A Robust Natural Language Question Answering System for Customer Helpdesk Applications

Lahiru Thilina Samarakoon

(108010U)

Thesis submitted in partial fulfilment of the requirements for the degree Master of Philosophy

Department of Electrical Engineering

University of Moratuwa

Sri Lanka

June 2012

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree of Diploma in any other University of Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: Date:

The above candidate has carried out research for the MPhil thesis under my supervision.

Signature of the supervisor: Date:

Signature of the supervisor: Date:

ABSTRACT

This thesis describes a restricted-domain question answering system which can be used in automating a customer helpdesk of a commercial organization. Even though there has been an increasing interest in data-driven methods over the past decade to achieve more natural human-machine interactions, such methods require a large amount of manually labeled representative data on how user converses with a machine. However, this is a requirement that is difficult to be satisfied in the early phase of system development. In addition, the systems should be maintainable by a domain expert who is less technically skilled when compared to a computer engineer. The knowledge based approach that is presented here is aimed at maximally making use of the user experience available with the customer service representatives (CSRs) in the organization and presents how true representative data can be collected. The approach takes into account the syntactic, lexical, and morphological variations, as well as a way of synonym transduction that is allowed to vary over the system's knowledge base. The query understanding method, which is based on a statistical classifier, a ranking algorithm based on Vector Space Model (VSM) and a pattern writing process, takes into account the intent, context, and content components of natural language meaning as well as the word order. A genetic algorithm-based method is presented for finding the domain specific ranking parameters. An evaluation of the approach is presented by deploying a system in a real-world enterprise helpdesk environment in the telecommunication domain. The evaluation shows that the system is able to answer user questions with an accuracy of 94.4%. Furthermore, maintenance of the deployed system is carried out by CSRs successfully.

Keywords:
Question Answering, Automated Customer Helpdesk, Vector Space Model.

ACKNOWLEDGEMENTS

I would like to thank Exetel Pty Ltd, Australia for sponsoring the project. Special thank goes to the Sri Lanka Institute of Information Technology (SLIIT) for providing the facilities of their research center. I also thank the University of Moratuwa for their support.

Special thanks to my supervisor, Professor Sisil Kumarawadu, for his constant encouragement and support, and the rest of my committee, Dr. Ranga Rodrigo, Dr. Asanka Rodrigo, Dr Malitha Wijesundara, Dr. Koliya Pulasinghe and Dr. Harsha Abeykone, for their assistance to guide this work.

Most importantly, I would like to thank my family for their love and support of this endeavor.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	Description
CSR	Customer Service Representative
QA	Question Answering
NLP	Natural Language Processing
IR	Information Retrieval
FAQ	Frequently Asked Question
VSM	Vector Space Model
NLU	Natural Language Understanding
GA	Genetic Algorithm
KB	Knowledge Base
AI	Artificial Intelligence
UC	Unix Consultant
MSRP	Microsoft Research Paraphrase
RBF	Radial Basis Function
BM	Boolean Model
MRR	Mean Reciprocal Rank
UE	User Experience
ADSL	Asymmetric Digital Subscriber Line

CHAPTER 1 INTRODUCTION

1.1 Automated Helpdesks

Many organizations maintain helpdesks to present a single point of contact to their customers. At these helpdesks, customers interact with well-trained customer service representatives (CSRs) who answer their queries and complaints. The information exchanged in a helpdesk between customers and CSRs are highly depended on the business domain of the organization.

Many helpdesks provide one or more online self-service tools for the usage of their customers to resolve their problems. These tools normally include a website with answers to frequent user questions and a trouble ticket tracking system. The demand for the use of automated question answering (QA) systems has increased significantly. These QA systems provide front-line support to their customers and extend the helpdesk's hours of availability to 24 hours a day and seven days a week. In addition, the availability of an automated QA system can reduce the demand for direct interaction between customers and CSRs. In traditional helpdesks, customers can get frustrated by long waits on call and email queues. Hence, automated solutions can increase customer satisfaction and retention while reducing costs.

In this thesis, research is focused on incorporating domain specific information of a real-world helpdesk into current state-of-the-art QA technology to automate helpdesk's question answering process. Using this approach, an automated QA system has been developed and deployed for a real-world enterprise helpdesk which provides a range of Internet and telephone services to its customers. All the evaluations were conducted on that system.

1.2 Question Answering

Question-answering (QA) is the most natural way of exchanging information in human interaction. The term question answering is used to describe the task of returning a particular piece of information in response to a question posed by users in human language. For example, a user interested in Cricket may ask "*Who won the Cricket world cup in 1996?*"to which the system might reply as "*Sri Lanka*". In contrast, a search engine retrieves ranked documents, which the user has to read and locate the answer to satisfy his information need.

Figure 1.1 shows a block diagram of a modern QA system, consisting of three phases: question processing, passage retrieval and ranking, and answer processing [15]. At the highest level of abstraction, a QA system can be thought of as a pipeline consisting of an Information Retrieval (IR) component surrounded by Natural Language Processing (NLP) components [16]. The first step, Question Processing, analyzes a natural language question to formulate a keyword query as an input to the ranking algorithm and detect the answer type by user query classification. Then, the formulated query is passed to the embedded IR component to retrieve passages likely to contain the answer. Then, in the final stage, Answer Processing, system extracts specific answers from the retrieved passages.

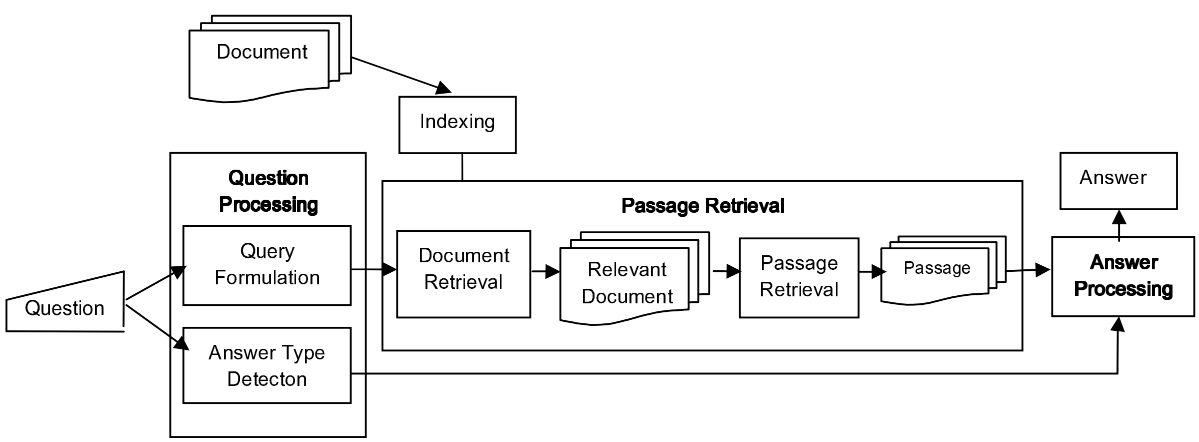


Figure 1-1: A Question Answering System.

1.3 Hypothesis

For a given helpdesk system, there exists a methodology to automate the question answering of customers by answering their questions with an accuracy of 90% or more. In addition, the developed system should be maintainable by a domain expert who is less technically skilled when compared to a computer engineer.

1.4 Contributions

The overall goal of this research is to introduce a methodology to automate the question answering (QA) process of customer helpdesks which can be used in situations where initially available data is not sufficient for data-driven approaches to system development. More specifically, the contributions of the presented approach consist of:

- This thesis proposes a Natural Language Understanding (NLU) method for helpdesk automation that is based on detecting the service type and the issue of a user question.

- A mechanism for knowledge base population and paraphrase detection is introduced, to be performed by a person with application domain experience and word processing skills.

- This work also proposes a method to incorporate domain specific prior knowledge to train a service detection classifier based on Support Vector Machines (SVM). Furthermore, the thesis also highlights the possibility of improving the service detection using user specific information which was not available at the time of training the classifier.

- A ranking algorithm which is proposed for issue identification is derived from the vector space model (VSM). A new technique is introduced to overcome the loss of information (especially word order) due to the bag-of-words nature of the VSM model. Moreover, a Genetic Algorithm (GA) based method is introduced to the ranking formula which can be trained for any domain specific training set with the necessary adjustments.

- Finally, the proposed approach is evaluated by developing and deploying an automated helpdesk system for real customers.

1.5 Outline of this Thesis

The remaining chapters of this thesis are organized as follows:

- Chapter 2 explains previous research on QA and related state-of-the-art systems.
- In Chapter 3, the philosophy of approach of this research is described in detail.
- Chapter 4 is the critical analysis of results and evaluations of the automated QA system.
- In Chapter 5, the conclusion is presented, In addition, few pointers to possible future works are discussed.

CHAPTER 2 BACKGROUND AND RELATED WORK

This chapter studies the early and state-of-the-art work on Question Answering (QA), focusing on its relationship to Information Retrieval (IR) and Natural Language Processing (NLP). This is not an exhaustive survey of the field of QA, but instead of an attempt to discuss the task of QA for automation of customer helpdesks for restricted domains and to identify issues that are addressed in this thesis.

2.1 Information Retrieval

Question Answering (QA) systems focus on finding answers to user questions in a collection of documents. Most of the time, it is not practical to linearly scan each document in a collection for every user question. Therefore, modern QA systems use an Information Retrieval (IR) based component to index the documents in advance, and provide a ranked retrieval mechanism to query the index to retrieve only the documents that are relevant to the question.

Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers) [14]. In IR, a **document** refers to the unit of text indexed in the system and available for retrieval. A **collection** refers to a set of documents being used to satisfy user requests. A **term** refers to a lexical item that occurs in a collection and a **query** represents a user's information need expressed as a set of terms [15]. The most popular IR model used by QA systems is the Vector Space Model (VSM). In VSM, documents and queries are represented as feature vectors of terms that occur within the collection [17]. For ranked retrieval, cosine similarity metric is used to calculate the similarity scores between query vectors and document vectors. The ranking algorithm used in this thesis for issue detection is derived from VSM. In the next chapter, it is discussed in detail.

2.2 Open-Domain QA versus Restricted-Domain QA

There are two types of question and answering mechanisms:

1) Open-domain question answering mechanisms deal with natural language questions which are not constrained to a specific domain. The eighth Text Retrieval Conference (TREC-8) first organized a competition on answering open-domain factoid questions. Researchers use publicly available data sets (e.g. Reuter's data set, TREC data set) [10]. Some researchers use web as a gigantic data repository which they exploit the data redundancy for QA [4];
2) Restricted-domain question and answering mechanisms deal with natural language questions constrained to a specific domain. Automated customer helpdesk applications fall into this category.

In determining the best techniques to be used in restricted-domains, and whether the techniques used in open-domain are effective in a restricted-domain, it is worthwhile to consider the size of the data. For example, data redundancy is exploited in some open-domain QA systems. The intuition is that the size of the data increases, it becomes more likely that the answer to a specific question can be found with data-intensive methods that do not require a complex language model [4]. In contrast, redundancy techniques have lesser value in restricted-domain QA especially in the case of domains with a relatively small amount of data.

However, possibilities of applying complex NLP techniques are higher in restricted-domains since those systems have a relatively small amount of data to handle. In addition, creation and maintenance of the index is less expensive.

The characteristics of questions asked in a restricted-domain are different from those asked in open-domain. Most of the restricted domain users are experts in that domain and will use specific terminology with technical questions. Generally, questions asked by those users are more complex than the questions asked in open-domain.

Therefore, there are a lot of opportunities to apply advanced NLP techniques in restricted-domain QA systems.

2.2.1 Ontological resources

There is an important difference between available resources in open-domain and in restricted-domain. One major resource used in QA systems for knowledge representation is ontology. An ontology is usually defined as a formal explicit description of concepts in the domain of discourse, together with their attributes, roles, restrictions, and other defining features [19].

The ontologies used in open-domain QA systems are developed without any domain specific restrictions. The WordNet [11] is the most widely used open-domain ontology in the field and others include Dbpedia [20], Wikipedia Infoboxes [21]. However, applications of those open-domain ontologies are limited when used in restricted domains. The main reason for this behavior is that the information in open-domain ontologies are not balanced when compared to the restricted domain. In other words, open-domain ontologies are too coarse-grained for specific restricted domains, whereas other parts are too fine-grained and it is possible that open-domain ontologies may contain information that may have an adverse effect on the restricted domain QA systems.

For example, consider the system described in this thesis and the open-domain ontology WordNet. The deployed system is developed for a technical domain and contains a considerable amount of technical terms which are not included in WordNet. In addition, WordNet has a vast amount of information which includes a lot of word senses for some words. For instance, the word “bank” has multiple word senses, including the meanings for financial institution, sloping land and the building of a financial institution. This disambiguation is unnecessary for the proposed QA system in which the term “*bank*” only refers to the financial institution. Therefore, the impact of word-sense disambiguation is reduced in restricted domains.

In most restricted-domain QA systems, the ontology is built manually using application specific data. Moreover, manual creation of a complete ontology is a time consuming task. Therefore, in the proposed system, a very simple ontology is used which is only constrained to service types.

2.3 Paradigms for QA

Research in QA has evolved from two different paradigms:

1) IR based approach pioneered in the annual TREC evaluations and used in commercial systems like IBM Watson [25] and Google [27]. In this paradigm, question answering focuses on finding text excerpts that contain the answer within large collections of documents using fast and shallow methods.

2) The knowledge base (KB) approach is focused on building an answer from understanding the parse tree or the structure of the question. These systems have its knowledge encoded in databases as an information source. Therefore, the question answering is restricted only to the information previously encoded in the database. The benefit of this approach is that having a conceptual model of the application domain represented in the database structure which allows the use of advanced NLP techniques in order to address complex information needs of users.

The following table represents a categorization of few commercially available QA systems to above paradigms.

Table 2-1: Commercial QA Systems Categorization

IR based Systems	KB based Systems
TREC [23]	Apple Siri [22]
IBM Watson [25]	IBM Watson
Google [27]	Wolfram Alpha [24]
	True Knowledge Evi [26]

Most of the modern systems use hybrid approaches where they combine both IR and KB based approaches. Generally, these systems build a shallow representation of the query and use IR based methods to come up with sets of candidate answers to questions. Then, KB based methods are used to score or filter these candidate answers. In other words, these systems use IR based methods to find candidate answers and KB method to score them. The system described in this thesis is also a hybrid system.

2.4 Early Work in Restricted-Domain QA

Most of the early work on restricted-domain QA is focused on storing knowledge in a database and providing a natural language interface. Two examples of these systems are Lunar [29] which answered questions about analysis of rock samples from moon missions and Baseball [28] which was restricted to baseball games played in the American league over one season. Both systems were very successful due to the very specific nature of their domains which enabled the construction of appropriate comprehensive databases.

The first system developed using complex NLP techniques was the Berkeley Unix Consultant (UC) project [30] which used the domain of the UNIX operating system to develop a helpdesk. In the UC project, NLP techniques were used to analyze user questions and to create meaning representations. Other traditional QA systems like CMU's Phoenix [8], SRI's Gemini [9], and MIT's TINA [7] were developed using manual translation of textual information into knowledge bases using handwritten rules. Even modern systems adopt handcrafted rules-based approach to develop systems when little or no data is available, which is usually the case in the early phase of an application. However, it has disadvantages such as lack of robustness, poor accuracy, and inconsistency when designed by different individuals [1].

2.5 State-of-the-art Work in Restricted-Domain QA

In restricted-domain QA, current research is focused on leveraging domain specific characteristics to improve the performance and practicability of the system. In order to do this, system developers need to collect true representative data and analyze them. This involves coming up with strategies for knowledge extraction and populating databases. PICO system done by Demner-Fushman et. al [41] and the approach proposed by Sang at al [42] describe several strategies for the domain of medicine. In 2004, Niu and Hirst [43] presented an approach to automatically build an ontology for the medical domain by identifying semantic classes and relations between them. Yu, Sable, and Zhu (2005) [44] described a classification algorithm to classify medical questions to an ontology. Benamara (2004) [45] described Webcoop a logic-based system that uses advanced reasoning procedures and knowledge representation approach to answer natural language questions in the tourism domain.

Modern restricted-domain QA systems employ two main different approaches to arrange domain knowledge: 1) Knowledge based 2) Free text based.

Systems that support knowledge based question answering includes the AP Chemistry question answering system [46], Cyc [47], the Botany Knowledge Base system [48], the two systems developed for DARPA's High Performance Knowledge Base (HPKB) project [49], and the two systems developed for DARPA's Rapid Knowledge Formation (RKF) project [50]. These systems organized their knowledge bases according to a defined structure that is built by taking the anticipated questions into account. Furthermore, these systems need human interaction in knowledge base creation and population.

In free text based QA, knowledge bases are made of a collection of unstructured text. Systems that use this approach, depends intensively on ontologies. The main reason for this observation is that these ontologies are used to overcome the unstructured nature of their knowledge [51]. The system described in this thesis employs the knowledge based approach.

Voicetone [1] is a successful restricted-domain QA system that was developed to automate customer helpdesk applications. It describes an intent oriented approach in organizing the knowledge base and uses a statistical classifier for question understanding. For intent identification they have come up with a new predicate-argument representation for semantic contents of the knowledge base. This intent oriented nature of Voicetone has enabled the facility to deploy applications rapidly for new domains with minimal human intervention. However, extra care need be provided for the maintenance of Voicetone and should be performed by engineers. Therefore, Voicetone lacks the important feature, the ability of maintaining the system by a less technically skilled domain expert. This thesis takes the above feature into consideration.

2.6 Choice of the Text Classification Algorithm

Text classification problem can be defined as follows:

Given a description $d \in X$ of a document, where X is the document space; and a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$. Using a training-set D of labelled documents $< d, c >$, it is needed to learn a classification function γ that maps documents to classes [14].

$$\gamma : X \rightarrow C \tag{2.1}$$

Then learned classifier function γ is used to classify new documents automatically. This learning method is called supervised learning because a human need to define classes and label training documents. In the deployed system, service types of user questions are identified by a text classifier. In deciding what classification algorithm to use, following factors were considered.

2.6.1 Generative versus Discriminative?

Based on the underline probabilistic model, classification algorithms can be categorized into two types: 1) Generative models 2) Discriminative models. Generative models give probabilities $P(d, c)$ and try to maximize the joint likelihood whereas discriminative models give probabilities $P(d|c)$ by taking the data as given and modelling only the conditional probability of the class.

In recent works, discriminative or conditional models are preferred in NLP and IR tasks because of these models give high accuracy performance when compared to generative models [52] , [69], [70]. The table below reports a result to support this observation of text classification when applied to Word Sense Disambiguation (WSD) [52].

Table 2-2: Discriminative versus Generative Models

Training Set		Test Set	
Objective	**Accuracy**	**Objective**	**Accuracy**
Generative	86.8	Generative	73.6
Descriminative	98.5	Descriminative	76.1

Due to high accuracy reports, It was decided to adopt a discriminative classifier in the system. In deciding the specific classification algorithm, regularized Support Vector Machines (SVM) classifier was chosen as high performance is recorded in the literature [52] when SVMs used with regularization with a limited number of training data.

2.7 Paraphrase Detection

Paraphrase detection is the problem of detecting whether two phrases or two sentences are similar in meaning, and this is considered as one of the difficult problems in NLP. In a QA system, users can ask the same question in many different forms. Therefore, detecting paraphrases is very important for real-world QA systems. Table 2-3 summarizes an evaluation [31] of state-of-the-art paraphrase detection algorithms on Microsoft Research Paraphrase Corpus (MSRP) [53].

Table 2-3: Paraphrase Detection Algorithms

Algorithm	Description	Accuracy	F
FHS [33]	supervised combination of MT evaluation measures as features	75.0%	82.7%
KM [35]	supervised combination of lexical and semantic features	76.6%	79.6%
RMLMG [38]	unsupervised graph subsumption	70.6%	80.5%
MCS [36]	unsupervised combination of several word similarity measures	70.3%	81.3%
STS [34]	unsupervised combination of semantic and string similarity	72.6%	81.3%
QKC [37]	supervised sentence dissimilarity classification	72.0%	81.6%
matrixJcn [32]	unsupervised JCN WordNet similarity with matrix	74.1%	82.4%
SHPNM [39]	supervised recursive auto encoder with dynamic pooling	76.8%	83.6%
WDDP [40]	supervised dependency-based features	75.6%	83.0%

However, these paraphrase detection algorithms have a weaker impact in restricted-domain QA as most of these methods use tools developed for the open-domain QA. For example, the developed system should identify both S_1 and S_2 sentences mentioned below as paraphrases that are very different in the normal context.

S_1: How can I cancel my service?

S_2: I want to leave Exetel?

To calculate sentence similarity, some of the algorithms mentioned in Table 2-3 use WordNet [11] based word similarity measures that are suited for open-domain. Most commonly used word similarity measures are mentioned below.

$$sim_{path}(c_1, c_2) = \frac{1}{pathlen(c_1, c_2)} \quad [54] \tag{2.2}$$

$$sim_{resnik}(c_1, c_2) = -\log P(LCS(c_1, c_2)) \quad [55] \tag{2.3}$$

$$sim_{lin}(c_1, c_2) = \frac{2 \log P(LCS(c_1, c_2))}{\log P(c_1) + \log P(c_2)} \quad [56] \tag{2.4}$$

$$sim_{jc}(c_1, c_2) = \frac{1}{\log P(c_1) + \log P(c_2) - 2 \log P(LCS(c_1, c_2))} \quad [57] \tag{2.5}$$

$$sim_{eLesk}(c_1, c_2) = \sum_{r,q \in RELS} overlap\left(gloss\left(r(c_1)\right), gloss\left(q(c_2)\right)\right) [58] \tag{2.6}$$

Every method mentioned above use the structure of WordNet [11] in word similarity calculations. However, For this automated helpdesk system, word similarity measures perform poorly as the content and the hierarchy of WordNet has a very weak connection to the domain. Domain knowledge is important in paraphrase detection and word similarity calculations. Therefore, for restricted-domain QA, it is needed to come up with new paraphrase detection models and word similarity measures. One promising approach is to build a thesaurus for the domain with relations and introduce new similarity measures and algorithms based on the structure of the thesaurus. This approach involves a lot of work that will not fit in the scope of this thesis. In addition, maintaining a thesaurus is a difficult task that needs a comprehensive understanding of linguistics and domain expertise. Furthermore, it is very challenging for a less technically skilled person. Therefore, a method called pattern writing is proposed that can be performed by a less technically skilled domain expert. Chapter 3 contains a full section on pattern writing.

2.8 Choice of the Ranking Algorithm

In an Information retrieval (IR) system, information needs of users are converted into queries. Both queries and documents are transferred into an internal representation depending on the underlying model. Then the ranking algorithm matches a query representation to document representations to determine the documents that satisfy information needs of users.

Ranking problem can be defined as follows:
Given a set of documents $D = \{d_1, d_2, \dots \dots, d_n\}$ and a query q, in what order the subset of relevant documents $D_r = \{d_{r1}, d_{r2}, \dots \dots, d_{rm}\}$ should be returned to the user. The ranking algorithm should retrieve the best document to be at rank 1, second best to be at rank 2 and so on.

2.8.1 Vector Space Models versus Probabilistic Models

Based on the underlying model, ranking algorithms can be categorized into two types: 1) Vector Space Models (VSM) 2) Probabilistic Models. In VSMs, both queries and documents are represented as vectors in "term space". In contrast, probabilistic models rank documents by their estimated probability of relevance with respect to the query.

Both VSM and probabilistic models support natural language queries and those convert queries and documents to the same internal representation according to the underlying model. In addition, both modelling approaches support ranked retrieval and relevance feedback. The primary difference is based on the theory.

In probabilistic models, probabilities need to be estimated as accurately as possible according to the available data and the model highly depends on this data. Terms are modelled as occurring in documents independently and these models do not recognize any association between terms. In a sense this assumption is equivalent to an assumption of the VSM, where each term is a dimension that is orthogonal to all other terms. Therefore, word order similarities are discarded in both models.

In automated helpdesk application development, it is difficult to collect true representative data which is needed for the estimation of probabilities, especially in the early stages. In addition, due to rapid changes of information and changes associated with the knowledge base population, the model need to be adjusted. Therefore, in the presented approach, VSM is used as the underlying model in the ranking algorithm. To incorporate word order similiarities, a method called n-gram is introduced to the ranking formula. It is disussed in Section 3.7.

2.9 Main Considerations of Building the QA System

The section emphasizes the main points to be taken into consideration when designing a QA system for a specific domain [18]. They can be listed as follows:

- Domain query system analysis
- Domain knowledge selection
- Domain knowledge representation
- System interface design

In the remaining part of this section, how the deployed automated customer helpdesk system took above factors into consideration is discussed.

Domain query system analysis: From the beginning of the system development, it is important to know the different ways users ask questions to satisfy their information needs. Even though, it is possible to ask CSRs for sample questions, studies [1] showed that language characteristics of human-machine interactions and human-human interactions are different. Therefore, it is important to collect data on how user converses with a machine. To collect questions, a web interface is provided to users and encouraged them to ask questions. Those questions were directly transferred into a trouble tracking system where CSRs provided answers. Collected questions needed to be analyzed, especially to mark stopwords and identify paraphrases. Then, those were manually classified by domain experts to services and products of the company for further analysis of building a rule based classifier.

Domain knowledge selection: In selecting the domain knowledge of the QA system, more general queries were preferred to user specific questions. This also simplified the pattern writing for paraphrase detection. For example, the approach prefers "*My Internet has been dropping after connection changes*" to "*I was previously on a 512/128 ADSL1 connection and recently upgraded to a 8000/386 ADSL1 connection. The Internet has been dropping out since then*" where both questions had the same answer.

Domain knowledge representation: It is important to define an internal model to represent domain knowledge. The selected model is also the factor that determines the kind of operational processes and algorithms required to build the QA system. In the system, a question is wrapped with other essential information in a unit called document, and the document is indexed using IR techniques to be retrieved when the question is asked by a user. This model is discussed in detail in Chapter 3 in Section 3.1 under the topic "*Understanding the User Query*".

System interface design: The system interface plays a major role in the mode of communication between users and the system. Therefore, it is important to tailor the system interface according to the characteristics of the domain and user requirements. In our system, a web based interface is provided for users to ask their questions by typing in natural language. Since the domain of the QA system is highly related to the Internet, it is safe to assume that all the users are familiar with using web based interfaces.

CHAPTER 3 : PHILOSOPHY OF APPROACH

The approach used in answering customer questions is based on understanding the user question in spite of the various different ways a given question can be asked and subsequently generating a predetermined fixed answer. The objective is to shift the burden of answering typical and frequent questions from the customer service representatives (CSRs) to the machine and hence allowing them to save time to attend more complex and difficult questions. To this end, a collection of latest frequently asked questions (FAQ) collected over a period of time is used to develop the knowledge base of the QA system. The FAQs are often used as a proxy for truly representative data since such data are hardly available in the early phases of system developments [5].

3.1 Understanding the User Query

Generally, for the understanding of a user query in an automated QA system of a customer helpdesk of product and services, it is needed extract two important pieces of information from the query.

- Service or the product related to the query
- Issue related to the product or the service

For instance, in the question *"What is the availability of ADSL service?"*,it is needed to extract *"ADSL"* as the service and the issue of the question should be identified as *"availability"*.

For the detection of the service, a machine learning based classifier is proposed. The issue detection is based on a ranking algorithm that is derived from the Vector Space Model (VSM). Both service detection and issue detection are discussed in the following sections of this chapter.

In addition, system needs to understand a variety of different yet most common ways a given question can be asked by the user. For example, in response to the following two questions, the system will respond to the user informing the time taken for activating an ADSL service.

- How long does it take to activate ADSL connection?
- When are you going to provision my ADSL service?

Any of above different forms of the question should be responded with the same answer, “This takes between three to five business days since the date of application”. The different most common ways a given question can be asked are termed as candidate forms. The mechanism used to incorporate these candidate forms in the knowledge base is discussed in Section 3.3 under pattern writing.

In the system, understanding the question is based on a question classifier for service detection and a ranking algorithm to retrieve top ranked documents according to the issue. For calculation of ranking scores, An IR based mechanism is used to create and index documents. A document refers to the unit of the text indexed in the system. Depending on the application, a document can refer to anything from common artifacts like newspaper articles or encyclopedia entries to smaller units such as paragraphs and sentences [15]. In the system, documents are created with four fields.

1) Original Question.
2) Pattern.
3) Answer.
4) Service Type.

The ranking algorithm only needs the content of the Pattern field. This Pattern field is used to incorporate all candidate forms for the question labelled in the document. The Service Type field is used to filter documents according to the correct service type identified by the service type detection classifier. Original Question and Answer fields are used as references for answer retrieval after ranking is done. This one-to-one mapping between an original question and a document, simplifies answer processing to retrieval of the answer field content from the top ranked document. The documents are ranked according to the user question to determine the top ranked document. If the ranking score of the top ranked document is above a threshold level, the query is assumed to have understood. Otherwise, the user is automatically connected with a human agent, a CSR. The threshold level is empirically set by qualitatively analyzing the system performance in the real world.

3.2 The System Architecture

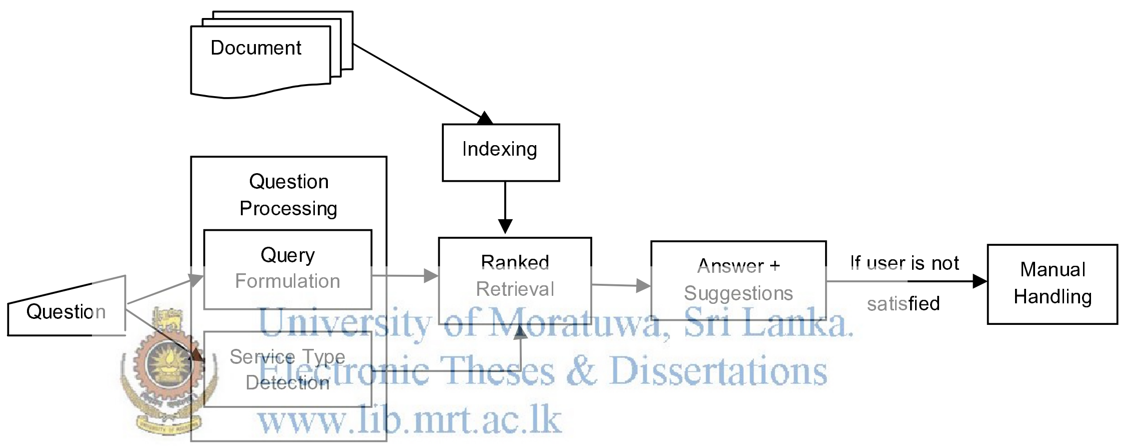


Figure 3-1: System Architecture

Figure 3-1 is the block diagram of the developed question answering system. First, documents are processed by the indexing algorithm which creates an index [14]. This index is created in advance to avoid linear scanning of documents for each user query. When a question is asked, it is processed to identify the service type and formulate a query for ranked retrieval. A classifier based on machine learning is used to identify the service of the user query. During the query formulation, input normalization techniques are applied. In this step, stop words are removed from the input and lemmatization is applied to normalize lexical and morphological variations. This step is also used to mark issue related terms for boosting. In the next step, the ranking algorithm retrieves top ranked results from the index according to the formulated query. This process is called ranked retrieval [14]. The document which has the highest score corresponds to the answer to the user query. This answer is given to the user. If the user is not satisfied with the answer, original question fields of the next four top ranked documents are displayed as suggestions. The user can view the corresponding answer fields by clicking on them. If the user is still not satisfied, the query is automatically referred to an online ticket tracking system where CSRs answer.

3.3 Pattern Writing

This section describes pattern writing, the process that is developed to address the problem of paraphrase detection in the presented approach.

In a complete user query, some of the words and phrases often carry relatively more information that helps to understand the query than the other words. For example, let's consider following two candidate forms.

- How long does it take to activate ADSL connection?
- When are you going to provision my ADSL service

In this example, salient terms like "ADSL" highlight the type of service and "activate/provision" correspond to the service related issue. The candidate forms are combined to create patterns as follows.

(How long/When) (activate/provision) ADSL (service/ connection)

Terms such as "does", "it", "to", "are", "you", "my", etc carry little or no information and hence are simply ignored. Such terms are known as stop words in information retrieval (IR) community. The term, "take", does not significantly contribute to the semantic content in the present context in this example.

Pattern writing is a major part of the system development effort. The patterns are written document by document to ensure they are context dependent to be within the context of the current question-answer pair. In order to write a pattern, most common candidate forms need to be identified. Patterns should be written incorporating the contextual information in the current context of the given question-answer pair in each individual document and also ensuring consistency across the documents in the knowledge base. Hence, pattern writer requires user experience (UE) expertise to understand the language characteristics of users in the specific application domain.

To better incorporate domain specific language characteristics in pattern writing, in the early phases of the development, pattern writers carefully study collections of transcribed voice dialogs between users and the customer service representatives (CSRs) and lists of salient words and phrases regularly generated by the experienced CSRs in active service. Pattern writing is a continuous process. After the deployment of the initial system, recorded user questions are continuously analyzed by pattern writers, and characteristics learn from these analysis are applied when writing new patterns or modifying them. However, it is natural for this process to involve some subjective judgment, which can cause inconsistency in pattern writing. To minimize such inconsistency, each pattern is verified by a different pattern writer. Clearly, more consistently written patterns help create better question answering systems. Potential synonymic variations are incorporated by taking into account the words and phrases that are similar in meaning to each other. They are separated by slashes within brackets in the patterns. This way, the usage of a synonym transducer is avoided. Synonym transducers, which are commonly used in contemporary methods [1], replace each word in the user input with its key synonym. Synonym transduction is local and hence should be allowed to differ from document to document. For instance, the term "mobile" stands for two different meanings in the following user inputs.

- I want to change my mobile plan
- What should I do if I lost my mobile

In understanding the user query, not just the propositional or literal content, but also the sense user makes in the context is considered. This is possible because patterns are written in the context of the original question-answer pairs in each document in the knowledge base.

3.4 Data Collection

In automated helpdesk application developments, it is important to gather representative data on how users converse with a machine. It has been shown that the language characteristics of the responses to machine prompts is significantly different from those to humans [1], [66], [67]. State-of-the-art systems collect application data using the wizard-of-oz approach. In a wizard-of-oz approach, a human acts on behalf of the system. Users of the system do not know about this and believe that they are using a machine. However, in the wizard-of-oz approach, it is difficult to maintain the availability for 24 hours. In addition, this approach needs more than one human operator when users are accessing the system concurrently.

In the presented approach, an initial system is deployed with a limited number of questions in the knowledge base. At this stage, only the ranking algorithm is used without the classifier as training data was not available. After recording user's question, the user is informed that his question is being transferred into a trouble ticket tracking system where customer service representatives (CSRs) answer.

3.5 Service Type Detection

An efficient and effective question answering system requires large numbers of Q-A pairs. In an approach based on natural language understanding (NLU) as presented in this thesis, accurate understanding of the question becomes very difficult when there are a large number of questions in a common knowledge base with close semantic meanings. Hence, a modular approach where the knowledge base is clustered into several clusters seems to be the most appropriate. Human customer services systems are also organized this way and an automated system should be of no exception. In the proposed approach the knowledge base is clustered into service-type based clusters for efficient retrieval. In addition, detecting the service type of a user query is essential for NLU. Table 3-1 shows the list of service types of the system.

Service Type	Description
ADSL	Questions related to ADSL Internet connections.
Fibre	Questions related to Fibre Internet connections.
Mobile Broadband	Questions related to Mobile broadband Internet connections.
Wired Telephony	Questions related to landline telephone connections.
Mobile Telephony	Questions related to mobile telephony.
Wireless Service	Questions which are common to Mobile Broadband and Mobile Telephony.
Messaging	Questions related to SMS, Email and Voicemail services.
VOIP	Questions related to Voice over IP services.
VPN	Questions related to Virtual Private Networks.
Hosting	Questions related to domain hosting.
Common	Detecting the service type is not required to answer questions belongs to this type.

Irrelevant	Questions which are not related to the business domain of the system.

Table 3-1: Definition of Service Types of the QA System

3.5.1 Rule based classifier for dataset formation

Often, one of the biggest practical challenges in fielding a machine learning classifier in real applications is creating or obtaining enough training data. To create a high quality training data set, a rule based classifier was developed. The questions collected over a period of time was classified using this rule based classifier to create high quality data set.

The rule based classifier contains term vectors that contain words or phrases. Classification rules are formulated by combining these term vectors using logical operators. The support of a domain expert is necessary in creating those term vectors and rules. Following section only displays few term vectors and rules. The complete classification algorithm is mentioned in the Appendix.

Term Vectors for Internet$_1$, Internet$_2$, CopperLine, ADSL are mentioned below.

$$\boldsymbol{w}_{Internet1} = [internet, broadband, www, authenticate, auth, bandwidth, bitrate, bit\ rate, datarate, bps, speed, dataWiFi, WiFi, modem, sync, ppp, splitter, router, relocate, relocation]$$

$$\boldsymbol{w}_{Internet2} = [data, upload, download, IP, web, website, webpage, URL, brows, browser, domain, FTP, search\ engine, TCPIP, ethernet, telnet, LAN, WAN, PC, computer, laptop, tablet, iPad, notebook, webcam, webcast, codec]$$

$\boldsymbol{w}_{CopperLine}$
$= [copper, landline, land\ line, wireline, fixed\ line, wire, receiver, PSTN,$
$landphone, land\ phone, home\ phone]$

$\boldsymbol{w}_{ADSL}$
$= [ADSL, ADSL+, ADSL1, ADSL2, ADSL2+, DSL, HSDSL, SHDSL, microfilter,$
$micro\ filter]$

Above term vectors are combined using logical set operators to form a rule for the ADSL service.

Internet : $\boldsymbol{w}_{Internet} = \boldsymbol{w}_{Internet1} \cup \boldsymbol{w}_{Internet2}$

ADSL broadband : $\boldsymbol{w}_{ADSL} \vee (\boldsymbol{w}_{CopperLine} \wedge \boldsymbol{w}_{Internet})$

3.5.2 Incoporating prior knowledge to support vector machines

As explained in section 2.6, a Support Vector Machines (SVM) based machine learning classifier is used as the service detection mechanism. The data set created by the rule based classifier is used as the training data. The goal of the SVM classifier is to produce a model based on training data, which predicts the service types of unseen questions. The standard SVM algorithm is mentioned below [60].

$$\text{minimize: } V(w,b) = \frac{1}{2}\left||w|\right|^2 \quad (3.1)$$

$$\text{subject to : } \forall i \ : \ y_i[w^T.x_i + b] \geq 1$$

In order to deal with non-separable data as well as to be less sensitive to outliers, the soft-margin SVMs are used. A set of slack variables ε_i are introduced to allow errors or points inside the margin and a hyper-parameter C is used to tune the trade-off between the amount of accepted errors and the maximization of the margin [60].This process is called regularization.

$$\text{minimize: } V(w,b,\xi) = \frac{1}{2}\left||w|\right|^2 + C\sum_i^n \xi_i \quad (3.2)$$

$$\text{subject to: } \forall i \ : y_i[w^T.x_i + b] \geq 1 - \xi_i$$

The standard SVMs learned the decision function based only on the training set. However, in restricted-domain classification applications, a certain amount of information on the problem is usually known beforehand. Incorporation of this information into the SVM is used to increase the performance of service type detection.

In restricted-domain QA systems, incorporation of prior knowledge is used as a technique to compensate for the lack of data in building robust classifiers [61]. Prior knowledge refers to all information about the problem available in addition to the training data [59].

For the incorporation of the prior knowledge, services of domain experts are used to label training instances with weights. A weight refers to the importance of a training instance. The changed algorithm is based on Cost-Proportionate Example Weighting [62] and as follows.

$$\text{minimize: } V(w,b,\xi) = \frac{1}{2}\left\|w\right\|^2 + C\sum_i^n c_i\xi_i \qquad (3.3)$$
$$\text{subject to: } \forall i \ : y_i[w^T.x_i + b] \geq 1 - \xi_i$$

where c_i is the importance of example i .

Freely available and widely used software called LibSVM [63] is used to conduct service detection classification training and testing. The questions, which are strings of characters, have to be transformed into the representation suitable for the LibSVM software. This representation uses the bag-of-words approach with boolean weights.

LibSVM shipped with four basic kernels. They are mentioned below.

- Linear : $K(x_i, x_j) = x_i^T x_j$. (3.4)
- Polynomial : $K(x_i, x_j) = \gamma(x_i^T x_j + r)$^$d$, $\gamma > 0$. (3.5)
- Radial basis function (RBF): $K(x_i, x_j) = exp\left(-\gamma \left\|x_i - x_j\right\|^2\right), \gamma > 0$. (3.6)
- Sigmoid : $K(x_i, x_j) = tanh(\gamma x_i^T x_j + r)$. (3.7)

Here, γ, r and d are kernel parameters. In the deployed system, RBF kernel is selected due to the reasons mentioned in the practical guide [64] of LibSVM.

The best values for model parameters C and γ is not known beforehand. Consequently, a model selection procedure should be used. This is achieved by using a technique called v-fold cross-validation [64]. The goal of the cross-validation is to identify the best values for C and γ that can make accurate predictions on unseen data.

3.5.3 Exploiting user specific information

When developing question answering systems for a restricted domain, it is important to leverage every information available about users and the domain, in order to be able to properly address information needs of users.

There is a very high probability of that users asking questions about the services they have subscribed. Therefore, the list of services owned by a user is a very important information that can be used in deciding the service type of the query. Furthermore, service types of previously asked questions from the system by a particular user can be a good indicator in deciding the service.

However, above mentioned types of user specific information is not available at the training time of the classifier. Hence, it is not possible to incorporate this information in the training model. Therefore, in the deployed system, user specific information is used to validate the output of the service classifier. For instance, if the service classifier output is same as the user specific information, it is a strong indicator to support the output of the classifier.

In a scenario where the output of the classifier is of low confidence or more than one service types have high confidence values, the information about the user can be used in deciding the service type of the query. For example, if a user asked "My internet connection is slow" and he is an owner of an ADSL internet connection, it is safe to assume that the service type of the query is ADSL.

In addition, in situations when classifier output is not explained by the services owned by the user, the rule based classifier is used to check for strong evidence in deciding the final service type of the query. In circumstances of lack of information to decide the service type, Follow-up dialogs are used to acquire missing information from the user.

3.5.4 Follow-up dialogs

Experience has shown that the user, in the initial input, often describes the service related issue sufficiently, but the underlining service type, which is often required to generate the most appropriate answer. Due to this user behavior, sometimes it is impossible for the service type detecting classifier to generate outputs of high confidence. In certain situations, if the user specific information also offer little assistance, it is hard to detect the service type accurately. To overcome this problem, sub-dialogs or follow-up dialogs are used to collect clarifying information from the user. Even though, it is important to give answers to the user with minimum iterations, it is better to ask a follow-up question than giving an incorrect answer. However, these sub-dialogs are activated only if it is extremely necessary.

These sub-dialog models are designed based on analysis of real user questions that are recorded over a period of time. Term vectors and rules developed for the rule based classifier are used in activating the correct dialog model and to understand the user response.

For example, if the user question belongs to the Telephony term vector, the following sub-dialog model is used to get the missing information to decide the service type of the query.

Sub-dialog: $w_i \in \boldsymbol{w}_{Telephony}$

System: Are you referring to standard mobile telephony, landline telephony, or VoIP?

User: Mobile → Mobile Telephony

Landline → Mobile Telephony/Wired Telephony/VoIP

VoIP → VoIP

3.5.5 Outlier detection

User questions that are not directly related to the domain of the automated helpdesk are considered irrelevant questions or outliers. Even though answering outlier questions is not adding a significant value to the performance of the QA system, it will be a good indicator of the capability of the system and will attract more customers to use the system.

Few sample outlier questions asked by users with provided answers are mentioned below.

- **Question**: What is the answer of life, universe and everything?
  Anwer: Forty-two.
- **Question**: What is the meaning of life?
  Answer: Try and do good, be nice to your mother.
- **Question**: What do you think?
  Answer : I don't think
- **Question**: What color is the sky?
  Answer: Black at night.

In a scenario where triggering a sub-dialog seems more appropriate, just before activating and finding the suitable sub-dialog, a user question is checked for any issue related features. The procedure for the collection of issue related features is explained in the issue identification section. If the user question does not contain any issue related features, it is marked as an outlier. Then, the ranking algorithm is applied only on the documents that are outliers or irrelevant questions in the knowledge base, for the user question to find the correct answer if available.

3.6 Issue Identification

In an automated helpdesk, it is essential to identify both service types and issues of user queries in answering questions. This section explains the proposed methodology for issue identification.

3.6.1 Usage of dependency parsing

To distinguish features within a user question from other words, use of typed dependency parsing is introduced. In a typed dependency parse of a sentence, the links between words are labelled with grammatical relations. Consider following user questions parsed by the Stanford Dependency Parser [65].

1) Question: What is the availability of ADSL service?

   Parse: attr(is-2, What-1)

   root(ROOT-0, is-2)

   det(availability-4, the-3)

   nsubj(is-2, availability-4)

   prep(availability-4, of-5)

   amod(service-7, adsl-6)

   pobj(of-5, service-7)

2) Question: Is there an activation fee for Mobile Broadband?

   Parse: root(ROOT-0, Is-1)

   expl(Is-1, there-2)

   det(fee-5, an-3)

   amod(fee-5, activation-4)

   nsubj(Is-1, fee-5)

   prep(fee-5, for-6)

   nn(Broadband-8, Mobile-7)

   pobj(for-6, Broadband-8)

3) Question: How do I register a Domain Name?

Parse: advmod(register-4, How-1)

aux(register-4, do-2)

nsubj(register-4, I-3)

root(ROOT-0, register-4)

det(Name-7, a-5)

nn(Name-7, Domain-6)

dobj(register-4, Name-7)

According to the examples above, one can observe that most features tend to appear as either subjects or objects within sentences. This is not too surprising as subjects and objects in the sentences are usually the targets at which the users express their opinions. In addition to the above observation, it is safe to say that nouns and modifiers associated with subjects and objects also carry salient information. For instance, in the second example, the subject *"fee"* is modified by the term *"activation"* and object *"broadband"* is modified by the noun *"mobile"*. Both *activation* and *mobile* are salient information in the context of the question. Therefore, the subject and object, main verbs, nouns and adjectives associated with those important terms and adverbs that modify the main verb and adjectives are extracted.

During the pattern writing process, the list of candidate issue related features is provided to pattern writers for further analysis. Pattern writers provide the final list of terms salient to the service related issue by filtering away noisy results. These terms that carry more information are boosted by the algorithm to make them contribute more in ranking calculations. The boosting factor is determined by the use of a GA optimization, which will be discussed in the following sections. Term boosting is applied during ranked retrieval which will be discussed later with how term boosting is taken into consideration in ranking calculations.

3.6.2 Text processing

The text preprocessing is a way to introduce meaning to the data, which will make the retrieval process easier. There are two main approaches to text preprocessing: 1) Removal of elements from original text; 2) Normalization. The process of normalization can be interpreted in terms of defining equivalence classes between different representations, and the use of one of the representations for all the occurrences of that class [14].

In the presented approach, three text preprocessing techniques, namely, stopwords removal, lemmatization, and length normalization are applied. Stopwords removal and lemmatization are two input normalization techniques that we use to remove syntactic and morphological variations that might not directly contribute to the semantic content of the input. Such variations in the user query as well as in the patterns fields in the documents are removed. Note that these variations were ignored in pattern writing. This also eases off the workload of the pattern writers allowing a speedy pattern writing process.

3.6.2.1 Stop words removal

A list of stop words to discard the most common words are removed that have little information in ranking calculations. This list mainly contains articles, pronouns, conjunctions and prepositions. However, it is noteworthy that words such as when, where, what, and who are important to understand questions and may not be considered stop words. Stopwords removal is performed for both the patterns and user query. The list of stop words is highly dependent on the application domain of the QA system, and expertise of domain experts should be provided to be certain that the important information is not discarded.

3.6.2.2 Lemmatization

The goal of lemmatization is to reduce an inflected form of a word to a common base form, which is known as lemma. It is the headword that appears in a dictionary definition. For example, the words “charging”, “charged”, “charges”, and “charge” have the lemma “charge”. This is done by using a vocabulary and morphological analysis. The lemmatization is applied for both the patterns and user query.

3.6.2.3 Length Normalization

This is computed in accordance with the number of terms in the pattern. The intuition behind the length normalization is that shorter patterns must contribute more to the ranking score than longer ones. Length normalization is calculated in effect at indexing. The formula for computing the length normalization value is defined in the next section about the ranking algorithm.

3.7 Ranking Algorithm

For each user question, it is not efficient to rank every document in the index. Therefore, first the Boolean Model (BM) of IR is used to mark the documents that contain query terms. In BM, both documents to be searched and the user's query are considered as term vectors. Boolean model retrieves every document that contains one or more query terms. The retrieved documents are scored and ranked by the ranking algorithm described below.

Ranking score of a document d for query q is calculated as

$$Score(q,d) = \lambda_1 VsmScore(q,d) + \lambda_2 BigramScore(q,d) + \lambda_3 TrigramScore(q,d) \quad (3.8)$$

where, λ_1 (VSM parameter), λ_2 (bigram parameter), λ_3 (trigramparameter) are constants. the definitions of VsmScore, BigramScore, and TrigramScore are given below.

3.7.1 VSM (Vector Space Model) score

In the VsmScore calculations, all the documents and queries are represented with multi-dimensional vectors. This representation is called bag-of-words model [15]. The terms are the dimensions of the vector and term weights are calculated using Term Frequency-Inverse Document Frequency (TF-IDF) [14]. These weights represent the importance of a particular term in the document. VsmScore is based on the similarity between document and query vectors. To calculate this similarity cosine similarity is used.

$$cos(q;\ d) = \frac{V(q).V(d)}{|V(q)||V(d)|} \quad (3.9)$$

where, $V\ (d)$ and $V\ (q)$ are document and query vectors, respectively. $V(q).V(d)$ is the dot product of the weighted vectors, and $|V(q)|$ and $|V(d)|$ are their Euclidean norms. In addition to the cosine similarity, some other metrics are used in VsmScore calculation. These metrics are derived from the scoring formula of Apache Lucene [6]. The complete formula for VsmScore can be defined as

$$VsmScore(q,d) = \frac{V(q).V(d)}{|V(q)|}\ .coord(q,d)\ .dnorm(d).qboost(q) \tag{3.10}$$

where,

- $coord(q,d)$ is the coordination factor which contributes to the ranking score according to the number of matching terms. A document that contains more query terms will receive a higher score than a document that has fewer query terms. This value is computed at the search time.

- $dnorm(d)$ is a value in accordance with the number of terms in the document d. This score is calculated at the time of indexing the document.

$$dnorm(d) = \frac{\alpha}{|t : t \in d|^{0.5}} \tag{3.11}$$

  where α (length normalization parameter) is a constant and $|t : t \in d|$ is the number of terms in document d.

- $qboost(q)$ is a fator that boost ranking scores of query terms. This is known at search time.

In cosine similarity calculations, each document vector is normalized by the Euclidean length of the vector, so that all document vectors turned into unit vectors. However, this normalization removes all information on the length of the original document. This can reduce the system performance on answering short user questions due to following reasons; first, longer documents will have higher term frequency values because they contain more terms. How term frequencies contribute the ranking score is explained in equation (3.12). Second, longer documents contain more distinct terms. These factors contribute to raise the scores of longer documents which will have an undesired effect on helpdesk QA systems. To address this issue, $dnorm(d)$ factor is introduced to the VsmScore calculations which normalizes the score based on the length of the document. This form of compensation for document length is known as pivoted document length normalization [14].

The practical formula for the calculation of ranking scores is derived from the equation (3.10) and mentioned below.

$$VsmScore(q,d) = \sum_{t\ in\ q} (tf(t\ in\ d).idf(t)^2\ .boostFactor)\ .coord(q,d)\ .dnorm(d).qnorm(q) \quad (3.12)$$

where,

- $tf(t\ in\ d)$ is a measure for term frequency (number of times term t appears in the currently scoring document d). Documents that have more occurences of a given term record a higher score.

$$tf(t\ in\ d) = frequency^{0.5} \quad (3.13)$$

- $idf(t)$ stands for inverse document frequency. It is a measure of importance of the term t for overall collection of documents.

$$idf(t) = 1 + log\ [\frac{|D|}{|1 + \{d : t \epsilon d\}|}] \quad (3.14)$$

  where, $|D|$ is the total number of documents and $|\{d : t \epsilon d\}|$ is the number of documents in which the term t appears.

- boostFactor is a retrieval time boost of term t in the query q.

- $qnorm(q)$ is a normalizing factor used to make scores between queries comparable. This factor does not affect document ranking and can be computed at the start of the search.

3.7.2 Bigram score

An n-gram is a subsequence of n terms from the term sequence of a query. Bigrams and trigrams are special cases of n-grams where, n = 2 and n = 3, respectively [15]. Bag-of-words models are based on the concept that the meaning of a query resides solely in the set of words in contains. In other words, these models ignore syntactic information like word order and constituency of the words that make up the sentences in determining their meaning. For example, "*I see what I eat*" and "*I eat what I see*" has the same meaning. Bigram and trigram scores are introduced to make sure that the word order similarities contribute to the ranking score.

$BigramScore(q, d)$ is calculated as

$$BigramScore(q, d) = \sum_{\text{bigram in q}} \frac{|\{bigram : bigram \in d\}|}{|B|} \tag{3.15}$$

where, $|B|$ is the total number of bigrams in q and $\{bigram : bigram \in d\}$ is the number of common bigrams of q and d.

3.7.3 Trigram score

$TigramScore(q, d)$ is calculated as

$$TigramScore(q, d) = \sum_{trigram\ in\ q} \frac{|\{trigram : trigram \in d\}|}{|T|} \tag{3.16}$$

where, $|T|$ is the total number of trigrams in q and $\{trigram : trigram \in d\}$ is the number of common trigrams of q and d.

3.7.4 GA optimization

The best values for constants, $\lambda_1, \lambda_2, \lambda_3$ in (3.8), $boostFactor$ in (3.12) and α in (3.11) are highly dependent on the application domain. Further, these values depend on the written patterns in the knowledge base as well as the consistency and quality maintained in the manual pattern writing process. For example, $boostFactor$ depends on the list of words selected for boosting. Bigram and trigram parameters, λ_2 and λ_3 depend on the correlation of word orders of the user query and word orders of patterns. A Genetic Algorithm (GA) based optimization algorithm to regularly update the optimum values of the mentioned parameters based on the system performance in the real world. The introduction of GA optimization for the calculation of domain specific ranking parameters has increased the ability to use the system in other domains and next section discusses the possibilities of porting to other domains.

3.8 Portability to Other Domains

Since developing a restricted-domain QA system is often time-consuming, In the presented approach, the ways to reuse technologies including the code is considered. Separation of the domain knowledge from the operational knowledge is the key to enhance the system portability between domains. The ranking algorithm in this approach is developed in this way where one can use GA optimization to construct the ranking formula for the domain at hand. However, the remaining steps need human intervention as they are highly dependent on the domain knowledge. The rest of the text in this section summarizes the approach when one is developing a QA system for a new domain using the approach described in this thesis.

Collecting questions can be done via giving users a web interface without a backend knowledge base. When users ask questions, they will be redirected to a trouble ticket tracking system where CSRs will contact the user with the answer. In this way one can collect true representative data for the new domain on how users converse with a machine without making user queries go unanswered.

Then these questions need to be analyzed as described in the thesis and mark stopwords. Then, selected questions for the knowledge base should be labelled by patterns. If a mild ontology like service types can be identified, a simple rule based classifier can be used to create a high quality data set. Using this data set, statistical classifier can be trained as discussed previously. The developed GA optimization mechanism can be used to find ranking parameters for the new domain.

Therefore, deploying a QA system for other domains can be achieved by adopting this approach and by structuring domain specific information as described in this thesis.

CHAPTER 4 : RESULTS AND EVALUATION

A system has been developed and deployed to automate an enterprise customer helpdesk of Internet services. This section describes the used evaluation metrics and experimental results produced by the live system deployed in the real world.

A restricted-domain QA system is developed for a certain application, it is clear that these systems require a situated evaluation [68]. While TREC comparisons are very successful in open domain evaluations, comparisons about system performance are only useful if the systems use the same data, or at least they are in the same domain. Therefore it is insufficient and unsuitable to use a generic evaluation for restricted domains. The evaluation has to be situated in the task, domain and users for which the system is developed. Therefore, in this chapter, the system evaluation is presented in the task oriented manner. However, section 2.8 discusses and compares the presented approach with two state-of-the-art restricted-domain QA systems, WebCoop [45] and Voicetone [1].

4.1 Evaluation Metrics

4.1.1 Mean reciprocal rank (MRR)

To evaluate the performance of the ranking algorithm and the effect of text preprocessing techniques, the mean reciprocal rank (MRR) [15] is used. The MRR is defined as

$$MRR = \frac{\sum_{i=1}^{N} \frac{1}{rank_i}}{N} \tag{4.1}$$

where, N is the total number of queries in the test set and $rank_i$ is the rank of the first correct answer. For example, in response to a given user query, if the system retrieves a number of documents as relevant and only the third ranked document contains the correct answer, the MRR score corresponds to this query would be $^{1}/_{3}$.

4.1.2 F_1-Measure

To evaluate the performance of the outlier detection mechanism, the F_1-Measure is used. The definition of the F_1-Measure is given below.

$$F_1 - Measure = \frac{2\,.\,Recall\,.\,Precision}{Recall + Precision} \tag{4.2}$$

where, precision and recall are defined below.

Precision (P) is the fraction of retrieved documents that are relevant.

$$Precision = \frac{Number\ of\ Relevant\ Documents\ Retrieved}{Number\ of\ Retrieved\ Documents} \tag{4.3}$$

Recall (R) is the fraction of relevant documents that are retrieved.

$$Recall = \frac{Number\ of\ Relevant\ Documents\ Retrieved}{Number\ of\ Relevant\ Documents} \tag{4.4}$$

4.1.3 Accuracy

To evaluate the service detection classifier, *accuracy* is used and it is the fraction of classifications that are correct.

4.2 Results of Cross Validation

Before SVM is trained with the RBF kernel, it is needed to find the optimal values for the parameters C and γ. This is called the parameter search or model selection. In the deployed system, a procedure known as v-fold cross-validation is used for model selection. In v-fold cross-validation, the training set is divided into v subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining v-1 subsets [64]. 5 fold cross-validation is used in the system.

To find C and γ using cross-validation, a technique called grid search is used. Grid search tries various pairs of C and γ values and the one with the best cross-validation accuracy is selected.

In evaluations, two SVM classifiers are trained for service type detection. One is trained soft margin SVM algorithm with domain specific weights for prior knowledge incorporation and the other is trained using standard soft margin SVM algorithm without domain specific weights. Optimal values for the parameters of two models are given below.

Model	**C**	**γ**
SVM with prior weights	2.0	0.125
SVM without prior weights	32.0	0.0078125

Table 4-1: Results of Cross-Validation

4.3 Service Detection Classifier Performance

	SVM with weights	SVM without weights
Training accuracy	98.8006% (659/667)	97.7511% (652/667)
Testing accuracy without user specific information	67.33% (101/150)	64% (96/150)
Testing accuracy with user specific information	82% (123/150)	77.33% (116/150)

Table 4-2: Classification Accuracies

Table 4-2 shows classification accuracies for service type detecting classifiers. There were 667 training instances and 150 user questions in the test set. It can be clearly seen that the accuracies are always higher in the SVM algorithm with domain specific weights for prior knowledge incorporation. Furthermore, significant improvements were detected in both classifiers when classifier output is validated and incorporated with user specific information. Most users tend to ask questions in general terms. Therefore, user specific information is needed to decide the correct service type for their query. A few example user questions that needed information about users are given below.

Q: What is the internet service availability? Service Type : ADSL

Q: How to check if my internet is slow ? Service Type : Mobile Broadband

Q: I want to activate international calls. Service Type : Wired Telephony

4.4 Outlier Detection

Precision	**Recall**	**F_1-Measure**
0.85 (39/46)	0.90 (39/43)	0.87

Table 4-3: Performance of Outlier Detection

In Table 4-3, precision, recall and F-Measure for the outlier detection are provided. As can be seen from the table, the simple heuristic used in detecting outliers recorded good values in evaluations. However, precision recorded a low value as some normal questions were detected as outliers because few issue related features were missed from the list. Therefore, the precision can be increased by including features related to missed issues to the list.

4.5 GA Optimization

The GA based optimization algorithm was implemented using a Java genetic algorithms package called JGAP [13]. The mean reciprocal rank (MRR) was used as the fitness function. Details of the GA optimization are given below.

- Genes: The rankingParameters are used as genes.
  - VSM parameter (λ_1)
  - Bigram parameter (λ_2)
  - Trigram parameter (λ_3)
  - Length normalization parameter (α)
  - boosting factor (boostFactor)
- Parameter value ranges:

  λ_1= [0, 3],

  λ_2= [0, 2],

  λ_3= [0, 2],

  α = [0, 3],

  boostFactor= [0, 4]
- Population: The population has randomly generated 200 individuals. An individual is a candidate solution to the problem at hand.
- Generations: 300
- Fitness Function: The fitness for each individual is calculated using MRR on the same test data. The test data set contained 500 question-answer pairs.

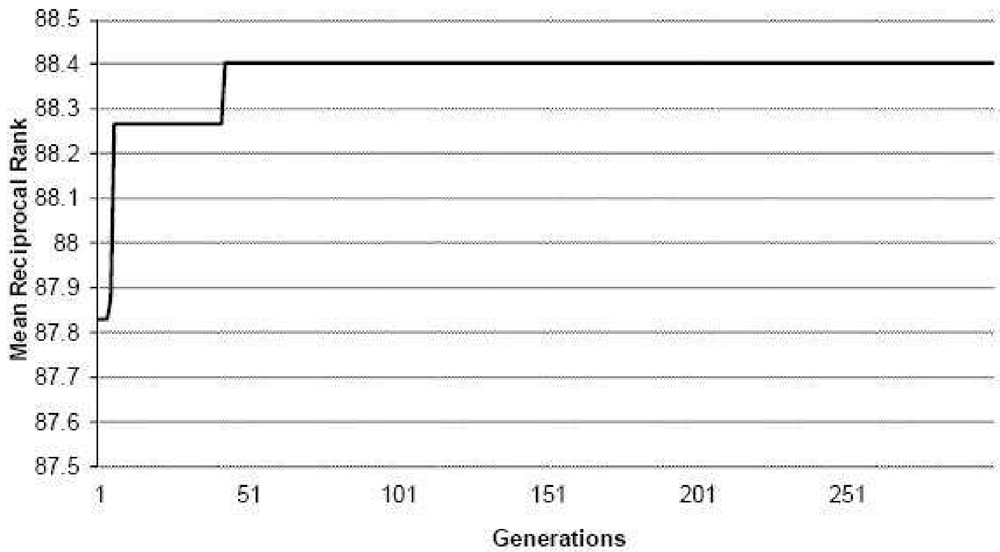


Figure 4-1: Fitness Landscape for the GA Optimization

Fig. 4-1 shows the convergence characteristics of the GA optimization on increasing number of generations. As it can be seen, after a finite number of generations, the fitness of the highest ranking solution has reached a plateau such that successive generations no longer produce better results. This is the termination condition. Results of the GA optimization are as follows.

$\lambda_1 = 1.90$

$\lambda_2 = 1.44$

$\lambda_3 = 1.29$

$\alpha = 1.87$

boostFactor= 2.65

4.6 Effects of Text Processing

Technique(s)	MRR
None	57.33%
LN	70.66%
Stop	62.67%
Lemma	64.00%
LN + Stop	74.66%
LN + Lemma	84.00%
Stop + Lemma	69.34%
LN + Stop + Lemma	89.33%

Table 4-4: Comparison of Text Preprocessing Techniques

In Table 4-4, The MRR values for the different text preprocessing techniques are provided. The MRR value for each technique is computed on the same test data set used in GA optimization. The row labeled as "None" shows the performance when none of the preprocessing techniques are applied. The technique labeled as "LN" shows the performance when only length normalization is applied. Likewise, techniques labeled as "Stop + Lemma" shows the performance when the stop words removal and lemmatization techniques are applied. Similarly, Table 0-4 summarizes the performances of all the combinations of text preprocessing techniques. Clearly, the length normalization has improved the performance significantly. The performance improvement from lemmatization is slightly higher than the improvement from stop words removal. It can be seen that the best performance is achieved when all techniques are used.

4.7 Performance of the Live System

	Number of Questions
Correctly Answered in 1st Attempt	166
Correctly Answered in Suggestions	36
Wrongly Answered	12
Identified New Questions	152
Unidentified New Questions	34

Table 4-5: Categorization of System Responses for Randomly Selected 400 User Questions

	Success Rate
Correctly Answered	77.57%
Correctly Answered (Suggestions)	94.39%
New Question Identification	81.72%

Table 4-6: Success Rates of the Live System

Table 4-5 is the categorization of responses of the live system for randomly collected 400 queries which were asked by real users. System performance is summarized in Table 4-6. The “Correctly Answered” success rate implies the rate of correctly answering in the first attempt without needing to display suggestions. The “Correctly Answered in Suggestions” success rate is a measure for system performance when a query is answered at least in suggestions. In calculation of above ratios, only the queries that the system is knowledgeable to answer are considered and neglected the new questions.

There is a possibility that the system returns answers to completely new questions. These answers will inevitably be incorrect. Understandably in a customer support helpdesk, giving a wrong answer is worse than no answer. Therefore, it is important for the system to have a mechanism to determine whether the answer is correct or wrong. A threshold is used where the system opts not to answer when the confidence level (ranking score) is low. This threshold value was determined empirically.

4.8 Discussion

In this section, the approach presented in the thesis is compared with two other state-of-the-art systems, Voicetone [1] and Webcoop [45]. Voicetone has developed systems for telecommunication and pharmaceutical domains. Webcoop is a restriced-domain QA system in the tourism domain.

In the presented approach the knowledge representation and language understanding is based on identifying the service type and the related issue for a user query. This method only requires to identify a high level ontology like service types, and the construction of the issue related feature list is semi-automatic. However, in Voicetone, natual language understanding is based on identifying user intent and the mentioned domain objects in the question. Domain objects are identified by implementing rule-based named entity recognizers. Webcoop uses a complete ontology for the domain with first order rules coded in Prolog. Therefore, the presented approach requires less effort when compared with Voicetone and Webcoop systems.

In Voicetone paraphrase detection is achieved by labelling a large number of training examples. In contrast, Webcoop uses first order logical rules for paraphrase detection. Both of the above mechanisms require specific technical knowledge and skills which may not be available to domain experts. However, the pattern writing process described in this thesis is comparatively simple as it only requires pattern writers to incorporate domain specific information in natural language. In addition, domain experts have been writing patterns for 12 months for the deployed system.

It is also possible to argue that the presented approach has a higher portability between domains, because the introduced ranking algorithm can be find tuned for other domain using the GA optimization mechanism. In contrast, Webcoop need to build the inference engine according to the new first order logic rules of the new domain.

CHAPTER 5 : CONCLUSIONS

This thesis describes a knowledge based method to develop a commercial question answering system. The approach takes into account the syntactic, lexical, and morphological variations by the use of two known input normalization steps and a synonym transduction, which is allowed to vary over the system's knowledge base. Unlike context independent general synonym transductions popular in IR, the pattern based approach takes into account not only the propositional or literal content, but also what sense the user query is made in the context. The simplicity of the pattern writing process that only requires to identify the most common candidate forms of a user query enable a less technically qualified person to maintain the knowledge base, which is a highly desirable requirement in a real industrial environment.

The presented approach to the question answering is based on understanding the user query, which is important to extend the system into more advanced future versions that will completely serve complex user requests via natural language dialogs. Understanding the user's initial input is based on a service type detecting classifier that incorporate prior knowledge and user specific information, and a ranking algorithm that take into account the intent, context, and content components as well as the word order similarity weighted by weighting factors. A genetic algorithm-based method was proposed for regular updating of the optimum values of the weighting factors to adapt to changes in the nature of users' queries over time.

However, the pattern writing approach is prone to individual biases and hence may suffer from lack of robustness mainly caused by inconsistencies when designed by different individuals. The system accuracy can significantly be improved by merely getting rid of certain inconsistencies still present in the patterns without requiring major changes to the algorithm in the system. The stop-word transduction is application dependent and the word list need to be carefully finalized by a user experience (UE) expert.

An evaluation is presented in a real-world system developed using this approach to automate the question and answering process of the real customer helpdesk. Furthermore, the possibilities of porting to other domains are discussed.

5.1 Future Work

As the next step, it is possible to introduce an automated mechanism to calculate the importance of a training instance by taking the domain specific prior information into account. Then, these weights can be used to train a SVM classifier as mentioned in Chapter 3. Dependency parsing can be used to extract important information like subject, object in a training instance in calculation these weights. This automated mechanism can save time of domain experts and reduce errors caused due to inconsistencies of subjective decisions made by humans.

In addition, it is possible to develop more advanced paraphrase detection techniques for restricted domains. These may involve creating a thesaurus for the domain with a defined structure. Existing automatic thesaurus construction mechanisms can be tested in the domain. Furthermore, the approaches used in open-domain paraphrase detection can be used.

CHAPTER 6 : REFERENCES

[1] N. Gupta, G. Tur, D. Hakkani, G. Riccardi and M. Gilbert, *The AT&T Spoken Language Understanding System*. IEEE Trans. Speech Audio Process, Jan. 2006, vol. 14, no. 1, pp. 213-222.

[2] A.M.Pasca, and S.M. Harabagiu, *High Performace Question/ Answering*. New Orleans, Louisiana, USA. SIGIR, 2001.

[3] A.L. Gorin, G. Riccardi, and J.H. Wright, *How may I help you?*, Speech Commun., 1997, vol. 23, pp. 113-127.

[4] E. Bril, S. Dumais, and M. Banko, *An analysis of the AskMSR Question-Answering System*, In MNLP2002.

[5] A. Berger, and V.O. Mittal, *Query - Relevant Summarization using FAQs*, Proceedings of the 38th Annual Meeting on Association for Computational Linguistics - Stroudsburg, PA, USA, 2000.

[6] Apache Software Foundation (May 2012). "*Apache Lucene*" [Online].Available: http://lucene.apache.org/java/docs/index.html.

[7] S. Seneff, *TINA: A natural language system for spoken language applications*, Computat. Linguist., 1992, vol. 18, no.1, pp. 61-86.

[8] W. Ward, and S. Issar, *Recent improvements in the CMU spoken language understanding system*, in Proc.ARPA HLT Workshop, Mar. 1994, pp. 213-216.

[9] J. Dowding, J.M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore,and D. Moran, *Gemini: A natural language system for spoken language understanding*, in Proc. ARPA Workshop on Human Language Technology, Princeton, NJ, Mar. 1993.

[10] S.M. Harabagiu, M.A. Pasca, and S.J. Maiorano, *Experiments with Open-Domain Textual Question Answering*, 18th International Conference on Computational Linguistics - Saarbrucken, Germany, COLING 2000.

[11] G. Miller, *WordNet a lexical database*, Communications of the ACM,1995.

[12] W. Lehnert, *The Process of Question Answering*, Lawrence Erlbaum,1978.

[13] M. Klaus et al., *JGAP - Java Genetic Algorithms and Genetic Programming Package*, URL: http://jgap.sf.net.

[14] C.D. Manning, P. Raghavan, and H. Schtze, *Introduction to Information Retrieval*, Cambridge University Press, NewYork, NY, 2008.

[15] D. Jurafsky, and J.H. Martin, *Speech and Language Processing (2nd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, 2009.

[16] L. Hirschman, and R. Gaizauskas, *Natural language question answering: The view from here*. Journal of Natural Language Engineering, Special Issue on Question Answering, Fall2001.

[17] G. Salton, *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice Hall. 1971.

[18] D. Molla, and J.L. Vicedo, *Question Answering in Restricted Domains: An Overview*, Special Section on Restricted-Domain Question Answering, Association for Computational Linguistics, 2007.

[19] N.F. Noy, and D.L.McGuinness,*Ontology development 101: A guide to creating yourfirstontology*.Technical Report KSL-01-05,Knowledge Systems Laboratory, 2001.

[20] S. Auer, C. Bizer, G. Kobilarov, J.Lehmann, and Zachary Ives, *DBpedia: A Nucleus for a Web of Open Data*, In 6th Int'l Semantic Web Conference, Busan, Korea, 2007.

[21] (May 2012).*Wikipedia*, [Online].Available:http://www.wikipedia.org/

[22] W Roush, *The story of Siri, from birth at Sri to acquisition by Apple: Virtual personal assistants go mobile*. Xconomy, June 2010.

[23] (May 2012). *TREC*, [Online]. Available:http://trec.nist.gov

[24] (May 2012). *Wolfram Alpha*,[Online]. Available:http://www.wolframalpha.com

[25] (May 2012). *IBM Watson*,[Online]. Available:http://www.aaai.org/Magazine/Watson/watson.php

[26] (May 2012). *True Knowledge Evi*,[Online].Available:http://www.evi.com/

[27] (May 2012). *Google*,[Online].Available:http://www.google.com/

[28] B. F.,Green, A. K.Wolf, C. Chomsk y, andK. Laughery, *Baseball: An* automatic *question answerer*. In Proceedings Western Computing Conference, 1961, vol 19,pp. 219–224.

[29] WA Woods, *Lunar rocks in natural English: Explorations in natural language question answering*. Linguistic structures processing, North-Holland, New York. 1977.

[30] W. Robert, N.C. David, M. Luria, J. Martin, J. Mayfield, and D. Wu, *The Berkeley Unix Consultant project*. Computational Linguistics, 1994,14(4),pp.35–84.

[31] (May 2012). *Paraphase Detection*, ACL web,[Online].

Available:http://aclweb.org/aclwiki/index.php?title=Paraphrase_Identification_(State_of_the_art).

[32] S. Fernando, and M. Stevenson, *A semantic similarity approach to paraphrase detection*, Computational Linguistics UK.11th Annual Research Colloquium. 2008.

[33] A. Finch, Y.S. Hwang, and E. Sumita, *Using machine translation evaluation techniques to determine sentence-level semantic equivalence*, "Proceedings of the Third International Workshop on Paraphrasing", Jeju Island, South Korea, 2005,pp. 17-24.

[34] A. Islam, and D. Inkpen, *Semantic similarity of short texts*, Proceedings of the International Conference on Recent Advances in Natural Language Processing, Borovets, Bulgaria, 2007, pp. 291-297.

[35] Z. Kozareva, and A. Montoyo, *Paraphrase identification on the basis of supervised machine learning techniques*, Advances in Natural Language Processing: 5th International Conference on NLP.Turku, Finland, 2006,pp. 524-533.

[36] R. Mihalcea, C. Corley, and C. Strapparava, *Corpus-based and knowledge-based measures of text semantic similarity*, Proceedings of the National Conference on Artificial Intelligence, Boston, Massachusetts, 2006, pp. 775-780.

[37] L. Qiu, M.Y. Kan, and T.S. Chua, *Paraphrase recognition via dissimilarity significance classification*, Proceedings of the Conference on Empirical Methods in Natural Language Processing.2006, pp. 18-26.

[38] V. Rus, P.M. McCarthy, M.C. Lintean, D.S. McNamara, and A.C.Graesser, *Paraphrase identification with lexico-syntactic graph subsumption*, FLAIRS 2008, pp. 201-206.

[39] R. Socher, E.H. Huang, J. Pennington, A.Y. Ng, and C.D. Manning, *Dynamic pooling and unfolding recursive auto encoders for paraphrase detection*, "Advances in Neural Information Processing Systems 24".2011.

[40] S. Wan, M. Dras, R. Dale, and C. Paris, *Using dependency-based features to take the "para-farce" out of paraphrase*, Proceedings of the Australasian Language Technology Workshop (ALTW 2006), pp. 131-138.

[41] D. Demner-Fushman, and J. Lin, *Knowledge extraction for clinical question answering: Preliminary results.* In Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence, Pittsburgh, PA, 2005,pp. 1–9.

[42] E.T.K. Sang, G.Bouma, and M. Rijke, *Developing offline strategies for answering medical questions.* In Workshop on Question Answering in Restricted

Domains.20th National Conference on Artificial Intelligence (AAAI-05), Pittsburgh, PA, 2005, pp. 41–45.

[43] Y. Niu, and G. Hirst, *Analysis of semantic classes in medical text for question answering.* In Workshop on Question Answering in Restricted Domains. 42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004), Barcelona, Spain, 2004, pp. 54–61.

[44] H. Yu, C. Sable, and H.R. Zhu. *Classifying medical questions based on an Evidence Taxonomy*. In Workshop on Question Answering in Restricted Domains. 20th National Conference on Artificial Intelligence(AAAI-05), Pittsburgh, PA, 2005,pp. 27–35.

[45]F. Benamara, *Cooperative question answering in restricted domains: The WEBCOOP Experiments*. In Workshop on Question Answering in Restricted Domains.42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004), Barcelona, Spain, 2004, pp. 31–38.

[46] K. Barker, V.K. Chaudhri, S.Y. Chaw, P. Clark, J. Fan, D. Israel, S. Mishra, B.W. Porter, P. Romero, D. Tecuci, and P.Z. Yeh, *A Question-answering system for AP chemistry: Assessing KR&R technologies. In Principles of Knowledge Representation and Reasoning:* Proceedings of the Ninth International Conference, Whistler, Canada, 2004,pp. 488–497.

[47] D. Lenat, and R.V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project.* Addison-Wesley,1990.

[48] B. Porter, J. Lester, K. Murray, K. Pittman, A. Souther, L. Acker, and T. Jones, *AI research in the context of a multifunctional knowledge base: The botany knowledge base project.* Technical Report, 1988, Department of Computer Sciences, University of Texas at Austin.

[49] P. Cohen, R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Easter, D. Gunning, and M. Burke, *The DARPA high performance knowledge bases project.* AI Magazine, 1988, 19(4), pp.25–49.

[50] R. Schrag, M. Pool, V. Chaudhri, R.C. Kahlert, J. Powers, P. Cohen, J. Fitzgerald, and S. Mishra, *Experimental evaluation of subject matter expert-oriented knowledge base authoring tools.* In Proceedings of the PerMIS Workshop: Measuring the Performance and Intelligence of Systems, Gaithersburg, MD. 2002, pp. 272–279.

[51] R. Zajac, *Towards ontological question answering*. In Proceedings of ACL,Workshop on Open Domain QA, Toulouse,2001.

[52] D. Klein and C.D. Manning, *Conditional structure versus conditional estimation in NLP models*, EMNLP '02 Proceedings of the ACL-02 conference on Empirical

methods in natural language processing. 2002. vol: 10, pp. 9-1.

[53] (May 2012). *Microsoft Research Paraphrase Corpus*, [Online].
Available: http://research.microsoft.com/en-us/downloads/607D14D9-20CD-47E3-85BC-A2F65CD28042/default.aspx.

[54] C. Leacock, and M.S. Chodorow, *Combining local context and WordNet similarity for word sense identification.* In Fellbaum, C. (Ed.), Wordnet: An Electronic Lexical Database, MIT Press,1998,pp. 265–283.

[55] P. Resnik, *Using information content to evaluate semantic similarity in a taxanomy.* In International Joint Conference for Artificial Intelligence (IJCAI-95), 1995,pp. 448–453.

[56] Lin, *An information-theoretic definition of similarity*. San Francisco, In *ICML 1998*, pp. 296–304.

[57] J.J. Jiang, and D.W. Conrath, *Semantic similarity based on corpus statistics and lexical taxonomy*. In ROCLINGX, Taiwan. 1997.

[58] S. Banerjee, and T. Pedersen, *Extended gloss overlaps as a measure of semantic relatedness*. In *IJCAI 2003*, pp. 805–810.

[59] F. Lauer, and G. Bloch, *Incorporating prior knowledge in support vector machines for classification: A review*. Neurocomputing vol. 71, Issues 7–9, March 2008, pp. 1578–1594.

[60] V. Vapnik, *Statistical Learning Theory*. John Wiley and Sons, Inc., New York, 1998.

[61] R.E. Schapire, M. Rochery, M. Rahim, and N. Gupta, *Boosting With Prior Knowledge for Call Classification*, IEEE Transactions on Speech and Audio Processing, vol. 13, No 2, March 2005.

[62] B. Zadrozny, J. Langford, and N. Abe,*Cost-Sensitive Learning by Cost-Proportionate Example Weighting*, ICDM 2003.

[63] (May 2012). *LibSVM*, [Online].
Available:http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[64] C.W Hsu, C.C. Chang, and C.J. Lin, *A Practical Guide to Support Vector Classification.*

[65] (May 2012). *Standford Dependency Parser*
Available: http://nlp.stanford.edu/software/lex-parser.shtml

[66] G. Riccardi, and A.L. Gorin, "*Spoken language adaptation over time and state in a natural spoken dialog system*" IEEE Trans. Speech AudioProcess,2000, vol. 8, no. 1, pp. 3–10.

[67] A. Jönsson, and N. Dahlbäack, "*Talking to a computer is not like talking to your best friend*," in Proc. 1st Scandinavian Conf. Artificial Intelligence, Tromsø, France, 1988.

[68] A.R. Diekema, O.Yilmazel, and E.D. Liddy. "*Evaluation of restricted domain question-answering systems*". In Workshop on Question Answering in Restricted Domains.42nd Annual Meeting of the Association for Computational Linguistics (ACL-2004),Barcelona, Spain,2004.pp. 2–7.

[69] G.D. Fabbrizio, D. Dutton, N. Gupta, B. Hollister, M. Rahim, G. Riccardi, R. Schapire, and J. Schroeter, "*AT&T help desk*" in Proc. Int. Conf. Spoken Language Processing, Denver, CO, Sep. 2002.

[70] P. Haffner, G. Tur, and J. Wright, "*Optimizing SVM's for complex call classification"* in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Hong Kong, 2003.

CHAPTER 7 : APPENDIX A - RULE BASED CLASSIFIER

It is important to cluster the knowledge base into service-type based clusters due to the following reasons

- Most common queries are often common to most of the service types but the answers to those queries
- User seems to sufficiently explain the service issue but the service type in the initial input

It will be shown in the following sections that sub-dialog models to collect clarifying information when service type is missing in the user's initial input can be developed strongly based on a service type-based classification rule base.

Let the user's initial input, after input normalization, be defined by the vector

$$\mathbf{x} = [\mathrm{w}_1, \mathrm{w}_2, \mathrm{w}_3, \cdots, \mathrm{w}_\mathrm{i}, \cdots, \mathrm{w}_\mathrm{n}]$$

where, $\mathrm{w}_\mathrm{i}; \mathrm{i} = 1, 2, 3, \cdots, \mathrm{n}$ are the words in the input.

7.1 Major Services

For convenience, the broadband services and standard telephony services are considered as major services. For reasons given in Section 7.2, in here, the terms salient to the auxiliary services such as VoIP/MoIP, hosting services, VPN, and messaging services (email, SMS, MMS, email to fax, and so forth) are ignored. Major services may further be classified as wireless and wired (copper and fiber) services.

First, the field vectors are defined. They consist of application specific field vocabulary words and phrases. The internet or broadband field vector is given by

$$\boldsymbol{w}_{Internet} = \boldsymbol{w}_{Internet1} \cup \boldsymbol{w}_{Internet2}$$

where,

$$w_{Internet1} = [internet, broadband, www, authenticate, auth, bandwidth, bitrate, bit\ rate, datarate, bps, speed, \ dataWiFi, WiFi, modem, sync, ppp, splitter, router, relocate, relocation]$$

$$\boldsymbol{w}_{Internet2} = [data, upload, download, IP, web, website, webpage, URL, brows, browser, domain, FTP, search\ engine, TCPIP, ethernet, telnet, LAN, WAN, PC, computer, laptop, tablet, iPad, notebook, webcam, webcast, codec]$$

Internet field vector is broken up into two to make it possible to write better sub-dialog models. For instance, if $w_i \in \boldsymbol{w}_{Internet1}$only, then it is often required to know the type of broadband service to answer the query. Queries with $w_i \in \boldsymbol{w}_{Internet2}$, may be directed to a Q-A common pool "Internet".

Telephony field vector is

$$\boldsymbol{w}_{Telephony} = [telephony, telephone, phone, call, voice, ring, dial, tone, handset, number, caller, talk, IDD, CallBack, SurePage, leave\ message, answering\ machine, preselect, pre\ select, preselection, override, long\ distance]$$

7.1.1 Major Wireless Services

The two major wireless services are wireless internet (wireless broadband) and wireless telephony. Wireless-service field vector is

$$\boldsymbol{w}_{WirelessService} = [wireless, mobile, cell, cellular, handphone, hand\ phone, cellphone, smartphone, smart\ phone, iPhone, PDA, SIM, USIM, GPRS, GPRS2G, GSM, reception, signal, roaming, antenna, PUK, PUC]$$

Wireless-broadband field vector is

$\boldsymbol{w}_{WirelessBB}$
$= [HSPA, HSDPA, WCDMA, GPRS3G, GPRS\ 3G, dongle, USB\ modem, USB\ stick,$
$WAP, APN, tethering]$

Classification rules for wireless services are

- Mobile Broadband: $\boldsymbol{w}_{WirelessBB} \vee (\boldsymbol{w}_{WirelessService} \wedge \boldsymbol{w}_{Internet})$
- Mobile Telephony: $(\boldsymbol{w}_{WirelessService} \wedge \boldsymbol{w}_{Telephony}) \wedge \neg\boldsymbol{w}_{Internet} \wedge \neg\boldsymbol{w}_{WirelessBB}$

For convenience, user inputs with $w_i \in \boldsymbol{w}_{WirelessService}$ may also be considered to be belonging to Mobile Telephony category. They may include the queries on the mobile carriers, mobile phones, roaming etc.

7.1.2 Major Wired Services

Major wired services are the two wired broadband services, ADSL and fiber optics, and wired telephony.

Copper-line field vector is

$\boldsymbol{w}_{CopperLine}$
$= [copper, landline, land\ line, wireline, fixed\ line, wire, receiver, PSTN,$
$landphone, land\ phone, home\ phone]$

ADSL field vector is

$\boldsymbol{w}_{ADSL}$
$= [ADSL, ADSL+, ADSL1, ADSL2, ADSL2+, DSL, HSDSL, SHDSL, microfilter,$
$micro\ filter]$

Fiber field vector is

$\boldsymbol{w}_{Fiber}$
$= [fiber, fibre, optic, optical, NBN, NBNCo, Opticomm, ONT, OLT, build\ drop]$

Classification rules for wired services are

- Fiber Broadband: $\boldsymbol{w}_{Fiber}$
- ADSL Broadband: $\boldsymbol{w}_{ADSL} \vee (\boldsymbol{w}_{CopperLine} \wedge \boldsymbol{w}_{Internet})$
- Wired Telephony: $(\boldsymbol{w}_{CopperLine} \wedge \boldsymbol{w}_{Telephony}) \wedge \neg\boldsymbol{w}_{Internet} \wedge \neg\boldsymbol{w}_{WiredBB}$

where, $\boldsymbol{w}_{WiredBB} \in (\boldsymbol{w}_{ADSL} \cup \boldsymbol{w}_{Fiber})$. For convenience, any user inputs with $w_i \in \boldsymbol{w}_{CopperLine}$ may also be considered to be belonging to Wired Telephony category.

7.2 Auxiliary Services

VoIP/MoIP, VPN, hosting, and messaging services are considered as auxiliary services. The messaging services considered here are: SMS, MMS, IMS, email, voicemail, email to SMS, email to fax, and FoIP. The knowledge base of the question answering system will have two main modules, namely, the major-services-module and the auxiliary-services-module. User queries that contain salient terms related to the auxiliary services, irrespective of the rest of the content words and phrases, will be directed to the auxiliary services module and served there. Hence, these questions will not be passed on to the major services module.

Field vector, $\boldsymbol{w}_{Messaging}$ consists of the terms that are common to most of the messaging services considered here.

$$\boldsymbol{w}_{Messaging} = [address\ book, inbox, outbox, message\ box, spam, send\ message, receive\ message, retrieve\ message, delete\ message, listen\ message, store\ message, save\ message, delivery\ report]$$

The SMS, email, voicemail, and facsimile field vectors, respectively, are defined as follows

$$\boldsymbol{w}_{SMS} = [SMS, SMSs, SMSes, short\ message, text\ message, exeSMS, webSMS, MMS, MMSs, MMSes, multimedia\ message, multi\ media\ message, multipart\ message, IMS, IMSs, virtual\ mobile\ number, VMN, VMNs, texting]$$

$$\boldsymbol{w}_{Email} = [email, mail, exemail, webmail, SMTP, IMAP]$$

$\boldsymbol{w}_{Voicemail} = [voicemail, voice\ mail, voice\ message, VMS, VMSs, VMSes]$

$\boldsymbol{w}_{Fax} = [fax, facsimile, FoIP]$

Classification rule for messaging services are

- Messaging Services: $(\boldsymbol{w}_{Messaging} \vee \boldsymbol{w}_{SMS} \vee \boldsymbol{w}_{Email} \vee \boldsymbol{w}_{Voicemail} \vee \boldsymbol{w}_{Fax}) \wedge \neg\boldsymbol{w}_{Hosting}$

where, $\boldsymbol{w}_{Hosting}$ is the hosting services field vector that will be defined shortly. Field vectors for VoIP and MoIP services are given below

$$\boldsymbol{w}_{VoIP} = [VoIP, IP\ telephony, voice\ over\ IP, internet\ voice, internet\ telephony, broadband\ telephony, voice\ over\ BB, VoBB, ATA, telephone\ adapter, phone\ adapter, DID, virtual\ telephone\ number, virtual\ phone\ number]$$

$\boldsymbol{w}_{MoIP} = [MoIP, mVoIP, mobile\ VoIP, API]$

Classification rule for VoIP service is

- VoIP Service: $\boldsymbol{w}_{VoIP}$
- VoIP Service (possibility): $\boldsymbol{w}_{Internet} \wedge \boldsymbol{w}_{Telephony}$

Possibilities are the scenarios that need to be verified through a brief dialog with the user.

Classification rule for MoIP service is

- MoIP Service: $\boldsymbol{w}_{MoIP}$
- MoIP Service (possibilities): $\boldsymbol{w}_{VoIP} \wedge (\boldsymbol{w}_{WirelessService} \vee \boldsymbol{w}_{SMS} \vee \boldsymbol{w}_{Email})$

Classification rules for hosting and VPN services, respectively, are

- Hosting Service: $\boldsymbol{w}_{Hosting}$
- VPN Service: $\boldsymbol{w}_{VPN}$

where,

$\boldsymbol{w}_{Hosting} = [webhosting, webspace, web\ space, DNS, CMS, domain \wedge host\ ,$
$create \wedge (web \vee website \vee webpage \vee homepage \vee home\ page), server]$
and $\boldsymbol{w}_{VPN} = [VPN, (virtual \vee private) \wedge (network \vee LAN)]$.

7.3 Control Strategy

Functional block diagram of the classifier control strategy is given in Figure 7-1. The main blocks therein are

- A pool of Q-A pairs common to all services
- Auxiliary services module that contains Q-A pairs specific to auxiliary services
- Main services classifier and sub-dialogs
- Main services module that contains Q-A pairs specific to the main services and unrelated to the auxiliary services

Once a user input has been received, first of all, the algorithm checks whether the query is on any of the service issues that are common to all the services. The rule-base to identify the queries that fall into this category is based purely on the terms salient to service issues and independent of the service type as follows.

- Service issues common to all services: $\boldsymbol{w}_{CommonIssues}$

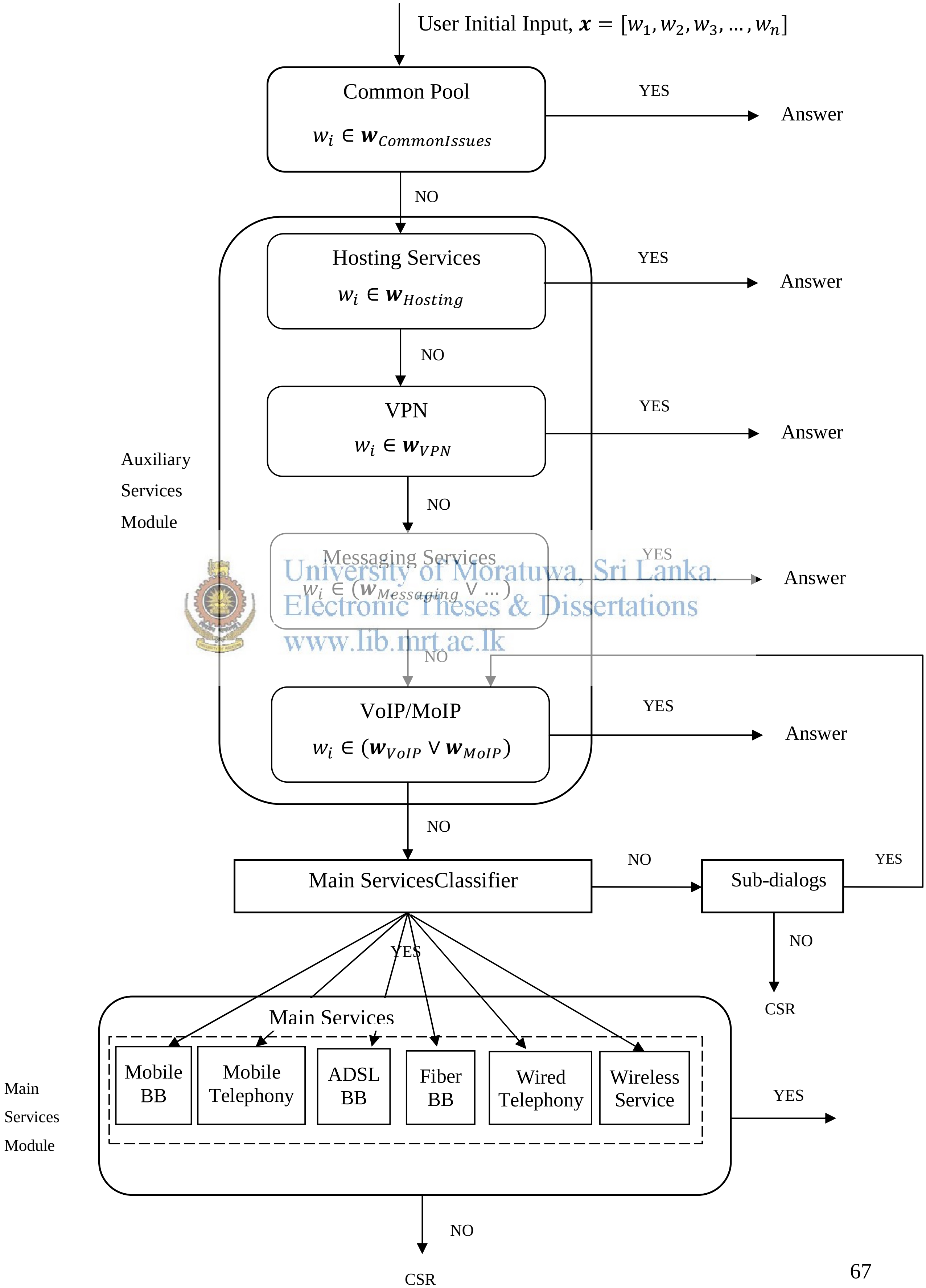
where, algorithm marks questions as common, if one or more rules mentioned below is satisfied.

R1: What ∧ billing ∧ (cycle ∨ period ∨ method)
R2: (how ∨ when) ∧ (bill ∨ charge)
R3: (pro rata ∨ prorata) ∧ (calculate ∨ calculation)
R4: Administration ∧ (fee ∨ charge)
R5: (service ∨ credit card) ∧ surcharge
R6: (want ∨ access ∨ read ∨ understand ∨ receive ∨ retrieve ∨ obtain ∨ previous ∨ old) ∧ (bill ∨ invoice)
R7: (mode ∨ method ∨ option) ∧ payment
R8: AMEX card
R9: (change ∨ edit ∨ update ∨ modify) ∧ (payment ∨ account) ∧ (details ∨ information)

R10: (adjust ∨ change) ∧ billing ∧ date
R11: (delay ∨ late ∨ overdue ∨ outstanding ∨ fail ∨ reject ∨ decline ∨ dishonor ∨ overdrawn ∨ unbilled) ∧ (payment ∨ invoice ∨ bill ∨ charge ∨ fee)
R12: Interim
R13: Excess usage
R14: Insufficient ∧ (fund ∨ money)
R15: Refund ∨ credit account back ∨ credit money back ∨ return money
R16: (merge ∨ combine ∨ connect) ∧ (account ∨ ID ∨ IDs ∨ invoice ∨ bill)
R17: (master ∨ one ∨ single) ∧ (account ∨ invoice ∨ bill)
R18: (early ∨ contract) ∧ (cancel ∨ cancellation ∨ terminate ∨ termination) ∧ (fee ∨ charge)∨ETC
R19: (contact ∨ speak) ∧ (billing ∨ sales ∨ support ∨ service ∨ department ∨ section ∨ division ∨ exetel ∨ details)
R20: (transfer ∨ change) ∧ ownership
R21: (How ∨ want ∨ need) ∧ (cancel ∨ terminate ∨ unsubscribe ∨ quit ∨ …) ∧ (ADSL ∨ fiber ∨ broadband ∨ BB ∨ mobile ∨ exemail ∨ mail ∨ email ∨ hosting ∨ VoIP ∨ SMS ∨ service ∨ connection ∨ internet ∨ telephone ∨ plan)

Quite in contrast to the main services classifier, the classification rules in the auxiliary service module are checked sequentially as shown in Figure 7-1. Handling auxiliary services related queries separately in a separate module simplifies the classification problem. For instance, isolating the messaging services related questions prevents them from being distributed across the five main services clusters. This simplifies the overall classification problem as well as the development process of sub-dialog models.

Figure 7-1: Control Strategy of the Classifier



7.4 Sub-dialog Models

Experience has shown that the user, in the initial input, often describes the service related issue sufficiently but the underlining service type, which is often required to generate the most appropriate answer. To overcome this problem, sub-dialogs are required to collect clarifying information from the user. This section discusses how such sub-dialogs may be generated based on the classification rules and the field vectors introduced in previous sections.

Note that the sub-dialogs are activated if and only if the user initial input does not satisfy any of the classification.

Sub-dialog 1: $w_i \in \boldsymbol{w}_{Internet1}$

Sub-dialog 1-1: $w_i \in [\mathrm{sync}, \mathrm{ppp}, \mathrm{splitter}, \mathrm{router}, \mathrm{relocate}, \mathrm{relocation}]$ (***Note***: Most probably on a subscribed service; it can either be ADSL or Fiber)

Cal: Are you referring to an ADSL (copper line) or fiber internet connection?

User: (ADSL/copper)/Fiber → ADSL/Fiber

Wireless/mobile → Mobile BB

Default → ADSL

Sub-dialog 1-2: $w_i \in [\mathrm{modem}]$ (***Note***: Whether it's on a subscribed service or otherwise is immaterial; Most probably ADSL. It can remotely be Wireless BB or Fiber)

Cal: Are you referring to an ADSL modem (wired copper line) or USB stick used for Wireless Internet?

User: ADSL/wired/copper/land → ADSL

Wireless/dongle/USB/stick → Mobile BB

Fiber/fibre/ONT → Fiber

Default → ADSL

Sub-dialog1-3:

$w_i \in$ [internet, broadband, www, authenticate, auth, bandwidth, bitrate, bit rate, datarate, data rate, bps, speed, WiFi, Wi − Fi] (***Note***: It can be on a subscribed or not both; Most probably ADSL. It can be any BB service)

Cal: Kindly let me know the type of internet service you are referring to: ADSL (wired: Copper line), Fiber (wired: Fiber link), or Wireless Internet

User: fiber → fiber

Wireless/mobile → Mobile BB

ADSL/copper → ADSL

Default → ADSL

Sub-dialog 2: $w_i \in \boldsymbol{w}_{Telephony}$

Cal: Are you referring to standard mobile telephony, landline telephony, or VoIP?

User: Mobile/Landline/VoIP → Mobile Telephony/Wired Telephony/VoIP

Sub-dialog 3: $w_i \notin \boldsymbol{w}_{ServiceType}$ where $\boldsymbol{w}_{ServiceType}$ is the vector of all the service-type-related salient terms.

Cal: Please specify the service type you are referring to: Wireless Internet, Mobile Telephony, Landline Telephony, ADSL Broadband, Fiber Broadband, Internet Telephony (VoIP)

User: Wireless Internet/Mobile Telephony/Landline Telephony/ADSL Broadband/Fiber Broadband/Internet Telephony (VoIP) → Mobile BB/Mobile Telephony/Wired Telephony/ADSL BB/Fiber BB/VoIP

Internet → Sub-dialog 1

Telephony → Sub-dialog 2

Sub-dialog 4: VoIP possibility $w_i \in \boldsymbol{w}_{Internet} \wedge \boldsymbol{w}_{Telephony}$

Cal: Are you referring to Internet Telephony (VoIP) service?

User: Yes/MoIP → VoIP/MoIP

W1 w2 w3 …. → Return to the Main Services Classifier