kc_kc_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_0a_i \rightarrow S_1]$

$\curvearrowright b_jb_jb_jb_jS_1b_jb_ja_ia_ia_ic_kc_kc_kc_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_1b_j \rightarrow S_5]$

$\curvearrowright b_jb_jb_jb_jb_jb_ja_ia_ia_ic_kc_kS_5c_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_5c_k \rightarrow S_6]$

$\curvearrowright b_jb_jb_jb_jb_jb_ja_ia_ia_ic_kc_kc_kc_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_iS_0$ $[S_0a_i \rightarrow S_8]$

$\curvearrowright b_jb_jb_jb_jb_jb_ja_ia_ia_ic_kc_kc_kc_kc_kc_kb_jS_8b_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_i$ $[S_8b_i \rightarrow S_9]$

$\curvearrowright b_jb_jb_jb_jb_jb_ja_ia_ia_ic_kS_9c_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_9c_k \rightarrow S_{10}]$

$\curvearrowright b_jb_jb_jb_jb_jS_{10}a_ia_ia_ic_kc_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_{10}b_j \rightarrow S_{11}]$

$\curvearrowright b_jb_jb_jb_jb_ja_ia_ia_ic_kc_kc_kS_{11}b_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_{11}c_k \rightarrow S_{12}]$

$\curvearrowright b_jb_jb_jb_jb_ja_iS_0a_ic_kc_kc_kb_jb_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_0a_i \rightarrow S_{14}]$

$\curvearrowright b_jb_jb_jb_jb_ja_ia_ia_ic_kc_kc_kS_{14}b_jb_jb_jb_jc_kc_kc_kc_kc_kc_ka_ia_ia_ia_i$ $[S_{14}b_j \rightarrow S_{15}]$

$\curvearrowright b_jb_jb_jb_jb_ja_ia_ia_ic_kc_kc_kb_jb_jb_jb_jc_kc_kc_kc_kS_{15}c_ka_ia_ia_ia_i$ $[S_{15}c_k \rightarrow S_{16}]$

$\curvearrowright b_jb_jb_jb_jb_ja_ia_ia_ic_kc_kc_kb_jb_jb_jb_jc_kc_kS_{15}c_ka_ic_kc_ka_ia_ia_i$ $[S_{15}c_k \rightarrow S_{16}]$

$\curvearrowright b_jb_jb_jb_jb_ja_iS_{16}a_ic_kc_kc_kb_jb_jb_jb_jc_kc_kc_ka_ia_ic_kc_ka_ia_i$ $[S_{16}a_i \rightarrow S_{17}]$

$\curvearrowright b_j b_j b_j S_{17} b_j a_i a_i c_k c_k c_k b_j b_j b_j b_j c_k c_k c_k a_i a_i c_k c_k a_i$ $[S_{17} b_j \rightarrow S_{18}]$

$\curvearrowright b_j b_j b_j b_j a_i a_i c_k c_k c_k b_j b_j b_j b_j c_k c_k S_{18} a_i a_i c_k c_k a_i$ $[S_{18} c_k \rightarrow S_{19}]$

$\curvearrowright b_j b_j b_j b_j a_i a_i c_k c_k c_k b_j b_j b_j b_j c_k c_k S_{16} a_i c_k c_k a_i$ $[S_{16} c_k \rightarrow S_{20}]$

$\curvearrowright b_j S_{20} b_j b_j a_i a_i c_k c_k c_k b_j b_j b_j b_j c_k c_k a_i c_k c_k a_i$ $[S_{20} b_j \rightarrow S_{21}]$

$\curvearrowright b_j b_j b_j a_i a_i c_k c_k S_{21} b_j b_j b_j b_j c_k c_k a_i c_k c_k a_i$ $[S_{21} c_k \rightarrow S_{22}]$

$\curvearrowright b_j b_j b_j a_i a_i c_k c_k b_j b_j b_j S_{22} c_k c_k a_i c_k c_k a_i$ $[S_{22} b_j \rightarrow S_{23}]$

$\curvearrowright b_j b_j b_j a_i a_i c_k c_k b_j b_j b_j c_k S_{23} a_i c_k c_k a_i$ $[S_{23} c_k \rightarrow S_{24}]$

$\curvearrowright b_j b_j b_j a_i S_0 c_k c_k b_j b_j b_j c_k a_i c_k c_k a_i$ $[S_0 c_k \rightarrow S_{25}]$

$\curvearrowright b_j S_{25} b_j a_i c_k c_k b_j b_j b_j c_k a_i c_k c_k a_i$ $[S_{25} b_j \rightarrow S_{26}]$

$\curvearrowright b_j b_j a_i c_k c_k b_j b_j b_j S_{26} a_i c_k c_k a_i$ $[S_{26} c_k \rightarrow S_{27}]$

$\curvearrowright b_j b_j a_i c_k c_k b_j b_j S_{27} a_i c_k c_k a_i$ $[S_{27} b_j \rightarrow S_{28}]$

$\curvearrowright b_j b_j S_{16} c_k c_k b_j b_j a_i c_k c_k a_i$ $[S_{16} a_i \rightarrow S_{29}]$

$\curvearrowright b_j b_j c_k c_k S_{29} b_j a_i c_k c_k a_i$ $[S_{29} b_j \rightarrow S_{30}]$

$\curvearrowright b_j b_j c_k S_{30} b_j a_i c_k c_k a_i$ $[S_{30} c_k \rightarrow S_{31}]$

$\curvearrowright b_j S_{31} c_k b_j a_i c_k c_k a_i$ $[S_{31} b_i \rightarrow S_{32}]$

$\curvearrowright b_j S_{32} b_j a_i c_k c_k a_i$ $[S_{32} c_k \rightarrow S_{33}]$

$\curvearrowright b_j b_j a_i c_k c_k S_0$ $[S_0 a_i \rightarrow S_{34}]$

$\curvearrowright S_{34} b_j a_i c_k c_k$ $[S_{34} b_j \rightarrow S_{35}]$

$\curvearrowright b_j S_0 c_k c_k$ $[S_{16} a_i \rightarrow S_{36}]$

$\curvearrowright b_j S_{36} c_k$ $[S_{36} a_i \rightarrow S_{37}]$

$\curvearrowright S_{37} b_j$ $[S_{37} a_i \rightarrow S_{38}]$

$\curvearrowright S_{38}$ $[S_{38} c_k \rightarrow S_{39}]$

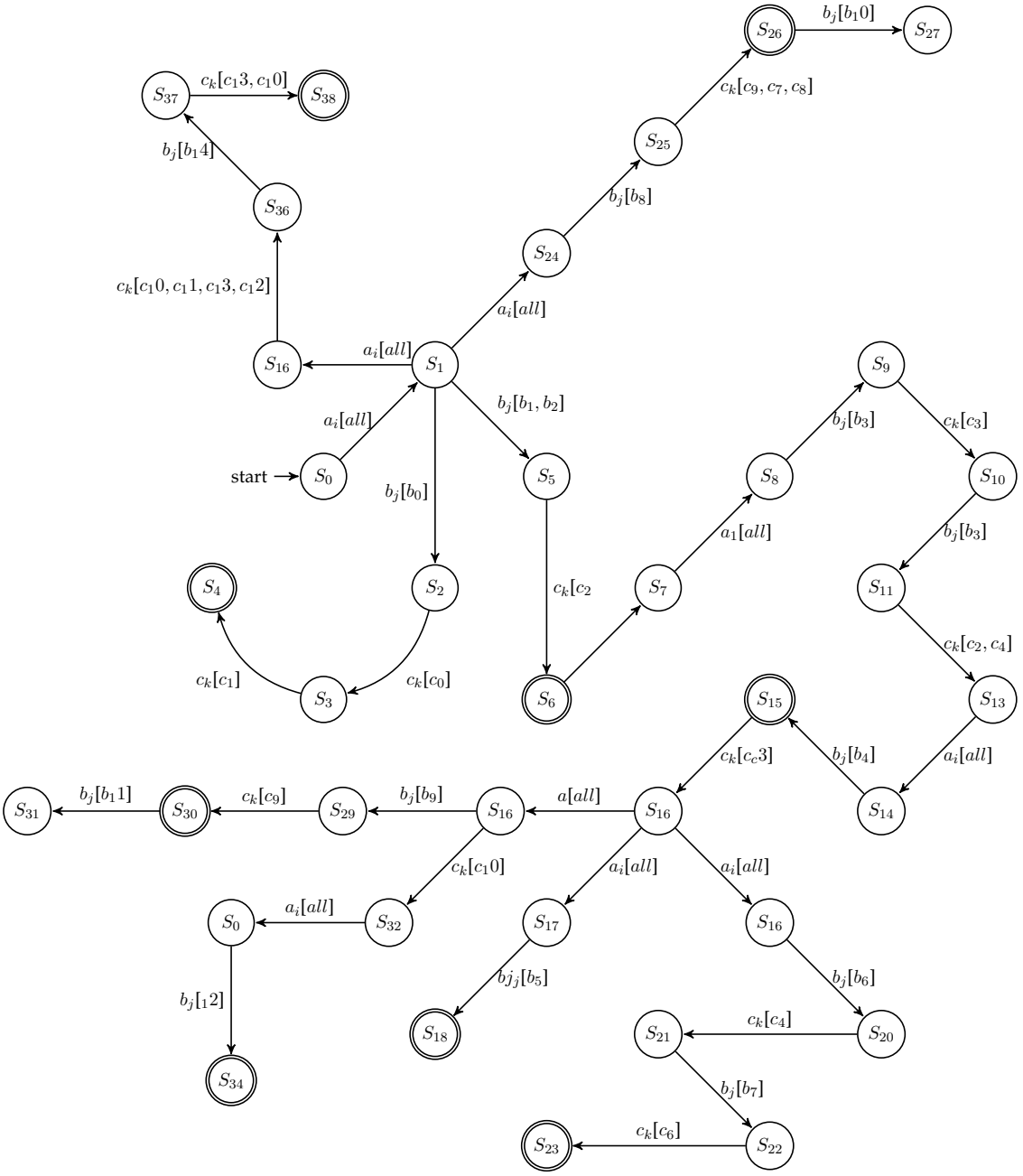


FIGURE 3.5: Complete Top Mail Queries

iii. **How data is represented**

In this work, with the aid of the categorisation process in the database chosen, data is denoted as Σ (Alphabets). This is because in the JFA, every input string constitutes of a Σ. Eg: $\{a_0, b_0, c_0\}$

3.3.2 Generating Feedback from FAQs using JFA

With the above conditions in the Abstract [2] of this research work, the customer queries are mapped to all possible JFAs, and a list of files ranked by relevance to the query posed by the customer is returned and displayed.

The following statement below is addressed, to have a clear understanding of how JFA is used to automatically generate feedback to queries:

i. **How the answers are matched with the questions in the FAQs file using JFA**

The users' question, is treated as a query to be matched against the library of the FAQ files. Lets bear in mind that in the theory of computation, the simpler abstract machine is Finite Automata. With the corpus of FAQs already extracted, when a query is given to the system, the JFA narrows the search to a single FAQ file using its discontinuous pattern of string reading, which is likely to contain an answer to the users' query (this is because our FAQ data is saved in packets/categories). With our data denoted by Σ (knowing that the alphabets contributes to our Σ) the choice of file will automatically be confirmed by the JFA, thereby matching each Q/A pair against the users' question to find the ones that best match it.

In Figure 3.6, When a customer poses a query in natural language to the chat-bot, it is processed through the parser1 to the JFAs, which maps the query to the possible FAQ in the FAQs database and returns to the JFAs through parser2. If match is found, answer will be displayed and if match is not found, an error message will be displayed.

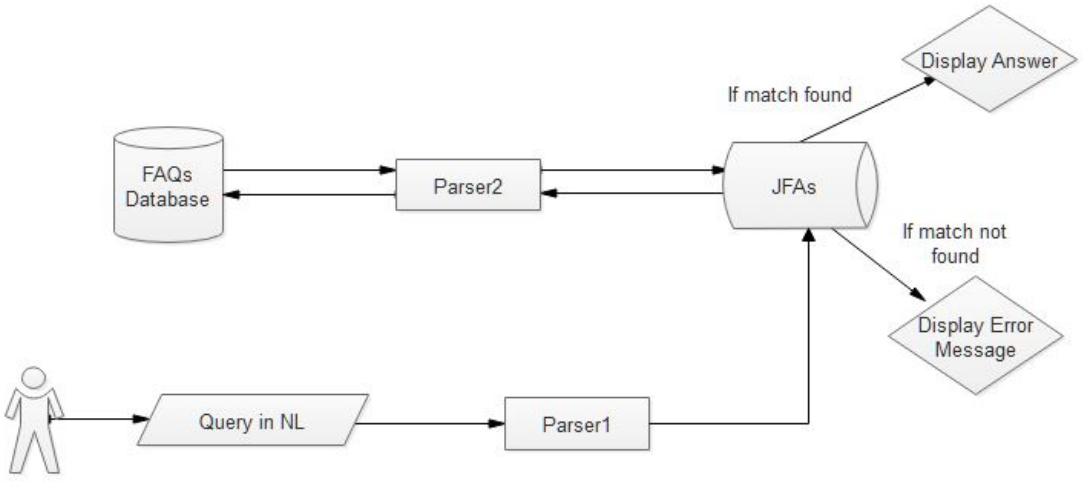


FIGURE 3.6: Match Process

3.4 Text Normalization

In this research work, Semantic Normalization of data was done by manually providing available synonyms to all possible JFAs in the training data set. Example of this approach is shown below.

- **delete;** erase, remove, close, clean, discard
- **archiving;** saving, , filing, chronicling, storing,
- **cataloguing;** recording, registering, documenting, cache
- **stay;** remain, lodge, logded, wait, keep
- **archived;** filed, saved, chronicled, stored, recorded, catalogued, ducumented
- **deleted;** erased, removed, closed, cleaned, discarded
- **archive;** save, file, chronicle, store, catalogue, record, register, document, cache
- **reply;** answer, respond, return
- **forward;** send, dispatch, repost
- **spell-check;** spell-check
- **write;** compose, draft, create, compile
- **have;** posses, comprise, contain, include,incorporate
- **share;** part, allow, allocate
- **support;** aid, assist, help
- **add;** include, attach
- **limitation;** limit, confinement, restriction, control, barrier, impediment, constraint
- **drag;** haul, relocate, pick, take
- **drop;** keep, leave
- **attach;** connect, affix, link
- **copy;** duplicate, move, transmit

- **organize;** put together, sort, put in order, arrange, catalogue, structure, assemble
- **create;** generate, produce, make, design, originate, develope, form
- **apply;** relate, input
- **nest;** cluster, set, put together, form
- **deleting;** erasing, removing, closing, cleaning, discarding
- **find;** locate, spot, get, obtain, secure
- **remove;** take out, delete, erase, discard
- **have access;** explore, use, entry
- **view;** see, peruse
- **log in;** sign in, access, get in
- **forgot;** do not remember, cannot recall, cannot remember, lost
- **do;** do
- **change;** alter, amend, modify
- **remember;** recall, memorize, store, save, recollect
- **sign out;** log out, leave, get out
- **remain;** stay, last, keep
- **prevent;** stop, keep from, avert, block, shut

WordNet, was also integrated to further strengthen the semantics structure in this research work. The organization of WordNet through lexical significance, instead of using lexemes makes it different from the traditional dictionaries and thesaurus [Miller, 1990]. WordNet was developed at the University of Princeton, and is a thesaurus for the English language based on psycholinguistics studies [Miller, 1995]. It covers lexico-semantic categories called synsets and was conceived as a data-processing resource. The synsets are sets of synonyms, which gather lexical items having similar significance, for example the words “support” and “aid” is grouped in the synset {support, aid}.

3.5 Results

The Chat-bot was tested with our training queries retrieved from Wits CNS FAQs web-page [1]. These FAQs contained 50 queries in eight topics, see [Appendix A]. The queries were mapped to all possible training JFAs and all 50 queries were recognized. Refer to Appendix B.

Further more two queries out of each of the eight topics in the 50 FAQs were rephrased and posed to the chat-bot as seen in Table (4.1) to Table (4.8) respectively. The Chat-bot was additionally tested on a large synthetic and real data set (real-time queries) obtained from the Wits CNS information desk office. The files obtained from Wits CNS information desk office were *1,000* in all and were in taxonomies such as; fees queries, admission status queries, access key cards queries, email accounts queries, etc. A data filtering process was performed on the *1,000* documents and *192* queries that were used to perform the test and *84* queries were recognised.

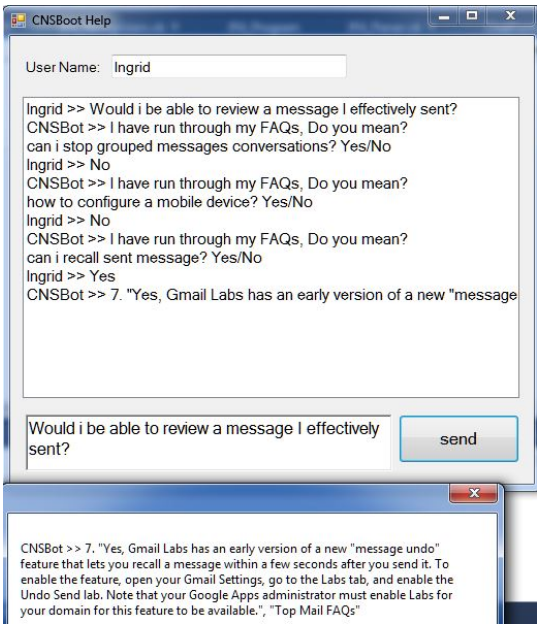
Some of the test results can be seen in Figure 3.7. Where ID 81, queried about student email account and how it could be setup, a device was mentioned, access to email was mentioned and also student number was provided. This system mapped the query to all possible JFAs as seen in the same Figure 3.7, and the query was recognized. Same as ID 86, 89 and 91 etc.

[1] https://sites.google.com/a/wits.ac.za/wits-google-apps-faq/faq/mail

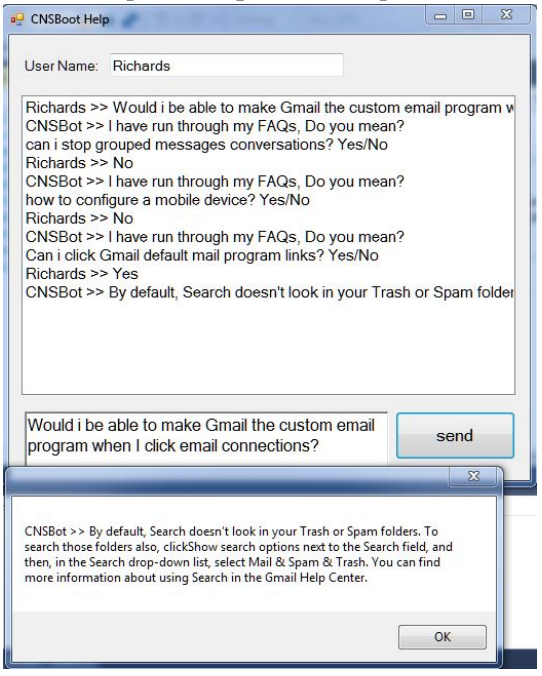
ID	Email Body	Possible JFAs
81	I am having difficulty accessing my student email. I have had to format my laptop and lost all my settings. My student number is 544870. Could you kindly assist with instructions on how to set up my email again.	Can I, stop, grouped, messages, conversations, **q5** how, have , run out, mail, messages, **q3** Can I, access, mail, offline, **q4** Can I, have, app, multiple, devices(Mobile App) no, folders, Gmail, How, organize, messages(Label) Does, Gmail, have, keyboard, shortcuts(Email Features)
86	I am a new student and would like to set up my email account. A progress report email was sent to me but I have not been able to access it. Please let me know the process for setting up student email.	Can I, stop, grouped, messages, conversations, **q5** how, to, configure, mobile, device, **q1** Can I, recall, sent, message, **q7** Can I, access, mail, offline, **q4** Can I, have, app, multiple, devices (Mobile App) Can I, apply, label, single, email, message(Label) Can I, share, email, another, employee(Email Features)
89	May you please assist me in activating my email account, my student number is 307038 and ID number is 900208 0841 080. I graduated in 2011 and I would like to apply again this year.	Can I, stop, grouped, messages, conversations, **q5** Can I, access, mail, offline, **q4** how, to, configure, mobile, device, **q1** Can I, apply, one, label, single, email, message(Label)
91	Could you help me please? I'm not able to access my emails on MyWits>StudentEmail. The Email login link takes me to Google Mail.	Can I, stop, grouped, messages, conversations, **q5** how, to, configure, mobile, device, **q1** how, have, run out, mail, messages, **q3** What, do, forgot, login, details (Mobile App) Can I, apply, label, single, email, message(Label) Can I, share, email, another, employee(Email Features)

FIGURE 3.7: Real time queries mapped to JFAs

Referencing the data set in Section 4.2, there are eight topics in the 50 FAQs we have as training data set. Two queries out of each of the eight topics in the 50 FAQs were rephrased and posed to the chat-bot, few results of the tests are shown in Figure 3.8a and Figure 3.8b respectively.



(A) 1st Rephrased queries in Top Mail FAQs



(B) Second Rephrased queries in Top Mail FAQs

3.6 Conclusion

This chapter described the method for this research. The design of this work was presented, and implementation to creating a friendly usable question answering system. Output results are also shown.

Chapter 4

Evaluation

4.1 Introduction

In the previous chapter, the design and implementation were discussed. This chapter is focused on the evaluation thereof. Section 4.2 introduces the data set used for our experiment. Section 4.3 is focused on the ground truth data set taken on the system. Section 4.4 is focused on evaluating the system's metrics. Section 4.5 discusses the performance metrics of the system. Section 4.6 concludes the chapter.

4.2 Data Set

The training data set of 50 FAQs in Appendix A and their related answers extracted from Wits CNS FAQ page, used to test the chat-bot was good enough to analyse system performance. As mentioned in the previous chapter, two queries out of each of the eight topics in the 50 FAQs were rephrased and posed to the chat-bot as seen in Table (4.1) to Table (4.8) respectively. The Chat-bot was also tested on a large synthetic and real data set (real-time queries) of *192* queries obtained from the Wits CNS information desk office.

Archiving and Deleting Messages	
x_1: How long do messages stay in the Trash?	y_1: To what extent do messages remain in the bin folder?
x_2: When should I delete a message vs. archiving it?	y_2: At the point when should I erase a communication versus chronicling it?

TABLE 4.1: Archiving and Deleting Messages with rephrased queries

Top Mail FAQs	
x_3: Can I make Gmail the default email program when I click email links?	y_3: Would i be able to make Gmail the custom email program when I click email connections?
x_4: Can I recall a message I already sent?	y_4: Would i be able to review a message I effectively sent?

TABLE 4.2: Top Mail FAQs with rephrased queries

Spam FAQs	
x_5: How long do messages remain in my Spam folder?	y_5 To what extent do messages stay in my Spam envelope?
x_6: How do I prevent messages from specific senders from being tagged as spam?	y_6: How would I keep messages from particular senders from being labeled as spam?

TABLE 4.3: Spam FAQs with rephrased queries

Mobile App FAQs	
x_7: I forgot my login details. What do I do?	y_7: I do not remember my login passwords. please What do I do?
x_8: Can another person log into the app on my device with their login details?	y_8: Will someone else be able to sign into the application on my gadget with their login passwords?

TABLE 4.4: Mobile App FAQs with rephrased queries

Labels FAQs	
x_9: There are no folders in Gmail. How do I organize my messages?	y_9: There are no organizers in Gmail. How would I sort out my messages?
x_{10}: Can I apply more than one label to a single email message?	y_{10}: Would i have the capacity to apply more than one mark to a solitary email message?

TABLE 4.5: Labels FAQs with rephrased queries

File Attachment FAQs	
x_{11}: Is there a size or type limitation for file attachments in Gmail?	y_{11}: Is there a size or sort confinement for record connections in Gmail?
x_{12}: Can I drag and drop a file to attach it to a message?	y_{12}: Would i be able to relocate a document to connect it to a message?

TABLE 4.6: File Attachment FAQs with rephrased queries

Email Features FAQs	
x_{13}: Can I share my email with another employee?	y_{13}: Would i be able to impart my email to another personnel?
x_{14}: Does Gmail have a "tasks" feature that lets me add messages to a list for follow-up?	y_{14}: Does Gmail have an "undertakings" highlight that gives me a chance to include messages to a rundown for development?

TABLE 4.7: Email Features FAQs with rephrased queries

Conversations and Messages FAQs	
x_{15}: Can I reply to or forward just a single message in a conversation?	y_{15}: Would i be able to answer to or repost only a solitary message in a discussion?
x_{16}: How can I spell-check a message I write?	y_{16}: How might I spell-check a message I compose?

TABLE 4.8: Conversations and Messages FAQs with rephrased queries

4.3 Ground Truth Data

The key to a successful question and answer system is to obtain a ground truth. In this project, the collection of ground-truth data enabled calibration of FAQs data, and aided in its interpretation and analysis. Each query in the FAQs, contained three *entities*; Query, Action and Object type denoted in a_i, b_j, c_k accordingly.
These were the labels used to test this work. The *entities* makes up the JFA symbols and with these symbols, queries posed to the chat bot are mapped to the FAQs and feedback is generated if match is found in the Q/A pairs.
To further establish the ground truth for this research work, real-time query assessment was chosen, to test the performance of the chat-bot built.
There were quite a number of queries posed by staff and students in the synthetic and real data set collected from the CNS help desk. These queries were tested on the chat-bot, and relevant documents related to most queries were retrieved. Some were not recognised though, as their queries were out of context of our training data set refer to Figure 4.1a and Figure 4.1b.

4.4 Evaluation Metrics

Considering that there is typically one right answer to be retrieved from each FAQ, these are not independent measures of performance. Our evaluation is based on document relevance and not ranking i.e. **relevance to queries given**.

Evaluation metrics explain the performance of a model. An important aspect of evaluation metrics is their capability to discriminate among model results. The motive of this work, is not to build a predictive model, but to create a model which gives high accuracy on out of sample data. Hence, it is crucial to check accuracy of this model prior to computing predicted values.
The term accuracy, can be defined in two different ways[1].

1. How close a measurement is to the true value, is referred to as accuracy in Math, Science and Engineering.

2. While the ISO (International Organization for Standardization) applies a more rigid definition, where accuracy refers to a measurement with both true and consistent results. In this definition, an accurate measurement has no systematic error and no random error. Basically, the ISO advises that the term "accurate" should be used when a measurement is both accurate and precise.

Accuracy Metrics; There are different ways to calculate accuracy metrics, but this work is interested in User's Accuracy.
The *User's Accuracy* is the accuracy from the point of view of a system user, not the system maker. The User's Accuracy essentially in this research work tells us how often relevant documents are retrieved. This is referred to as reliability[2].

Hence, the classification accuracy Ai of an individual program i depends on the number of samples correctly classified *(true positives , true negatives)* and is evaluated by the formula;

$$Ai \quad = \frac{t}{n} \times 100\%.$$

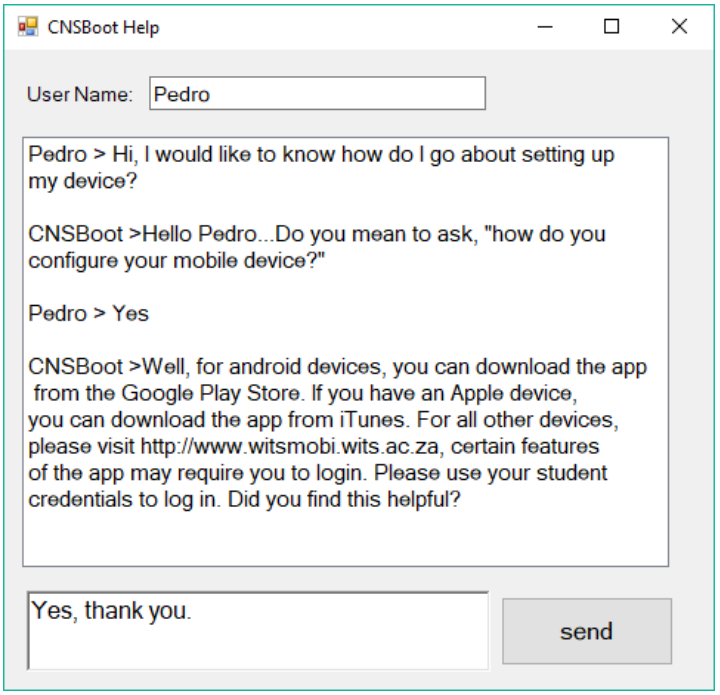
where t is the number of sample cases correctly classified, and n is the total number of sample cases.

4.5 System Performance Metrics

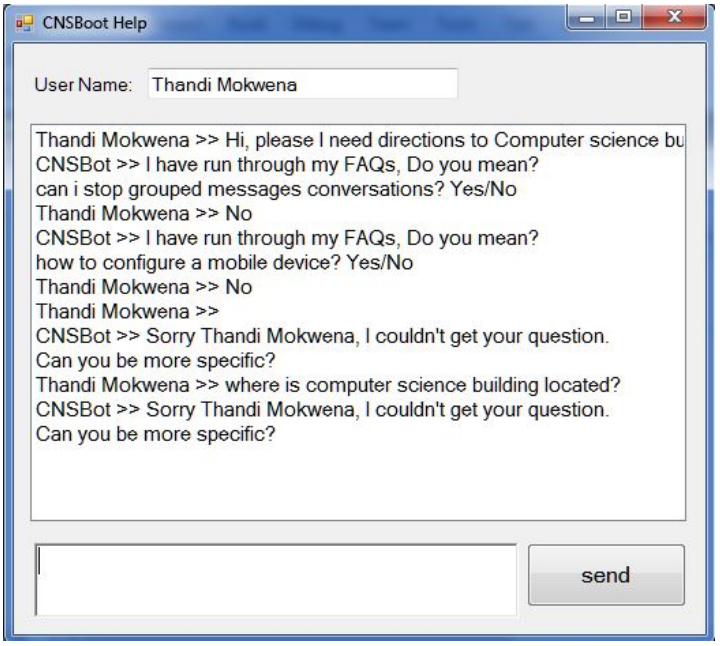
A software application called CNSBot, has been developed. CNSBot takes a customer query and attempts to map it to an FAQ in order to provide feedback.
An evaluation was carried out on the performance of the CNSBot with the 50 training FAQs obtained from Wits CNS [3] , they were mapped to all possible JFAs and we had 98% conversion rate. Refer to Appendix B. Furthermore, few individuals tested the chatbot with their queries as shown in Figure 4.1a, where query was recognised and

[1] https://www.thoughtco.com/difference-between-accuracy-and-precision-609328
[2] http://gsp.humboldt.edu/olm_2016/Courses/GSP_216_Online/lesson6-2/metrics.html
[3] https://sites.google.com/a/wits.ac.za/wits-google-apps-faq/faq/mail

Figure 3.8b, where query was not recognised because query was out of our training data context.



(A) Query recognised by CNSBot



(B) Query not recognised by CNSBot

The standard way to deal with information retrieval system evaluation, circles around the thought of relevant and non-relevant documents [Manning et al., 2008]. This research work, is interested in retrieving relevant documents related to queries posed by customers.

This research work assumes right answers exist (a single QA pair that best address the users' question as it was posed). The CNSbot is tasked to return answers within the small fixed-size set of results that can be displayed.
Referencing Section 4.2, where the chat-bot is tested with two synthetically generated queries each from the eight topics of the FAQs, the system performance metrics of this work is show below.

Training Data (x)	
Q	A
x_1	$x_{1,2}$
x_2	$x_{2,2}$
x_3	$x_{3,2}$
x_4	$x_{4,2}$
x_5	$x_{5,2}$
x_6	$x_{6,2}$
x_7	$x_{7,2}$
x_8	$x_{8,2}$
x_9	$x_{9,2}$
x_{10}	$x_{10,2}$
x_{11}	$x_{11,2}$
x_{12}	$x_{12,2}$
x_{13}	$x_{13,2}$
x_{14}	$x_{14,2}$
x_{15}	$x_{15,2}$
x_{16}	$x_{16,2}$

TABLE 4.9: Original Queries

Test Data (y)	
Q	A
y_1	$y_{1,2}$
y_2	$y_{2,2}$
y_3	$y_{3,2}$
y_4	$y_{4,2}$
y_5	$y_{5,2}$
y_6	$y_{6,2}$
y_7	$y_{7,2}$
y_8	$y_{8,2}$
y_9	$y_{9,2}$
y_{10}	$y_{10,2}$
y_{11}	$y_{11,2}$
y_{12}	$y_{12,2}$
y_{13}	$y_{13,2}$
y_{14}	$y_{14,2}$
y_{15}	$y_{15,2}$
y_{16}	$y_{16,2}$

TABLE 4.10: Synthetically Generated Queries

Considering that Table (4.9), consists of two queries each from the eight topics of our training JFAs (x), the recognition rate was *accurate* after they were posed to the chat-bot. Meanwhile Table (4.10), contains the synthetically generated queries two each from the eight topics in the FAQs file which made up our test data set (y). A test was carried out with the synthetically generated queries, and out of the sixteen queries thirteen were recognized correctly by the chat-bot, and three were not recognised because semantically, the words that replaced our JFAs' in the synthetically generated queries were not found in our synonyms data set.

Where

Training data (x) are denoted as;

$x_1 \dots x_{16}$ = Queries

$x_{1,2} \dots x_{16,2}$ = Answers

And

Test data (y) denoted as;

$y_1 \dots y_{16}$ = Queries

$y_{1,2} \dots y_{16,2}$ = Answers

Here

$x_1 = y_1$ — $x_{2,2} = y_{2,2}$

$x_2 = y_2$ — $x_{2,2} = y_{2,2}$

$\vdots$

$x_{16} = y_{16}$ — $x_{16,2} = y_{16,2}$

For training data (x), recognition rate = accurate. Refer to Appendix B

While

In the test data set (y), we have

true positive = 13

true negative = 3

Total number of test queries = 16

Therefore;

For our test data (y), the classification accuracy is

$$Ai = \frac{13}{16} \times 100 = 81.25\%$$

For the large synthetic and real data set (real-time queries) of *192* queries obtained from the Wits CNS information desk office refer to Subsection 3.5, we have;

true positive = 84

true negative = 108

Total number of synthetic queries = 192

The classification accuracy is therefore;

$$Ai = \frac{84}{192} \times 100 = 43.75\%$$

Reason for the decrease in accuracy percentage on the large synthetic and real data set (real-time queries) test, is because most of the queries were out of our JFA training data context. *This can be improved in future work by adding more semantic rules to the chat-bot.*

4.6 Conclusion

This chapter presented the evaluation of this research work. It started by giving precised details of the data set used to achieve the results in this work, the ground truth data briefly explained, evaluation metrics discussed alongside system performance metrics with tests and the results shown.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This research work, is a Jumping Finite Automata (JFA) web-accessible knowledge based information retrieval system, that relied on the understanding of collection of Frequently Asked Questions (FAQs) in natural language. As explained in the abstract of this work, that question answering can be reduced to matching new questions against QA pairs when there is an existing collection of Question Answer (QA) pairs. The system combined statistical measures and shallow lexical semantics to match users' questions against QA pairs from FAQ files. Our evaluation, conducted with 50 questions from the Wits CNS FAQ page on her website, demonstrated prospect for the work.

The control of our approach is from the fact that we are using JFA computing module to model natural language IR problems. Referencing our second research question, this system doesn't need to actually comprehend the queries received or generate new text that explains the answer, it only had to identify the files that were relevant to the query posed i.e. mapped the customer queries to all possible JFAs, and then matched against the segments of text that were used to organize the files themselves.

A chat-bot was built to test how much queries that could be recognised, and the 50 QA pairs from the Wits CNS FAQ page were recognised because they were properly trained.

5.2 Limitations of the Study

One major limitation of this research, was that we could not test all the queries from the large scale of data obtained from CNS help desk, which contained real-time queries

from both staff and students with different topics. Most of the documents were redundant so we only considered the topics related to our training data set. More queries will be trained in future work.

5.3 Contributions and Future Work

This work has contributed a new extension to IR problems, in NLP, that involves the abstraction of all FAQs to a JFA and applying algorithms to map customer queries to the underlying JFA of all possible queries.

Other contributions of this work are;

Reduced search-time for customers in obtaining desired answers to their queries from FAQs files.

Real-time computation of QA in NLP i.e. customers are allowed to ask their questions in natural language, and the meaning of their input text extracted, then automatically feedback is provided from a pool of FAQs file on the go.

There are still unexplored areas in NLP domain to build Q/A systems. Given the results obtained from the experiments we carried out using JFA (an abstract computing machine — in performing this IR task), it will be interesting to explore more NLP techniques to further implement this work, such as using statistical NLP.

Train additional data and add extra semantic rules to the chat-bot, to enable more recognition of possible queries in FAQs data set.

Bibliography

[Ade-Ibijola, 2016] Abejide Ade-Ibijola. FINCHAN: A grammar-based tool for automatic comprehension of financial instant messages. In *Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists*, pages 1–10. ACM, 2016.

[Aho and Johnson, 1974] Alfred V. Aho and Stephen C. Johnson. LR parsing. *ACM Computing Surveys (CSUR)*, 6(2):99–124, 1974.

[Aström and Murray, 2010] Karl Johan Aström and Richard M Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.

[Bekhti et al., 2011] Smain Bekhti, Amjad Rehman, Maryam Al-Harbi, and Tanzila Saba. Aquasys: An arabic question-answering system based on extensive question analysis and answer relevance scoring. *International Journal of Academic Research*, 3(4), 2011.

[Bordignon, 2016] Ana Claudia de Almeida Bordignon. A systematic literature review on natural language processing in business process identification and modeling. 2016.

[Buckley et al., 1993] C Buckley, G Salton, and J Allan. The smart information retrieval project. In *Proceedings of the workshop on Human Language Technology*, pages 392–392. Association for Computational Linguistics, 1993.

[Buckley, 1985] Chris Buckley. Implementation of the smart information retrieval system. Technical report, Cornell University, 1985.

[Burke et al., 1997] Robin D Burke, Kristian J Hammond, Vladimir Kulyukin, Steven L Lytinen, Noriko Tomuro, and Scott Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. *AI magazine*, 18(2):57, 1997.

[Büttcher et al., 2016] Stefan Büttcher, Charles LA Clarke, and Gordon V Cormack. *Information retrieval: Implementing and evaluating search engines*. MIT Press, 2016.

[Cairns et al., 2011] Brian L Cairns, Rodney D Nielsen, James J Masanz, James H Martin, Martha S Palmer, Wayne H Ward, and Guergana K Savova. The mipacq clinical question answering system. In *AMIA Annu Symp Proc*, volume 2011, pages 171–180, 2011.

[Chatterjee et al., 2015] Shubham Chatterjee, Kasturi Paul, Reek Roy, and Asoke Nath. A pilot study on natural language processing–applications of finite state automation. 2015.

[Chomsky, 1956] Noam Chomsky. Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124, 1956.

[Chowdhury, 2003] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[Christopher et al., 2008] D Manning Christopher, Raghavan Prabhakar, and SCHuTZE Hinrich. Introduction to information retrieval. *An Introduction To Information Retrieval*, 151:177, 2008.

[Cormen et al., 2001] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. Introduction to algorithms second edition, 2001.

[D'angelo et al., 2013] Adam D'angelo, Charles Duplain Cheever, Kevin G Der, and Rebekah Marie Cox. Methods and systems for soliciting an answer to a question, August 20 2013. US Patent 8,516,379.

[Di Fabbrizio et al., 2014] Giuseppe Di Fabbrizio, Dawn L Dutton, Narendra K Gupta, Barbara B Hollister, Mazin G Rahim, Giuseppe Riccardi, Robert Elias Schapire, and Juergen Schroeter. Method of handling frequently asked questions in a natural language dialog service, February 4 2014. US Patent 8,645,122.

[Dretske, 1981] Fred Dretske. Knowledge and the flow of information. 1981.

[Earley, 1970] Jay Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 13(2):94–102, 1970.

[Eggebraaten et al., 2014] Thomas J Eggebraaten, Richard J Stevens, and Eric W Will. Natural language processing (NLP), January 28 2014. US Patent 8,639,497.

[Ezhilarasu et al., 2015] P Ezhilarasu, N Krishnaraj, and Suresh V Babu. Applications of finite automata in text search-a review. *International Journal of Science, Engineering and Computer Technology*, 5(5):116, 2015.

[Fan et al., 2012] James Fan, Aditya Kalyanpur, David C Gondek, and David A Ferrucci. Automatic knowledge extraction from documents. *IBM Journal of Research and Development*, 56(3.4):5–1, 2012.

[Fernau et al., 2017] Henning Fernau, Meenakshi Paramasivan, Markus L Schmid, and Vojtěch Vorel. Characterization and complexity results on jumping finite automata. *Theoretical Computer Science*, 679:31–52, 2017.

[Fox, 1983] Edward A Fox. Some considerations for implementing the smart information retrieval system under unix. Technical report, Cornell University, 1983.

[Green Jr et al., 1961] Bert F Green Jr, 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*, pages 219–224. ACM, 1961.

[Greengrass, 2000] Ed Greengrass. Information retrieval: A survey. 2000.

[Hambir and Srivastav, 2012] Nitin Hambir and Ambrish Srivastav. Hindi parser-based on cky algorithm. *Int. J. Computer Technology & Applications*, pages 851–853, 2012.

[Harrison, 1978] Michael A Harrison. *Introduction to formal language theory*. Addison-Wesley Longman Publishing Co., Inc., 1978.

[Hirschman and Gaizauskas, 2001] Lynette Hirschman and Robert Gaizauskas. Natural language question answering: the view from here. *natural language engineering*, 7(04): 275–300, 2001.

[Hofs et al., 2011] Dennis Hofs, Boris van Schooten, and Rieks op den Akker. The imix demonstrator: An information search assistant for the medical domain. In *Interactive Multi-modal Question-Answering*, pages 11–21. Springer, 2011.

[Hopcroft and Ullman, 1969] John E Hopcroft and Jeffrey D Ullman. Formal languages and their relation to automata. 1969.

[Jackson and Moulinier, 2007] Peter Jackson and Isabelle Moulinier. *Natural language processing for online applications: Text retrieval, extraction and categorization*, volume 5. John Benjamins Publishing, 2007.

[Jacobs, 2014] Paul S Jacobs. *Text-based intelligent systems: Current research and practice in information extraction and retrieval*. Psychology Press, 2014.

[Jijkoun and de Rijke, 2005] Valentin Jijkoun and Maarten de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 76–83. ACM, 2005.

[Jones, 1999] Karen Sparck Jones. What is the role of nlp in text retrieval? In *Natural language information retrieval*, pages 1–24. Springer, 1999.

[Jurafsky and Martin, 2014] Dan Jurafsky and James H Martin. *Speech and language processing*, volume 3. Pearson London:, 2014.

[Khurana et al., 2017] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: State of the art, current trends and challenges. *arXiv preprint arXiv:1708.05148*, 2017.

[Kolomiyets and Moens, 2011] Oleksandr Kolomiyets and Marie-Francine Moens. A survey on question answering technology from an information retrieval perspective. *Information Sciences*, 181(24):5412–5434, 2011.

[Kramer, 2007] Jeff Kramer. Is abstraction the key to computing? *Communications of the ACM*, 50(4):36–42, 2007.

[Liddy, 2001] Elizabeth D Liddy. Natural language processing. 2001.

[Machinery, 1950] Computing Machinery. Computing machinery and intelligence-am turing. *Mind*, 59(236):433, 1950.

[Magnini et al., 2003] Bernardo Magnini, Simone Romagnoli, Alessandro Vallin, Jesús Herrera, Anselmo Penas, Víctor Peinado, Felisa Verdejo, and Maarten de Rijke. The multiple language question answering track at clef 2003. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 471–486. Springer, 2003.

[Malhotra, 2017] Pooja Malhotra. Question-answering system. 2017.

[Manning et al., 2008] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[Manning et al., 2014] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[Meduna and Zemek, 2012] Alexander Meduna and Petr Zemek. Jumping finite automata. *International Journal of Foundations of Computer Science*, 23(07):1555–1556, 2012.

[Meduna and Zemek, 2014a] Alexander Meduna and Petr Zemek. Jumping finite automata. In *Regulated Grammars and Automata*, pages 567–585. Springer, 2014a.

[Meduna and Zemek, 2014b] Alexander Meduna and Petr Zemek. *Regulated Grammars and Automata*. Springer, 2014b.

[Miller, 1990] George A Miller. Nouns in wordnet: a lexical inheritance system. *International journal of Lexicography*, 3(4):245–264, 1990.

[Miller, 1995] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[Mollá-Aliod and Vicedo, 2010] Diego Mollá-Aliod and José-Luis Vicedo. Question answering. In *Handbook of Natural Language Processing, Second Edition*, pages 485–510. Chapman and Hall/CRC, 2010.

[Parkes, 2008] Alan P Parkes. Elements of formal languages. In *A Concise Introduction to Languages and Machines*, pages 11–42. Springer, 2008.

[Quillian and Memory, 1968] MR Quillian and Semantic Memory. In m. minsky (ed.), semantic information processing, 1968.

[Raban and Harper, 2008] D Raban and F Harper. Motivations for answering questions online. *New media and innovative technologies*, 73, 2008.

[Reshamwala et al., 2013] Alpa Reshamwala, Dhirendra Mishra, and Prajakta Pawar. Review on natural language processing. *IRACST Engineering Science and Technology: An International Journal (ESTIJ)*, 3(1):113–116, 2013.

[Rice et al., 2001] Ronald E Rice, Maureen McCreadie, and Shan-Ju L Chang. *Accessing and browsing information and communication*. Mit Press, 2001.

[Sarle, 1995] Warren S Sarle. Measurement theory: Frequently asked questions. *Disseminations of the International Statistical Applications Institute*, 1(4):61–66, 1995.

[Schwalb, 2004] Robert Schwalb. *A humane education frequently asked questions document*. PhD thesis, Cambridge College Cambridge, Massachusetts, 2004.

[Sebastiani, 2002] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47, 2002.

[Simmons, 1965] Robert F Simmons. Answering english questions by computer: a survey. *Communications of the ACM*, 8(1):53–70, 1965.

[Singhal, 2001] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[Sosnin, 2012] Petr Sosnin. Question-answer approach to human-computer interaction in collaborative designing. *Chapter in the book "Cognitively Informed Intelligent Interfaces: Systems Design and Development" Published IGI Global*, pages 157–176, 2012.

[Srihari and Li, 2000] Rohini Srihari and Wei Li. A question answering system supported by information extraction. In *Proceedings of the sixth conference on Applied natural language processing*, pages 166–172. Association for Computational Linguistics, 2000.

[van Delden and Gomez, 2004] Sebastian van Delden and Fernando Gomez. Retrieving NASA problem reports: a case study in natural language information retrieval. *Data & Knowledge Engineering*, 48(2):231–246, 2004.

[Voorhees and Tice, 1999] Ellen M Voorhees and Dawn M Tice. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82, 1999.

[Walker et al., 2014] Andrew Walker, Andrew Starkey, Jeff Z Pan, and Advaith Siddharthan. Making test corpora for question answering more representative. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 1–6. Springer, 2014.

[Wang et al., 2013] Gang Wang, Konark Gill, Manish Mohanlal, Haitao Zheng, and Ben Y Zhao. Wisdom in the social crowd: an analysis of quora. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1341–1352. ACM, 2013.

[Wang et al., 2016] Xiaodong Wang, Bei Xu, and Hai Zhuge. Automatic question answering based on single document. In *Semantics, Knowledge and Grids (SKG), 2016 12th International Conference on*, pages 90–96. IEEE, 2016.

[Webber and Webb, 2010] Bonnie Webber and Nick Webb. Question answering. *The handbook of computational linguistics and natural language processing*, pages 630–654, 2010.

[Weller, 2007] Susan C Weller. Cultural consensus theory: Applications and frequently asked questions. *Field methods*, 19(4):339–368, 2007.

[Yao, 2014] Xuchen Yao. *Feature-driven Question Answering with Natural Language Alignment*. PhD thesis, 2014.

[Zhang and Clark, 2011] Yue Zhang and Stephen Clark. Shift-reduce ccg parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 683–692. Association for Computational Linguistics, 2011.

[Zou et al., 2014] Lei Zou, Ruizhe Huang, Haixun Wang, Jeffrey Xu Yu, Wenqiang He, and Dongyan Zhao. Natural language question answering over rdf: a graph data driven approach. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 313–324. ACM, 2014.

Appendix A

Ground Truth Work

Below, are FAQs extracted from The University of Witwatersrand Computer and Network Services (CNS) FAQs page and they are categorized into topics with their related answers for easy access to visitors.

- Archiving and Deleting Messages:

Question: When should I delete a message vs. archiving it?
Answer: "Deleting or archiving a message removes it from your inbox. If you delete a message, it's placed in the Trash and then permanently removed from your Google Apps account after 30 days. If you archive a message, it's moved to All Mail (your archive), where you can easily find it in the future, using Google's powerful search feature. Since you have plenty of space for storing all your mail (25 Gb), we recommend that you archive messages rather than permanently deleting them. "
Question: How long do messages stay in my archive?
Answer: "Messages remain in your archive forever, unless you choose to delete them."
Question: How long do messages stay in the Trash?
Answer: "Messages remain in the Trash for 30 days. After that, Gmail permanently deletes them. "
Question: How do I move a message out of the Trash?
Answer: "Find the message in the Trash and select it. Then, in the Move to drop-down list at the top of the Mail window, select Inbox."

Question: Why does a message I archived or deleted show up again in my Inbox?
Answer: "If someone replies to a message you archived or trashed, that message reappears in your Inbox as a conversation, bringing the message you archived or deleted, with it. If the message has already been permanently deleted from the Trash, you'll see an option at the bottom of the message to view the deleted messages in the conversation"
Question: Should I delete or archive message in my Sent folder?
Answer: "There's no need to delete or archive messages in the Sent folder. Messages remain in this folder forever, unless you delete them. But, because you have 25 GB of storage space, you can keep messages in this folder to refer to them later, if needed. Also, note that messages in the Sent folder are actually archived in All Mail, so even if you archive these messages, they stay in the Sent folder."

- Email features:

Question: Does Gmail have an Out of Office feature?
Answer: "Yes, in Gmail, you can set up your "vacation responder," which is similar to the Out of Office feature in Outlook or Lotus Notes. For details, see the Gmail Help Center.
Question: Can I share my email with another employee?
Answer: "Shared mailboxes aren't supported. However, you can easily set up an email filter (rule) to forward specific types of messages to another email."
Question: Does Gmail have keyboard shortcuts?
Answer: "Yes, Gmail includes a full set of keyboard shortcuts. First, you must enable keyboard shortcuts:

In the upper-right corner of the Mail window, click Settings.

Under Keyboard shortcuts, select Keyboard shortcuts on.

Click Save Settings. Then, to see the shortcuts, press SHIFT+? While viewing your list of messages in the main Mail window."

Question: Does Gmail support shared mailboxes?
Answer: " Not exactly. But as a workaround, you can create your own mailing list (called a "group") for all the employees who want to share an email address. This requires that your administrator has enabled User-Managed groups for your domain. If user-managed groups aren't available, then you should ask your Google Apps administrator to set up a mailing list (group) for all the employees who want to share an email address. Or if email delegation is enabled for your domain, you can use that to allow up to 10 other users access a single email account. "
Question: Why Does Gmail have a "tasks" feature that lets me add messages to a list for follow-up?
Answer: " "Yes, the Google Task gadget is available in Gmail and Calendar. For details, see Using the Task gadget."

- File Attachments:

Question: Is there a size or type limitation for file attachments in Gmail?
Answer: "Yes, in Gmail, you can set up your "vacation responder," which is similar to the Out of Office feature in Outlook or Lotus Notes. For details, see the Gmail Help Center.
Question: Can I drag and drop a file to attach it to a message?
Answer: "Yes, if you're using a Chrome browser. Otherwise, you must browse to a file to attach it."
Question: How can I copy a file attachment from one message to another?
Answer: "Because Gmail is a web-based system, you can't drag a file attachment from one message to another. As a workaround, you can do the following:

Open the message or conversation that contains the file attachment.

If the file is attached to a single message, click Forward (from the drop-menu at the top of the message).

If it's attached to a message in a conversation, click Forward all on the right. Delete all the "forwarded" content from original messages, which appears at the bottom of your new message. Note that the file attachment remains with the forwarded message.Then compose your new message and send it. Alternatively, you can download the attachment and then upload it to another message."

Question: Can I attach a message or conversation to a new message?
Answer: "No, you can't embed one message into another directly. As a workaround, you can do the following:

To attach a single message, open it and click Forward (from the drop-menu at the top of the message). To attach a conversation, open it click Forward all on the right.

Then compose your new message and send it. The earlier message will be included below your new message. Alternatively, you can copy the text from the earlier message and paste it into a new message"

- Labels:

Question: There are no folders in Gmail. How do I organize my messages?
Answer: "Instead of folders, Gmail has a "labels" feature. Labels are similar to folders, but are more powerful and flexible, because you can add multiple labels to a message to categorize it in several ways. For details, see the Gmail Help Center."
Question: How many labels can I create?
Answer: "You can create up to about 200 labels."
Question: Can I apply more than one label to a single email message?
Answer: "Yes, you can apply any number of labels to a message: Select the message in your Inbox, or open it, and select one or more labels in the Labels drop-down list at the top of your Mail window."
Question: Can I nest labels like I nested folders in old mail program?
Answer: "Yes. For details, see Create nested labels."
Question: Does deleting a label delete any messages that have that label?
Answer: "No. All it does is remove the label from the messages."

- Mobile App:

Question: Where can I find the WitsM app?
Answer: "For android devices, you can download the app from the Google Play Store. ",

If you have an Apple device, you can download the app from iTunes.

For all other devices, please visit witsmobi.wits.ac.za. Certain features of the app may require you to login. Please use your student credentials to log in."
Question: How do I remove the app from my device?
Answer: "To remove the app from the device, follow the standard procedure as per the device. Android – in settings, select applications, find WitsM and uninstall."

Question: Which features and resources do I have access to on the app?
Answer: "Bus Schedule - View the latest bus timetable

Comms – Receive important short messages from Wits University

Contacts – Find a list of important contact details

Exam Marks – View your mid-year and finals marks

Maps – Pinpoint and get directions to Wits buildings on the campus relative to your current location

Meals – Access the meal booking system for students who have a catered residence status

News – See an RSS feed of all the latest news being published by Wits University

Student Fees - Get a summary of your fees statements in a predefined date

This is Wits – Hear the stories that make Wits the great place it is today

Timetable – View your updated timetable of lectures with venues

Voice of Wits – Listen to your favourite VOW FM DJS live."
Question: How do I view my timetable?
Answer: "Click on the timetable icon. If you are already logged in it will display else it will prompt you to login."
Question: How do I book a meal?
Answer: " The meal feature works exactly the same as the Wits Dining system that can be found here."
Question: What can I see in Comms?
Answer: "The comms feature allows Wits to deliver urgent short messages to any individual who has the device loaded at this time."
Question: Which devices does the WitsM app support?
Answer: "The WitsM app will run on Andriod 4 and higher and any device that can open a mobi site."
Question: How do I log in to the WitsM App?
Answer: "From your device, launch the application by selecting the "WitsM" icon.On certain features you will be asked for your login details.For the timetable feature, use your Wits google credentials:studno@students.wits.ac.za and your password.For the meals feature use your stud number and password."
Question: I forgot my login details. What do I do?
Answer: "To reset your password, click here and select the Password self service option."
Question: Can I change the dashboard / homescreen?
Answer: "Not at this time."

Question: Can I have this app on multiple devices?

Answer: "The app can be loaded on any number of devices."

Question: Can another person log into the app on my device with their login details?

Answer: "As the devices stores the credentials, it is advisable for the owner of the device to use their credentials."

Question: Does the app remember login details?

Answer: "For use of the timetable feature, the login details are recorded. However, for the meals feature the student would need to log in each time."

Question: How do I sign out from the app?

Answer: "To sign out of the timetable feature, select Settings and then select Logout."

- Spam:

Question: How long do messages remain in my Spam folder?

Answer: "Messages remain in the Spam folder for 30 days. After that, Gmail permanently deletes them."

Question: How do I prevent messages from specific senders from being tagged as spam?

Answer: "If messages from a sender outside your domain are being incorrectly tagged as spam, you can prevent this from happening by creating an email filter using the Never send it to Spam option:

In Gmail, click Settings ¡ Filters ¿ Create a new filter. Enter the person's address in the "From field", and then click "Next Step". Select "Never send it to spam", and then click "Create Filter."

- Conversations and Messages:

Question: Can I reply to or forward just a single message in a conversation?

Answer: "Yes. Open the conversation and expand the individual message. From the drop-menu at the top-right of the message, click Reply or Forward."

Question: Can I delete a message from a conversation?

Answer: "Yes, you can delete one or more messages in a conversation as follows:

Open the conversation and expand the message you want to delete.

Open the drop-menu at the top-right of the message.

Select Delete this message."

Question: Can I open a message in a separate window from my main Mail window?

Answer: "Yes, if your browser is set to display pop-ups in a new window, you can do the following:

If you're reading a message, click the New Window icon in the upper-right corner of the message.

If you're composing a new message, click the New Window icon in the upper-right corner of the message."

Question: How can I spell-check a message I write?

Answer: "Click Check Spelling at the top of the message you're composing. Misspelled words are highlighted in yellow. Click a misspelled word to see suggested corrections."

Appendix B

Evaluation Results

"Archiving and Deleting Messages"	
Possible Query	**Possible JFA**
1. "When should I delete a message vs. archiving it?"	(when, delete, message, archiving)
2. "How long do messages stay in my archive?"	(How, messages, stay, archive)
3. "How long do messages stay in the Trash?"	(How, messages, stay, trash)
4. "How do I move a message out of the Trash?"	(How, move, message, out, trash)
5. "Why does a message I archived or deleted show up again in my Inbox?"	(Why, message, archived, deleted,show-up again, inbox
6. "Should I delete or archive message in my Sent folder?"	(Should I, delete, archive, message, sent folder)

FIGURE B.1: Test 1

"Conversations and Messages"	
Possible Query	**Possible JFA**
1. "Can I reply to or forward just a single message in a conversation?"	(Can I, reply, forward, single, message, conversation)
2. "Can I delete a messages from a conversation?"	(Can I, delete, messages, conversation)
3. "Can I open a message in a separate window from my main Mail window?"	(Can I, open, message, seperate window, main mail window)
4. "How can I spell-check a message I write?"	(How, spell-check, message, write)

FIGURE B.2: Test 2

"Email features"	
Possible Query	**Possible JFA**
1. "Does Gmail have an Out of Office feature?"	(Does, Gmail, out of, office, feature)
2. "Can I share my email with another employee?"	(Can I, share, email, another, employee)
3. "Does Gmail have keyboard shortcuts?"	(Does, Gmail, have, keyboard shortcuts)
4. "Does Gmail support shared mailboxes?"	(Does, Gmail, support, shared mailboxes)
5. "Does Gmail have a "tasks" feature that lets me add messages to a list for follow-up?"	(Does, Gmail, have, tasks, feature, add, messages, list, follow-up)
5. 6. "Does Gmail features involve copying and sharing attachments within user's contacts?"	(Does, Gmail, features, copying, sharing, attachments, user's, contact)

FIGURE B.3: Test 3

"File Attachments"	
Possible Query	**Possible JFA**
1. "Is there a size or type limitation for file attachments in Gmail?"	(Is there, size, type limitation, file, attachments, Gmail)
2. "Can I drag and drop a file to attach it to a message?"	(Can I, drag, drop, file, attach, message)
3. "How can I copy a file attachment from one message to another?"	(How, copy, file, attachment, message, another)
4. "Can I attach a message or conversation to a new message?"	(Can I, attach, message, conversation, new message)

FIGURE B.4: Test 4

"Label"	
Possible Query	**Possible JFA**
1. "There are no folders in Gmail. How do I organize my messages?", "Label"	(no, folders, Gmail, How, organize, messages)
2. "How many labels can I create?", "Label"	(How, labels,create)
3. "Can I apply more than one label to a single email message?", "Label"	(Can I, apply, one, label, single, email, message)
4. "Can I nest labels like I nested folders in old mail program?", "Label"	(Can I, nest, labels, nested, folders, old, mail, program)
5. "Does deleting a label delete any messages that have that label?", "Label"	(Does, deleting, label, delete, messages)

FIGURE B.5: Test 5

"Mobile App"	
Possible Query	**Possible JFA**
1 ."Where can I find the WitsM app?"	(Where, find, WitsM app)
2. "How do I remove the app from my device?"	(How, remove, app, device)
3. "Which features and resources do I have access to on the app? "	(Which, features, resources, have, access, app)
4. "How do I view my timetable? "	(How, view, timetable)
5. "How do I book a meal?"	(How, book, meal)
6. "What can I see in Comms?"	(What, see, comms)
7. "Which devices does the WitsM app support?"	(Which, devices, WitsM app, support)
8. "How do I log in to the WitsM App?"	(How, log, WitsM app)
9. "I forgot my login details. What do I do?"	(What, do, forgot, login, details)
10. "Can I change the dashboard / homescreen?"	(Can I, change, dashboard, homescreen)
11. "Can I have this app on multiple devices?"	(Can I, have, app, multiple, devices)
12. "Can another person log into the app on my device with their login details?"	(Can, another person, log, app, device, login, details)
13. "Does the app remember login details?"	(Does, app, remember, login, details)
14. "How do I sign out from the app?"	(How, sign out, app)

FIGURE B.6: Test 6

"Spam"	
Possible Query	**Possible JFA**
1. "How long do messages remain in my Spam folder?"	(How, long, messages, remain, spam, folder)
2. "How do I prevent messages from specific senders from being tagged as spam?"	(How, prevent, messages, specific, senders, tagged, spam)

FIGURE B.7: Test 7

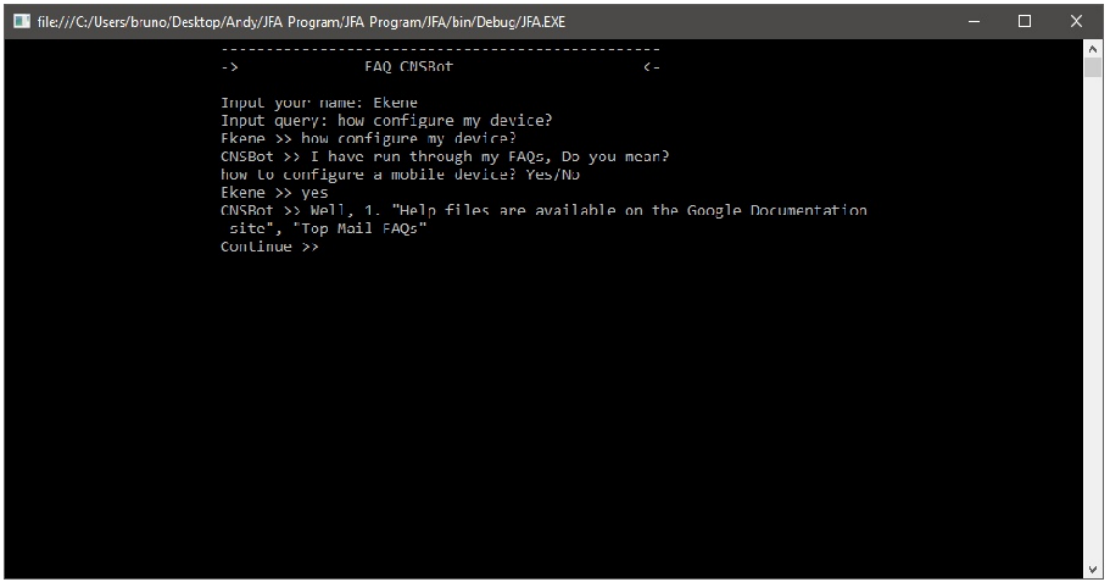


FIGURE B.8: Test 4