INTERACTIVE QUESTION ANSWERING USING FRAME-BASED KNOWLEDGE REPRESENTATION

EMAD GOHARI BOROUJERDI

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

MASTER OF SCIENCE

GRADUATE PROGRAM IN ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

JULY 2018

Abstract

A framework for building an interactive question answering (IQA) system is proposed based on a frame-based approach in dialogue systems. This method uses natural language processing techniques such as semantic role labeling for building a knowledge base from a set of question-answer pairs. The proposed IQA system uses this knowledge base to represent its set of question-answer pairs. The knowledge base consists of frame representations which are created from the questions in the set of question-answer pairs. A frame representation of a question contains slot-value pairs presenting the question's semantic. A frame-based representation enables a question answering system to engage in dialogue interactions with a user. The purpose of the dialogue is to better capture the user's intention and find the answer from the knowledge base which best matches the user's information need. A procedure for extracting slots (attributes) and their values for representing questions in the knowledge base is proposed. In addition, a dialogue management system is presented to interactively answer a user's question based on the generated knowledge base. Our framework was tested on datasets in the domain of car manuals. We conducted experiments and user studies for evaluation of the generated IQA system. Our evaluation results, based on car manual questions, show the effectiveness of the proposed framework for knowledge base generation and better performance in question answering compared to different baseline methods. Moreover, the deployment of this method effectively reduced the manual effort for knowledge base generation.

Dedicated to

my dear parents for everything,

and my dear sister.

Acknowledgements

First of all, I would like to thank my supervisor, Dr. Aijun An, for her endless patience, guidance, and support during the process of writing my thesis. I would like to thank my committee member, Dr. Jimmy Huang, for his helpful advice. Also, I would like to show my gratitude for Dr. Nick Cercone (my former supervisor), an excellent professor and pioneer in the field of data mining, who unfortunately passed away during my Master's studies.

This work was part of a collaborative project between York University and iNAGO Inc. and financially supported by the BRAIN Alliance and iNAGO Inc. I am thankful to the team at iNAGO Inc. for creating the problem to be solved in this thesis and for providing the datasets, guidance, and feedback for the research.

The members of the BRAIN lab have contributed to my graduate career and personal life at York University. I am grateful for their friendships as well as collaborations. I am also thankful to the administrative and technical staff of our department for their finest support.

Lastly, my deep gratitude goes to my family for always being there on my side and offering their tremendous support, encouragement, and love in every aspect of my life.

Table of Contents

List of Tables

List of Figures

1 Introduction

The goal of Question Answering (QA) systems is to understand natural language questions and use this understanding to provide a concise answer from their information sources. In this manner, QA is different from keyword search regarding two main aspects. First, a question conveys information beyond its list of keywords. Second, a QA system targets answering a question rather than just returning a list of links to the relevant documents [60].

1.1 Approaches to Question Answering

Two major paradigms to QA are Information Retrieval (IR)-based QA and Knowledge-based QA [27].

In the IR-based QA, the information source is the available text on the web or collections of documents from a specific domain, e.g., PubMed. The research related to the former type of information source is focused on Open-domain QA, while the latter focuses on Restricted-domain QA problems [35]. The categorization of QA

systems into open domain and restricted domain is also applicable to the knowledge-based QA systems. One of the well-established methods in IR for building a QA system is as follows: the related sentences to the question are ranked and extracted from the documents using IR techniques and summarized into a passage as the answer [29, 28].

Knowledge-based QA is developed from research in AI and building smart systems. In knowledge-based QA, the knowledge is encoded into a database which is used as the information source of the QA system. It might be a complex database (such as the GeoQuery database of questions on U.S Geography [64]), or a database of simple relations (for example, triple stores like Freebase [9] or DBpedia [2]). The querying approach in knowledge-based QA is to first build a semantic representation of the question, and then use this representation for querying the database to find the answer. Depending on the type of database, logical (e.g., predicate calculus), SPARQL, or SQL queries can be used for searching the knowledge base.

Recently, neural network approaches, especially Deep Learning methods, are showing significant improvements on many NLP-related tasks including text generation [10, 65], machine translation [3], and question answering [50, 47, 62]. The neural network approach in IR-based QA enables effective modelling of word & sentence similarities to enhance text ranking and answer selection in QA systems. In question retrieval for Community Question Answering (CQA), learning distributed

word representations can be enhanced by simultaneously embedding questions' category metadata [66]. A Collaborative Adversarial Network (CAN) architecture using a common feature extractor [13] and a Recurrent Neural Network (RNN) with gating mechanism for context alignment [12] can be used for learning sentence similarities from pairs of similar sentences as the training data. Another possible way to use deep learning methods in QA is formulating the problem as a combination of the question generation and question answering problems [53]. Models based on Generative Adversarial Networks (GAN) have shown promising results for solving the combined problem of question generation & answering [50, 62].

1.1.1 Using Machine Learning vs. Multi-component Methods in QA Systems

Addressing natural language related problems like QA without using conventional natural language specific features have become popular recently. This has been made possible following the success in deploying machine learning methods and rapid improvements in computational power for training such models. A good example is the impact of neural network models on the area of Natural Language Processing in the past few years. This data-driven approach to QA needs a large volume of labeled data (e.g., question-answer pairs) for training, especially for neural networks. In many real-world situations, domain-specific labeled data is rare

and thus training these models with the desired accuracy is very difficult.

In this work, we aim to develop an approach that does not rely on labeled training data. In addition, unlike neural networks, we would like our approach to be more transparent and easy to understand. For achieving this, our approach is based on a stack of Natural Language Processing (NLP) components which allows debugging and enhancement for the desired results.

An important method of QA categorization is using the question type: Factoid and non-factoid QA systems. Factoid questions are recognized by their specific words, such as *who, what* and *where*, and their answers are simple facts or named entities expressed in a single word or a short phrase. For example, "Who is Canada's Prime Minister?". A large volume of research is focused on answering factoid questions. Many of the advances in Factoid QA comes from the research on the QA track of Text REtrieval Conference (TREC) [56, 20].

Non-factoid questions require more complex reasoning and usually longer answers. Non-factoid question types include questions about definition, causation, reason, etc. For example, "What is the Brake Warning Light?" is a *definition* question and "Why does the Brake Warning Light flash?" is a *reason* question in the domain of car manuals.

Recently non-factoid QA research has become a hot topic, mainly due to more availability of datasets for these types of questions. [52]. The user-generated content

from Community Q&A (CQA) websites (such as *Yahoo! Answers*, Quora, and StackExchange) enabled research in non-factoid QA. In these websites, users answer questions posed by other users, and the best answer is up-voted by the participants in the thread. An overview of recent research on non-factoid QA will be described in chapter 2.

1.1.2 Interactive interface for QA Systems

Since the structure of information in a QA system is hidden from the user, often their initial question is not perfectly complete and well-formed [60]. A user might not be confident about how or what terms to use for asking the question, or want to explore the available information from the system without a specific request in mind. For example, a user would like to find a restaurant, but with no special type of food in mind. So the user may simply ask: "Any restaurants nearby?" In such situations, the user can benefit from a conversational interface capable of answering incomplete and vague questions through a series of dialogue interactions.

From the IR perspective, the common gap between the initial and the perfect query leads to a query cycle, in which a user modifies the query multiple times to get the desired answer. It is preferred that the IR system assists the search process with an interactive mechanism. The interactive mechanism gives the QA system the ability to remember the context in case of anaphora or coreference in

subsequent queries, and suggest next actions towards user's information need [23]. Following the above example, the system may ask "what type of food would you like to have?"

The goal of Interactive Question Answering (IQA) is to assist the user to better specify the question through interactive communication, such as conversational interfaces. The system interacts with the user through either a textual or vocal interface. The interactive input can involve selecting from a list, or any text or voice input. The free-form input is mapped to the relevant options from the database using Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) components of the system.

IQA is also related to a large portion of research in Dialogue Systems. While it is fairly new to the QA and IR communities, IQA has a long history in the research of dialogue systems [60]. For example, QA systems in Intelligent Tutoring assess the student's knowledge and correct their errors through spoken interactions [31]. A major role of dialogue systems, especially in industrial applications, is for building task-oriented systems capable of performing a special task from information provided during a dialogue. The goal in a task-oriented dialogue system is information provision from special databases or accomplishing tasks such as booking a flight/hotels, finding a restaurant, routing customer calls to related staff, etc.

1.2 Motivation and Challenges

This work is done in collaboration with an industrial partner, iNAGO Inc., to improve their current workflow for building a knowledge base for IQA. iNAGO is specialized in enabling a human-like conversational and intelligent assistant on any computer device ranging from PCs to robots. One of iNAGO's key products is an interactive question-answering system that converses with users to obtain the most appropriate answers to their questions based on a domain-specific knowledge base of questions and answers. Currently, this knowledge base is created manually by first creating a set of questions and answers from domain documents and then converting them into a knowledge base of a special format that supports interactive question answering. However, creating a knowledge base from textual sources, e.g., user manuals, requires significant human time, effort and cost.

In collaboration with iNAGO, we have been working on automating the knowledge base generation process by:

1. Developing an approach to automatically generate a set of question-answer pairs (which we refer to as the *q-a set*) from text documents.

2. Developing an approach to automatically converting a q-a set into a knowledge base of a format that enables interactive question answering. In this thesis, we focus on the second task, which is automatic knowledge base generation

from a q-a set.

Figure 1.1 shows iNAGO's current workflow for building a domain-specific knowledge base for IQA. The process involves domain term identification, term organization into a conceptual hierarchy, adding synonyms into the terms, determining attributes for representing questions, identifying attributes and their values for a question, adding attribute-value representations for q-a pairs into the knowledge base, and updating the attribute mapping rules (which enable the query understanding component in IQA to understand different ways a user may ask a question). The resulting knowledge base consists of a set of question-answer pairs, where the question in each QA is represented by a set of attribute-value pairs.

To illustrate the attribute-value representation, consider the following question "What does this 'check engine' light mean on my speedometer?". This question can be represented by the following attribute-value pairs 'Question Type'='information', 'Term Requiring Information'='check engine', 'Indicator Light'='check engine light', 'Cabin Control'='speedometer', 'Location'='on my speedometer'. The first part of the pairs ('Question Type', 'Term Requiring Information', etc.) are the terms showing the attributes. The second part of the pairs ('check engine', 'check engine light', etc.) are the terms from the question, which shows the values.

The current workflow has many subjective decisions and needs lots of manual work. We would like to propose a systematic solution to reduce the manual effort

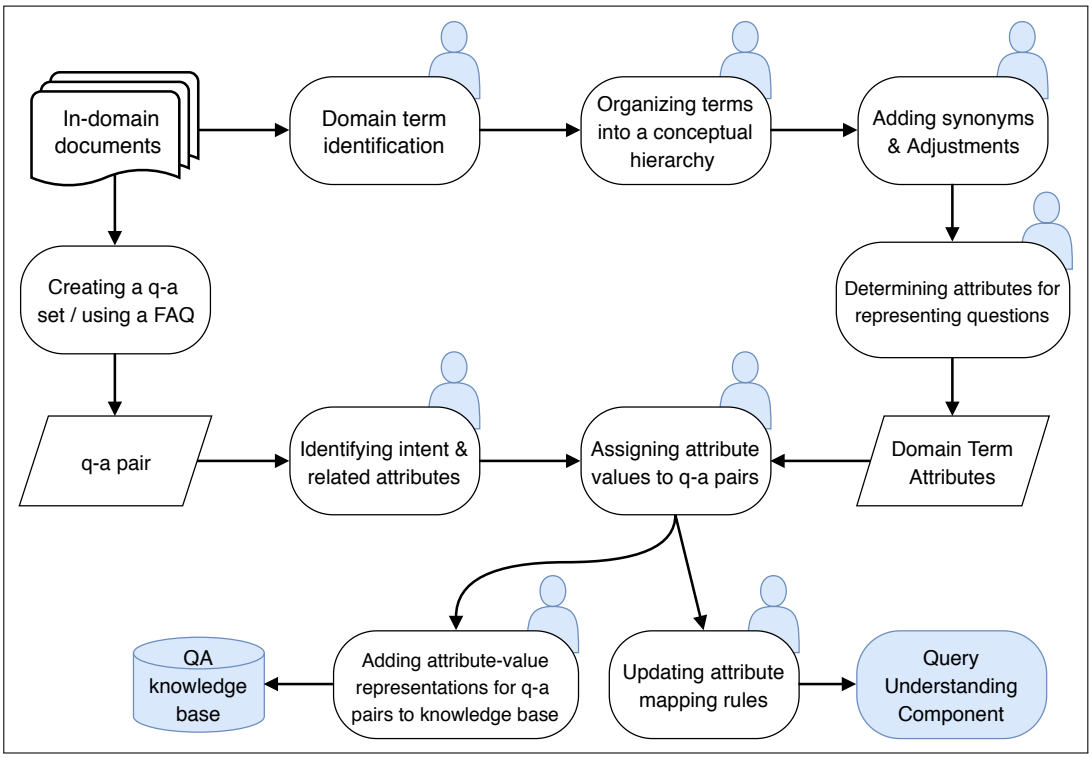


Figure 1.1: The manual knowledge base generation framework at iNAGO Inc.

and subjective decisions required for building the knowledge base. The proposed approach for building knowledge base should be applicable to information sources from any domain.

We aim at exploiting the available linguistic resources and techniques for semantic understanding. These techniques are used to build a framework with transparent and human-understandable processing steps. Current end-to-end neural network models do not offer the required transparency. Also, the proposed method allows the expert knowledge and domain-specific information to be encoded as rules in the

IQA system.

This work is focused on building a QA system for a set of questions and answers (q-a set) in a specific domain. It is directly applicable to cases where a q-a set for a specific domain exists. For example, this could be applicable to any products or services which have a collection of Frequently Asked Questions (FAQ) [11]. Community Q&A websites like StackExchange, Quora, and Yahoo! Answers provide large databases of manually generated q-a pairs for many domains [1]. However, existing QA methods cannot support an interactive question answering process for such data. Our method for building IQA systems can provide a conversational and interactive interface for these Q&A resources which was not previously available.

When designing an IQA system, it is necessary to represent questions in a format that allows interactive QA. The representation should capture a set of information points which are essential for showing a user's intention according to the domain of QA. These points are required for finding the question from the knowledge base that precisely addresses a user's query.

A real-world example of this would be an IQA system for flight booking. The necessary information points or attributes for finding a flight are departure time & date, airline, price range, etc. In task-oriented dialogue systems, identifying the set of essential attributes for understanding the user's intention is a straightforward process. However, automatically identify the set of attributes for an IQA

system based on text documents is a complex problem. Finding a solution to this challenging problem is essential for building a domain-specific QA knowledge base.

An IQA system requires a Natural Language Understanding (NLU) component for building semantic representation from a q-a set and user's queries. Suitable semantic representation is crucial for understanding the user's intentions and correctly responding to their requests. A task-oriented dialogue system uses a domain ontology and a hand-crafted frame structure for each specific intent to create the semantic representation. For example, a travel dialogue system has a frame to map the following questions to the 'search flight' intent: "show me the flights from Toronto to Vancouver on June 10" or "I want to fly to Vancouver on June 10 from Toronto."

In task-oriented dialogue systems, it is possible to manually determine the intents and various ways of expressing them to build the NLU component. However, using a manual method is not suitable for creating the semantic representation for a non-task-oriented IQA system due to the very high volume of effort required. This inapplicability of manual methods and high volume of effort are caused by two reasons, which are also the challenges for automating this process. First, the number of intents for the IQA system equals the number of questions in the q-a set, assuming each question is unique in meaning in the q-a set. Second, the possible variations to express a specific question (intent) is usually larger than variations of

requesting a specific intent within the task-oriented dialogue system. For example, consider the following equivalent questions related to the domain of car manuals: "Where is the parking brake?", "What is the location of the parking brake?", "How can I find the parking brake?", and "I want to know where the parking brake is located." all these questions express the same intent in an IQA system.

A domain-specific ontology can reduce the manual effort required for building the NLU component. This effort reduction is possible by using the ontology features, such as synonym terms, to detect various phrases for describing an entity in questions. An automatic ontology creation process is presented in our IQA framework using domain term extraction and phrase clustering. In addition, Semantic Role Labeling (SRL) is used for discovering frame structure and creating the semantic representation of a question. SRL provides a generic semantic structure for representing the information in sentences. Using the SRL representation for building the NLU component can reduce the manual effort by eliminating the textual differences when expressing an intent (as seen in the above example from car manuals domain) and providing the same or very close SRL representation for semantically similar questions.

1.3 Objectives and Contributions

A data-driven approach to dialogue system generation leverages data to facilitate the building process. Such methods extract the structure and features of domain-specific dialogue from a corpus of recorded dialogues. To the best of our knowledge, there is no well-established method for building dialogue systems from a domain-specific corpus of questions and answers. Our objective is to develop a framework for building the IQA dialogue system by analyzing domain-specific q-a sets and documents. The dialogue system enables user interaction for QA tasks and dialogue management.

The structure of the IQA system presented in this work is based on iNAGO's IQA system described in Section 1.2. The main contribution of this work is to automate the knowledge base creation process in developing such an IQA system. The main part of our IQA framework is the NLU component and its supporting ontology and knowledge base. The NLU has two important roles in the framework. First, it generates the semantic frame representations from questions which form the knowledge base of our IQA system. Second, the NLU component generates the same semantic frame representation from a user's utterance during dialogue. This enables the IQA system to understand the user's intent, facilitates dialogue management, and matches the input request to the knowledge base, in order to complete the

question answering process. The other component of our IQA framework is a dialogue management system, which interacts with the user to catch the user's intention to best answer the user's query.

The questions in the q-a set are usually non-factoid with multi-line answers. The intention is to provide a comprehensive response to each question. If the user asks a question equivalent to one of the q-a pairs in the knowledge base, the answer from the matching q-a pair is shown as a response. If the question does not lead to matching with exactly one of the questions in the set, the IQA's interactive process assists the user by asking supplementary questions to narrow down the set of related q-a pairs to a single match.

The contributions of this thesis are summarized as follows:

- Reducing the manual effort for developing an IQA system by presenting an automated ontology creation and a rule-based method for IQA knowledge base generation.

- An unsupervised method to automate the ontology creation, where the ontology is used to identify attributes and values for question representation.

- A rule-based approach to generate a knowledge base for IQA based on SRL.

- A dialogue management system to interactively answer a user's question based on the generated knowledge base.

- Experiments in the domain of car manuals to evaluate the proposed methods by comparing it to multiple baseline QA systems.

1.4 Thesis Outline

The rest of the thesis is organized as follows. Chapter 2 offers a review of the research works related to QA systems, QA for question-answer databases, and dialogue systems. Chapter 3 describes our IQA framework. In chapter 4, we explained our experiments and presented the evaluation results in comparison to the baseline methods. Finally, chapter 5 summarizes our contributions, analyzes the strengths and weaknesses of our method, and offers directions for improvement of this work.

2 Background and Related Work

The research of Question Answering (QA) has a very long history, almost as long as scientists have been working on Artificial Intelligence. There are many different approaches to QA. The focus of this work is on QA research based on text documents and knowledge bases. This chapter explains the QA research in the two mentioned areas to familiarize the reader with the relevant concepts and techniques which are used in our proposed method.

2.1 Common Architecture of QA Systems

Approaches to QA can be differentiated based on the type of their information source, the application domain, the architecture of the system's components. These components use NLP and IR methods for processing the input or searching. The QA methods are categorized according to the type of information source and their target question types. Thus, QA is too complex of a problem to have a unified algorithm or model solving all sort of QA variations and needs.

2.1.1 IR-based QA

In a QA system using unstructured documents as its information source, the main components are query processing, document retrieval, and answer extraction components. The following paragraphs briefly describe the components in an IR-based QA system.

1. **Query processing**: The input query is processed in multiple steps as follows. Preprocessing of the input by removing stop-words and stemming; creating an IR keyword query; Query Expansion to increase the chance of finding the target information from the documents mentioning the same concepts but expressed in different terms. One processing step in factoid QA is question type classification and predicting the expected type of answer. (person, place, time, etc.) The QA system only considers extracting the occurrences of the predicted type of answer from the documents.

2. **Document retrieval**: The methods of document retrieval are differentiated based on how they formulate the similarity of a query and the documents. The successful models use statistical similarity measures such as Vector Space Models (VSM) [44, 55], Term Frequency and Document Frequency of keywords appeared in the query (TF-IDF) for assigning weights to query terms [45, 49].

3. **Answer extraction**: The Answer extraction component uses the following steps. Passage retrieval for finding the snippets from the document which potentially include the answer, Named Entity Recognition (NER) for finding special names in the text, Information Extraction (IE) for detecting events or relations from the text, and Document Summarization for complex questions which requires combining the information from multiple sources to create a comprehensive answer.

2.1.2 Knowledge-based QA

In knowledge-based QA systems the main components are a knowledge base, knowledge base creation/completion, query formulation, and query search components. A knowledge base is a special kind of database containing structured representation of information for the purpose of applying automatic deductive reasoning to give answer to a complex request or question. An ontology can be considered as a specific kind of knowledge base for building knowledge-based QA systems. The answer retrieval task in ontology-base QA systems is addressed by means of an unambiguous internal knowledge representation. Both query formulation from questions and the knowledge representation in the ontology are based on standardized collection of entities, concepts, and relations. [35]

The following paragraphs briefly describes the components in a knowledge-based

QA system.

1. **Ontology**: The primary component for this type of QA systems is an ontology. It represents the span of information available to the QA system. A formal definition of ontology is given in [18]. An ontologyis usually defined as a formal explicit description of concepts in the domain of discourse, together with their attributes, roles, restrictions, and other defining features [37]. Similarly, the relations between the entities or concepts are represented formally. The relations for sub-type (sub-class) and instance-of are two very common relations in ontologies. A domain ontology includes the types of concepts and entities that are used in domain-specific documents.

2. **Knowledge base creation/completion**: One type of work in building knowledge base is about ontologies created for representation of general knowledge. For example, Yago is generated based on Wikipedia [25], and Freebase is built as a large community-driven collaborative project for structuring a wide range of human knowledge [9]. These knowledge bases can be used for answering factoid open-domain questions.

3. **Query formulation and search**: In knowledge-based QA, a user's question in natural language is converted into a searchable query for the knowledge base. The query formulation step generates a database or logic form query

from the user's question. One of the query formulation methods is *Semantic Parsing* for creating formal logic queries, or database queries, from a textual question [63, 6]. The generated query is used in a search engine to find the answer from the knowledge base. For example, a SQL query is generated from a user's question and is searched in an SQL database (QA system's knowledge base) for finding the answer.

2.2 QA from Database of Q-A Pairs

One common form of information is a collection of question-answer pairs (q-a set) such as Frequently Asked Questions (FAQ) lists. Such databases are created by a community, or business as the knowledge base for a specific domain. Building a QA system for these resources facilitates accessing the information for the user. Considering the unstructured text format of the q-a pairs, IR-based methods were studied for this type of QA task in previous words.

The *FAQ Finder system* tackled the QA task for FAQ archives extracted from the web [11]. The FAQ files for this system cover wide variety of topics. To find a matching question, IR search using TF-IDF is used combined with ontology-based semantic similarity. The semantic similarity method was the marker passing algorithm over WordNet [43].

In the FAQ Finder system, they made a number of important assumptions about

the q-a database which should be considered for building similar QA systems:

1. Locality of information: All the information needed to determine the relevance of a q/a pair can be found within the q/a pair

2. Question adequacy: The question part of a q/a pair is adequate and most relevant part for determining the match to a user's question

3. Distinct set: All the q/a pairs in the dataset should be distinct so the question matching can effectively find a unique q/a pair from the set which matches the user's question very closely.

Also, these points are important for the task of question&answer generation from a domain-specific document to assess the coverage and usability of the result q-a set.

The common IR evaluation metrics precision and recall are used for evaluation of the QA system. Precision is the ratio of retrieved documents that are relevant. Recall is the ratio of relevant documents retrieved to all documents that are relevant from the document set.

$$Precision = \frac{TP}{TP + FP} \tag{2.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

Since all the questions are unique, for any input question there is only one matching q-a tuple. Accordingly, the precision and recall measures are modified to better represent the performance of the QA system. *Modified Recall* is the percentage of questions for which FAQ Finder returns a correct answer when one exists. This calculation does not penalize the system if there is more than one correct answer in the database. *Rejection* is calculated as the ratio of questions that FAQ Finder correctly reports as being unanswered in the file. Rejection in FAQ Finder is possible by setting a minimum threshold for the relevancy score of a q-a tuple assigned by the IR model.

These measures also has a trade-off like the one between normal precision and recall. If the threshold for cutoff the returned q-a pairs from FAQ Finder is very high, then the rejection will increase and the recall significantly decrease and vice versa. However, the modified recall needs effective answering techniques to increase to very higher values and is the more relevant metric to the performance and quality of answering method.

2.3 Community Question Answering

Community Q&A (CQA) Websites creates a platform for users to ask and answer questions. There are many such websites designed for CQA like Quora, Yahoo! Answers, and StackExchange. Usually, the questions are classified and tagged based

on their topic. The community of a CQA platform cooperate to maintain the relatedness of questions and vote to select correct answers. This collaborative contribution creates a valuable resource which is useful to answer other users having the same question.

The problem of identifying similar questions from CQA archives is similar to the QA for question-answer databases. The research in CQA is about how to answer another user's question by finding similar existing questions from the CQA archive that are previously answered.

The availability of non-factoid questions with CQA archives provides the opportunity of researching methods for complex and non-factoid QA. Most of the previous works in CQA used IR search engine framework and studied the effectiveness of various methods for scoring question-question and question-answer relatedness [34, 52]. In the following subsections, we reviewed the important methods in CQA for calculating relatedness between question and answer text.

2.3.1 Question Search with Statistical Translation Models

One problem is word mismatch between semantically similar questions. This problem is called Lexical Chasm [7], is well-known in Question Retrieval for CQA websites and QA area. Jeon et al. addressed this problem by training a statistical translation model with pairs of *similar questions* from CQA archives [26]. Two

questions are considered similar when their answers are semantically similar. The semantic closeness of answers is measured using four document similarity measures and their ranking positions. The learned translation probabilities showed better performance in comparison to language model QA methods.

2.3.2 Question Retrieval with Language Model for IR

A language model approach for question search is proposed by Duan et al [22]. introducing notions of question topic and question focus. The target set of questions is used to build a question tree data structure. The question tree is the prefix tree built from the topic chains of questions. Then, a Minimum Description Length cut in the tree separates it into branch heads (question topics) and tails (question focuses).

In language modelling based QA, the relevance of a question to a given query is predicted with the probability distribution of the language model (LM) [41]. In this work, two multinomial probability distributions over terms are estimated for topic and focus of questions. The learned LM is used to find the most probable target question given a new query. The evaluation of this method shows improvement over normal VSM and LM for QA.

2.3.3 Question Search with Syntactic Tree Matching

In another question search method Wang et al. proposed a Syntactic Tree Matching (STM) method to measure question-question similarity [58]. A Tree Kernel is generated from the constituent parse tree of a sentence. STM encodes syntactic, semantic, and lexical features extracted from tree kernels [19] of questions. This method needs no training data and is robust to minor grammatical errors.

2.4 Dialogue Systems for QA

Dialogue systems (DS) and chatbots enable computer programs and applications to communicate with users in natural language form. Task-oriented dialogue agents are designed for a particular task to interact with the user in conversation format. The DS gets the required information from the user during the dialogue to accomplish a specific task. In contrast, chatbots are designed for unstructured human-like chats without focusing on accomplishing a particular goal. They are set up to mimic the characteristics of human-human conversations. In this context, an IQA system is classified as a task-oriented dialogue system.

2.4.1 Frame-based DS and Domain Ontology

Most of the modern task-oriented dialogue systems use frame structure to represent the information provided during the dialogue. A *domain ontology* specifies the kinds of intentions the system understands from the user utterances. The ontology is defined with one or more frames, each has a collection of slots, and a set of possible values specified for each slot. The frame-based DS architecture has been influential in extracting the structure of task-oriented dialogue and building automatic systems [17]. Also, the frame-based DS has been fundamental in building of modern commercial digital assistant systems like Siri, Google now, etc.

The slots of the ontology specifies what the system requires to understand, and the set of values, filling each slot, are constrained to a particular semantic type. For example, in the travel domain a frame for booking a flight can have the slots and filler types as in Figure 2.1.

2.4.2 Natural Language Understanding in Frame-based DS

The Natural Language Understanding (NLU) enables the dialogue system to extract three things from the user's utterance:

1. Domain classification (only in multi-domain systems), e.g., travel planning, searching restaurants, or managing calendar

Slot	Filler Type
Origin City	CITY
Destination City	CITY
Departure Time	TIME
Departure Date	DATE
Arrival Time	TIME
Arrival Date	DATE

(a)

```
DATE
  DAY         (BOUNDED INTERGER 1-31)
  MONTH       (BOUNDED INTEGER 1-12)
  YEAR        (INTEGER)
  MONTH-NAME (MEMBER {January, February,…, December})
  WEEKDAY     (MEMBER {Monday, Tuesday,…, Sunday})
```

(b)

Figure 2.1: (a) A frame for flight booking in a dialogue system with typed slots. (b) A semantic value type with hierarchical structure.

2. Intent determination for the specific task the user wants to accomplish like showing a flight, or adding an event to the calendar

3. Slot filling for identifying the slots and extract the fillers from user's utterance for performing the task.

Different techniques can be used in the NLU component for extracting these information.

In many commercial dialogue systems, NLU component is based on rule-based techniques to extract the slots and filler values. The example of these techniques include regular expressions, complex automata, and semantic grammars (CFG or Recursive Transition Network) [8, 59, 27]. Rule-based NLU provide accurate results however defining the set of rules is usually an expensive and long process. Another problem with them is low recall especially in broader domains.

2.4.2.1 Semantic Grammars

One of these rule-based methods is *Semantic Grammar*. A Semantic Grammar is a Context Free Grammar (CFG) with left hand of rules corresponding to semantic entities being expressed (slots) and right-hand side shows the patterns of words in utterances that implies the slot fillers for the associated slot. Some nodes in a semantic grammar may correspond directly to a slot in the frame. For example, the ORIGIN slot in a flight booking DS. Therefore, the slot-value can be directly produced from the sentence parsing based on the designed semantic grammar. In such cases, when direct mapping from parsing to slots is possible, the semantic grammar definition is useful for creating a canonical form for the value. for example DD:MM:YY for a TIME slot. Despite its accuracy, semantic grammar approach is unable to deal with ambiguity in the input and requires many hand-written rules. These issues limits the semantic grammars to be used for large domains like QA.

2.4.3 Supervised Machine Learning

The discussed rule-based methods are very precise in interpreting the text but the production can be incorrect for some special cases like negative values, e.g. “can’t go on Monday”, “anytime except Friday”. Moreover, the understanding is restricted to their rules and cannot understand the variations of a request not included in

their rule set.

Machine learning methods can help to improve these limitations with models capable to generalize. Supervised machine learning methods can be used if labeled data is available. The training data is in form of sentences annotated with correct slots and filler values. A classifier model can be trained to determine the domain and intent from the sentences, and a sequence learning model is used to map sentences to related slots and values [39]. For building the slot filling, it is possible to use rule-based systems to bootstrap the machine learning models and perform the training in a semi-supervised way [51].

2.4.4 Dialogue Management in Dialogue Systems

The Dialogue Manager (DM) component controls the structure of the dialogue in the DS. DM responsibilities in DS is as follows.

1. Processing the input given by the Automatic Speech Recognition (ASR) and the NLU component.

2. Using a state mechanism for tracking the dialogue status.

3. Interfacing with the task manager (like a domain ontology) for answering the user's request.

The most common architectures for dialogue management are finite-state and frame-based. Also, these architectures are widely used in commercial dialogue systems. Other approaches are the classical plan-based, and the information-state dialogue managers. The latter is a more powerful approach that includes a probabilistic version of information-state manager based on Markov Decision Processes (MDP).

The simplest dialogue managers use finite-state automaton (FSA). Figure 2.2 shows the FSA's structure for a simple flight booking system. The FSA dialogue manager asks a series of questions to determine the value of slots. Also, the dialogue system only expects user inputs relevant to the current question, otherwise the user utterance is ignored. The state of the FSA is associated with the current question and the arcs correspond to actions dependant on the user's response.

The speaker in control of a conversation is described as having the *initiative*. Hence, dialogue systems with FSA dialogue management are *system initiative*. A completely system initiative DS is very restrictive and impractical for task-oriented dialogues. Therefore, practical dialogue systems use architectures that allow *mixed initiative* conversation. This enables the user to take control of the conversation at some points during the dialogue to restart or modify their previous utterance.

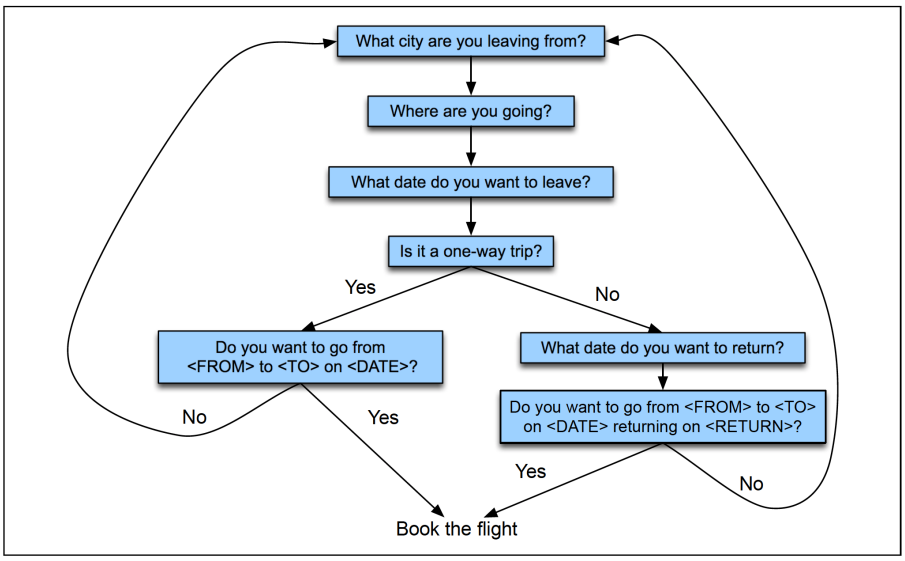


Figure 2.2: Dialogue manager with finite-state architecture for booking flight [27]

2.4.4.1 Frame-based Dialogue Management

The goal of a frame-based dialogue manager is to ask questions from the user to detect and fill slots based on the user's answers. The dialogue continues until the frame is completed for doing the task, e.g. a database query for available flights. The system can recognize and fill multiple slots if the user's answer include information related to other slots. The system skips a question if its slot is filled.

An advantage of Frame-based DM is that it allows *mixed-initiative* dialogue. In a frame-based DM, the dialogue advances by asking user questions to fill the slots in the frame. The DM allows users to provide the values for multiple slots in one sentence. Hence, DM checks the current status of dialogue frame and skips asking

a question for a filled slot. Frame-based DM does not impose strict constraints on the order of providing information during the dialogue, like the FSA architecture.

Some domains require multiple frames to be able to handle different types of user's intentions or requests. For example, in a dialogue system for travel domain, it may be required to have multiple frames for different types of user's questions such as general route information (*Show me flights from New York to Toronto for next month.*), car or hotel reservation (*I need to reserve a hotel in Toronto for one week.*), etc. Having multiple frames is necessary to disambiguate which slot from which frame is expressed in the user's utterance when he/she switches to a different frame during the dialogue. Frame-based systems, can be implemented as *production rule* systems to enable dynamical frame switching [46].

2.5 Interactive Question Answering

Interactive Question Answering (IQA) emerged from adding Dialogue Systems capabilities to QA systems. Methods such as discourse model, dialogue management, and context handling enable conversational interaction with user to enhance QA experience. A dialogue can be initiated with a user in cases where there are too many or too few answers, or there is some ambiguity in the request.

One challenge when building an interactive system working on dialogues is context management. It is one of the responsibilities of DM component in DS architec-

ture [54]. Context management is necessary to understand references in following user's questions. Also, context management identifies and adjusts the *Topic* of the queries during IQA dialogue. In real conversations the topic is usually specified implicitly. In TREC-QA task for IQA, this issue is addressed by introducing an explicit *Target* value to manage the topic during IQA dialogue [20].

In the following paragraphs we reviewed some related work to IQA.

2.5.1 YourQA Open-domain IQA System

In a work by Quarteroni and Manandhar [42], YourQA, an open-domain QA system is combined with conversational capabilities for creating an IQA system. The open-domain QA system uses the web for obtaining the answer and is capable to answer factoid and non-factoid questions. The Artificial Intelligence Markup Language is used for adding the interactive interface to the QA system. The user utterance is matched against a set of dialogue patterns and produces a coherent answer based on a list of responses associated with the patterns.

Their framework divides into QA and dialogue system. The QA system consists of question processing, document retrieval, answer extraction and user modelling modules. Question type classification is the main focus of the question processing. The document retrieval is based on Google search engine and the top 20 returned documents are used for answer retrieval. For answer retrieval, the returned doc-

uments are segmented into sentences, ranked and filtered using different features depending on the question type. The set of features include BoW, Named Entity Recognition, phrase extraction with manual rules, head nodes of phrases in syntactic parse tree (VP, NP, PP), bigrams, and WordNet lexical similarity.

In this work, the summary points of the desiderata for an IQA system are context maintenance, utterance understanding, mixed initiative, follow-up proposal, and natural interaction. There are a range of possible dialogue patterns registered for the system which the user's utterance is matched against them. The matched pattern helps producing a coherent answer based on a set of answer templates associated with the dialogue patterns. The DM algorithm uses state transitions based on text patterns and manual rules.

The experiments are based on Wizard-of-Oz (WOZ) method for dialogue systems evaluation [36]. In WOZ method, a human operator, *wizard*, emulates the behaviour of the QA system while the user believes to be interacting with a automated dialogue system. Six tasks were defined for finding specific information using the IQA system. The evaluations of this system showed users' preference 87.5% and 58.3% toward the interactive interface of the system instead of the simple QA system.

2.6 Semantic Role Labeling

Semantic Role Labeling (SRL) is a method for making a semantic representation from sentences. A common SRL method is Natural Language Processing (NLP) pipeline starting with Dependency Parsing [16] and assigning semantic role labels to the phrases of the dependency parse tree. The SRL roles describe a collection of *thematic relations* specified by a linguistic model of semantics. SRL provides a general meaning representation applicable to any text which provides useful semantic features for QA and DS tasks [48]. There are different linguistic models of semantic roles such as Frame semantics [4], AMR [5], and Propositional semantics [21].

In this work, we used Propositional semantic model which is based on thematic proto-roles and argument selection in Linguistics [21]. Each sentence is represented with one or more propositions. Each proposition is shown as a predicate (usually a verb) and its arguments. The arguments are the phrases from the sentence carrying the semantic roles of the predicate. For example, in the sentence *'John drives a 1958 DeSoto.'* the proposition can be shown as drive(John (A0), a 1958 DeSoto (A1)). [1] The argument A0 has the role of agent (driver) for the predicate drive and the argument A1 has the role of patient (vehicle).

The semantic representation of a sentence is called *predicate-argument structure*

[1] propbank-website:https://verbs.colorado.edu/propbank/framesets-english/drive-v.html

(PAS). PAS is the set of verb propositions for a sentence. PAS explains *who* did *what* to *whom*? *when*, *where*, and *how*. The semantic roles in PAS are denoted with numbers as coarse-grained labels like Arg0 (Agent), Arg1 (Patient), etc. and optional roles like modifiers and adjuncts to the verb are denoted with combination of ArgM and a functional tag such as TMP (when?), LOC (where?), MNR (how?), etc. As we describe in chapter 3, PAS is used for creating the frames in our IQA knowledge base generation framework. Figure below shows the representation of PAS for a sentence:

PAS('Kristina hit Scott with a baseball yesterday.') = {

hit.01: [Kristina (A0)] [hit (PRED)] [Scott (A1)]
[with a baseball (AM-MNR)] [yesterday (AM-TMP)].

}

2.6.1 The Proposition Bank (PropBank)

PropBank [38] is a collection of linguistic resources for propositional semantics. PropBank *role-sets* provides a collection of frames for all the predicates. A frame provides very fine-grained (predicate-specific) argument labels and descriptions for semantic roles. PropBank also provides manually annotated corpora for training models for SRL parsing. The manually annotated PropBank corpus uses the above number format. This helps the parsing models learn better SRL patterns across

different predicates. PropBank corpus is based on Penn TreeBank [32], which provides syntactic parse trees for an English text corpus.

3 Methodology

3.1 Overview of Framework for Building the IQA System

In this chapter, we explain our method for building a knowledge-based Interactive Question Answering (IQA) system. A brief overview of the method is as follows. The information source of the QA system is a domain-specific set of question-answer pairs (q-a set). The objective of the IQA system is to answer a user's question by matching it to a question-answer pair (q-a pair) from the q-a set. A knowledge base for the questions in the q-a set is built using the frame-based representation. The matching is done by searching the IQA dialogue frame in the knowledge base. The Dialogue Manager (DM) controls the response of the IQA system. Based on the result of searching the knowledge base, DM either finds and returns the answer, or asks a question to reduce the number of matched q-a pairs. The IQA dialogue process is repeated until the user finds the q-a pair which answers their information need.

3.1.1 Question-Answer Set

The domain of questions is bound to the source document from which the q-a pairs are generated. The q-a set is organized based on the following assumptions:

- *Completeness*: the q-a set covers the information in the source document. In another word, there exists one q-a pair that matches any user's question. The q-a set defines the scope of users' questions from the IQA system.

- *Question adequacy*: we assume the information in the question part of a q-a pair is adequate for determining the match to a user's question. The answer in the q-a pair provides the necessary information to be presented to the user as the answer.

- *Uniqueness*: questions in the q-a set are all semantically unique. Thus, there are no duplicate matching situation when using the set in the IQA system. In another word, the questions in the q-a set are semantically distinct so that at most one q-a pair answers a user's question. In this work, the car instruction manual domain is used as a case study. Hence, we often use examples related to cars which are extracted from a car manual.

3.1.2 IQA System Architecture and Dialogue Management

The general architecture and subsystems of an Interactive Question Answering system are shown in Figure 3.1. Automatic Speech Recognition (ASR) subsystem is responsible for converting input utterances to text. The NLU component gets the text created by the ASR component and creates a semantic representation from it. This representation is sent to the Dialogue Manager (DM) component which updates the dialogue frame based on the received semantic representation of the user's input. Then the Question Answering (QA) system searches for the dialogue frame in the knowledge base and returns the search result back to the DM component. The dialogue status is adjusted and the response of the IQA system is determined by the DM component. This response is given to the Natural Language Generation component, where it is transformed into a proper text to be understandable for the user. Finally, the response in speech format, from the Text-to-Speech Synthesis component, is presented to the user. In this work a text-based interface is used for our IQA system, even though speech interactions can be enabled in our IQA system by addition of speech processing components.

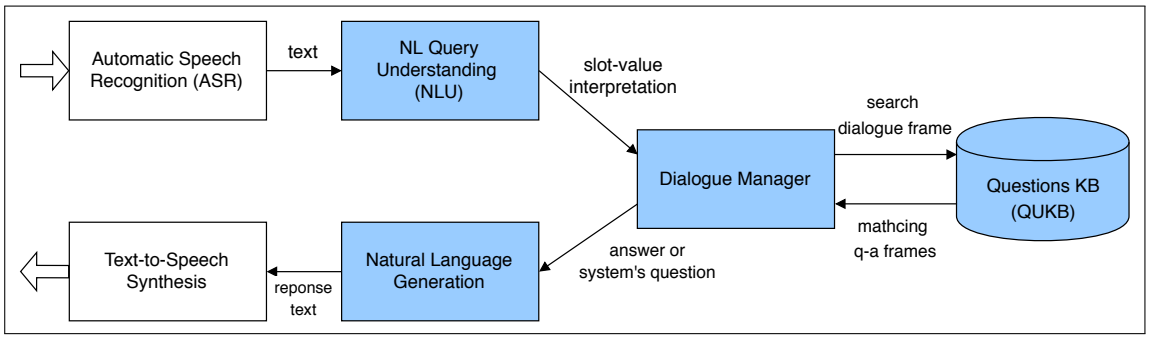


Figure 3.1: General architecture of the components in an IQA dialogue system

3.1.3 Knowledge Base Generation and PAS Frame Domain Mapping

The knowledge base of the system comprises of a collection of frames for each of the questions within the q-a set. The generation process for the knowledge base is shown in Figure 3.2. This process involves creating domain-specific frames using our NLP Pipeline for Frame Production (NLPF), and storing the domain frames of questions and the corresponding answers into the Questions Knowledge Base (QUKB). In order to generate a domain-specific frame, the NLPF uses the domain ontology and the mapping rule-set provided by the Ontology Building and Rule-set Development processes. A PAS frame for a question q_i is generated by the Domain-aware SRL component equipped with the domain ontology. Then, the PAS frame is converted into the domain-specific frame by the Domain Adaptation component which uses the mapping rule-set for this conversion. Finally, the generated domain frame of q_i is added to the QUKB.

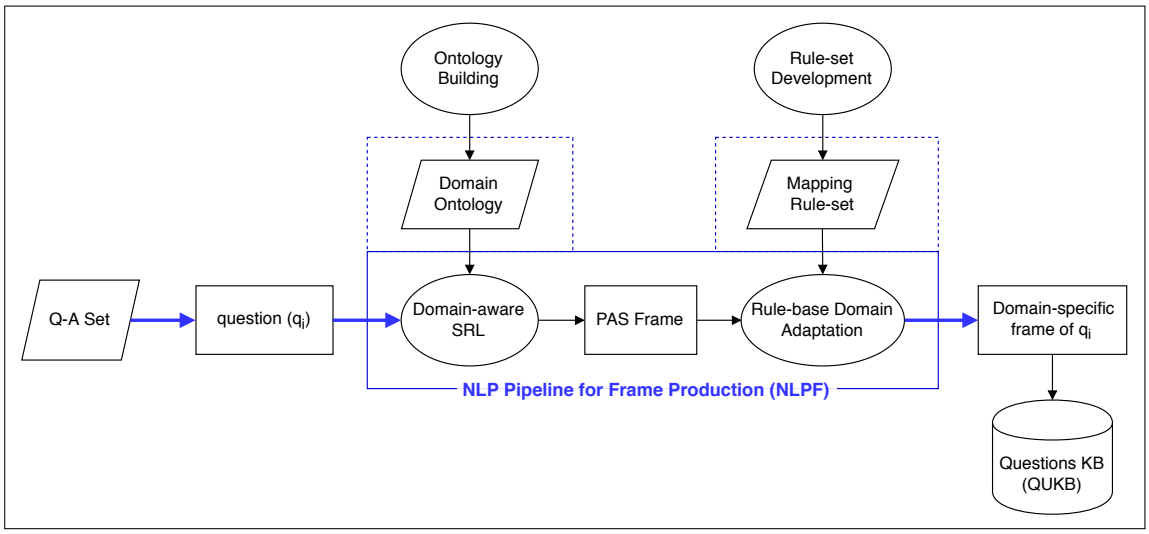


Figure 3.2: The framework for QUKB generation using the NLPF for producing domain-specific frames

Once a question is asked a corresponding dialogue frame is initialized. Then the question is processed, and a number of slots within the dialogue frame are populated by the NLPF component. In order to find the answer to a user's question, the IQA system matches the dialogue frame to existing frames within the knowledge base.

This approach highlights the requirement of solving two main issues in knowledge base generation. First, the frame and slots for the application domain must be defined, and second, the slots have to be filled using information within the question. Usually in data-driven approaches to the slot definition problem, a dialogue corpus from the human-human or human-system utterances is analyzed, and the structure of the task-oriented dialogue (i.e. frame and slots) for a specific applica-

tion is extracted [17]. However, our goal is building the slots and values from the q-a set without utilizing a dialogue corpus. The second problem issued about slot filling can be addressed through rule-based or machine learning NLU techniques.

To generate the frame-based representation we use a linguistic approach by leveraging existing NLP methods and resources. Semantic Role Labeling (SRL) is used to create the basis of our frame representation. SRL gets an input sentence and creates a Predicate-Argument Structure (PAS). PAS is a frame-like representation of the question intent. However, PAS frames are generic and need to be adapted to the system's application domain.

The linguistic semantic approach in SRL creates a general representation which is applicable to any domain. SRL creates a fairly accurate representation of the meaning for any sentence. Therefore, all the unique questions in our q-a set have distinct PAS frames. However, the PAS frames and slots may not fit the terminology of the target application domain, therefore there is a requirement for domain adaptation of PAS.

There are two aspects to the incompatibility between PAS frames and target frames.

First, the granularity of the PAS and target frames may differ. The granularity of the frame representation can be defined based on the number of its slots and how the terms are distributed among the slots. The granularity of a PAS frame

usually does not match that of the target frame. For example, in a car manual domain we may want the verbs 'fix', 'maintain', and 'repair' to represent a single equivalent value for the slot 'User Action'. However, in PAS frames each of those verbs has a separate frame which contains different slots. Therefore, it may be the case that numerous PAS frames will be mapped to a single target frame or vice versa. This can be achieved by merging or splitting the PAS slots to fit the target domain granularity.

Second, the slot names of a PAS frame may not properly represent the corresponding domain terms. The PAS slot names are defined based on the generic meaning of verbs and its arguments. However, they should be representative of domain terms. PAS slots are generally verb-oriented and do not properly convey the concepts that their associated values represent within the target domain. In our method, domain adaptation was achieved using an ontology of domain terms and a rule-based slot-value mapping.

3.2 Components of the IQA System

There are many variations in the design of components in QA systems. These variations are based on the requirements of the specific problem they aim to solve. In this section we describe each of the components within our IQA system.

3.2.1 Questions Knowledge Base for IQA

The **Questions Knowledge base (QUKB)** of our IQA system contains the frame representation of a question and its textual answer for each question within the q-a set. Each question-answer pair from the set is represented as a frame with a set of slot-value pairs in the QUKB. Therefore, we can define the QUKB as an aggregate of frames for all questions within the q-a set.

A frame-based dialogue system can offer support for multiple intents, where each intent is represented by its own *frame structure*, and each of these frame structures would have a fixed set of slots. For example, in a dialogue system in the travel domain, different intents such as 'booking a flight', 'reserving a hotel', and 'renting a car', can each have a separate frame with a fixed set of slots. Even though the QUKB has a single frame structure, not all slots are present within the frame representation of a question. This means that the frames representing a question in QUKB may include a subset of the domain slots of the q-a set. A question frame in the QUKB specifies the slots and the corresponding values which should be interpreted from the user's utterance. Therefore, the IQA has a defined scope of questions which it can answer. This scope prevents it from interpreting out-of-scope phrases and intents.

In a task-oriented dialogue system, once a frame is interpreted from the users

utterance, the system then decides how to respond to the request, and performs the required action(s).. However, our IQA system performs the question search even if the frame for a user's question includes incomplete slot-value information and does not exactly match a frame in the QUKB. The incomplete matching can trigger a dialogue with the user in our IQA system so that the frame for the user's question can evolve to better match a frame in the QUKB.

3.2.2 Control Structure for IQA Dialogue

In order to conduct a successful interaction with the user, a control structure is required for the Dialogue Manager component. A control structure, manages the flow of the conversation with the user. Existing control structures for task-oriented dialogues, are not compatible with our IQA frame design.

A **Control Structure for the IQA Dialogue (CSID)** was designed based on the frame definitions with a variable number of slots, and it specified the possible actions to complete the interaction with the user. This emulates the responsibilities of a Dialogue Manager component within an IQA system as shown in Figure 3.1. The goal of the CSID is to build a dialogue frame using the domain frame provided by the NLPF component. The NLPF creates the domain frame by identifying and filling the slots with user-provided information. Once the dialogue frame is built, the QA component searches the QUKB for q-a pairs which match the dialogue

frame, and then the CSID deciding on the required action to fulfil the request according to the matching results.

Two special cases may occur within the matching process. First, it is possible that there are too many matching items between the QUKB and the users request. In such cases, it is not preferable to return all the obtained matches as a response to the user. Therefore, the obtained matches should be narrowed down by adding slots to the dialogue frame. Another case which may occur is when there are no matches to the dialogue frame. This is a dead-end situation for the interaction, since no information has been obtained through the matching with the QUKB. However, if we broaden our frame by removing some of the slot-value pairs, the matching result could be relevant to the user's request.

The matching results were used to determine possible slots to be added and asked of the user, or which slots can be removed from the dialogue frame. Conventional task-oriented methods use a fixed order (system-initiative design) to fill frame slots, whereas our CSID incorporates a dynamic method.

A *SELECT* algorithm in our CSID enables the dynamic slot filling for the IQA system. This algorithm determines the next slot to be filled based on the following three factors: the content within the dialogue frame, the frames of matching questions from the QUKB, and the maximum number of q-a pairs to show to the user which is denoted by the parameter K. These factors in the deployed SELECT

algorithm control the behavior of the IQA dialogue during the IQA task. This dynamic slot filling method gives more control to the user to direct the conversation and enables a method to resolve disambiguates within a user's intention.

Using a slot selection algorithm, the next slot to be filled is determined based on the search result of matching question frames in the QUKB. Another advantage of our CSID is that it detects any slot value in the user's utterance and is capable of filling multiple slots from one utterance. Hence, our CSID is classified as a *mixed-initiative* design for the dialogue management control structure.

Figure 3.2 offers a schematic view of a sample interaction with the CSID. The dialogue starts with a generic initial question, in which the systems interprets the user's utterance, and initiates a dialogue frame. This dialogue frame is searched for through the QUKB to find matches. If this search results in no matches, the SELECT algorithm specifies a slot to be removed. If this search does result in a number of matches, but does not exceed the required K-parameter, then we show the results to the user. Otherwise, the SELECT algorithm will specify the frame slot which requires more information. The dialogue then resumes by obtaining the next utterance, and this cycle is repeated until the user's request is fulfilled.

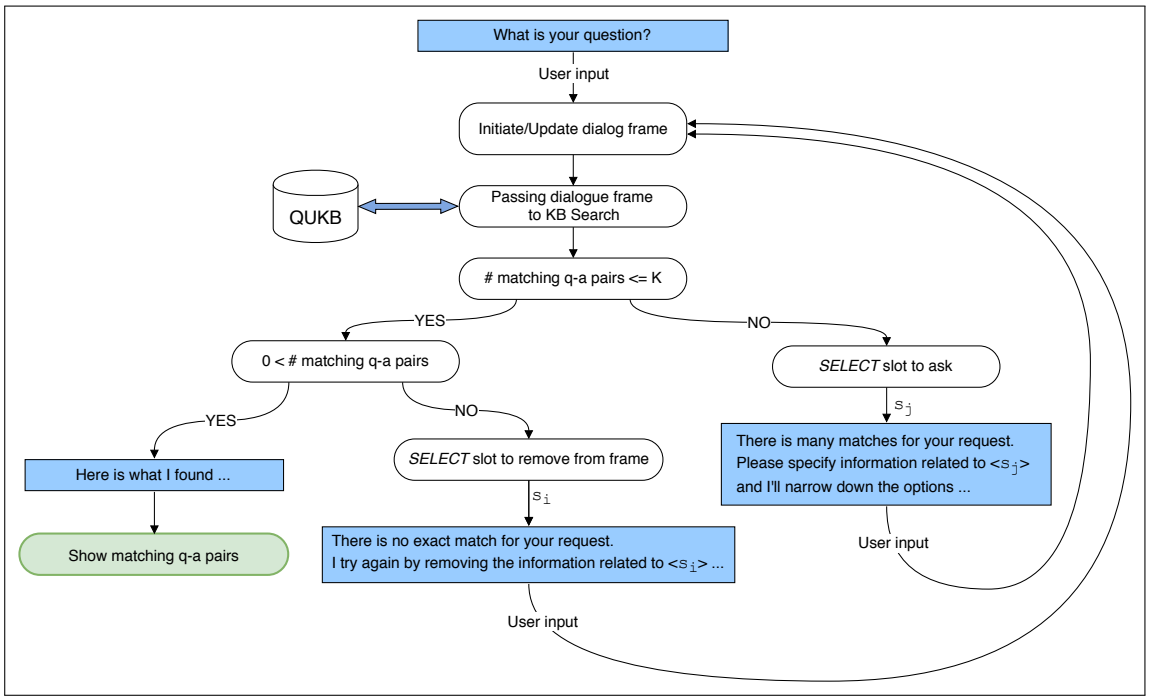


Figure 3.3: The dynamic control structure for IQA dialogue

3.2.3 Natural Language Processing Pipeline for Frame Production

Another key component of the IQA system is the NLP for Frame Production (NLPF) component. This component is responsible for identifying frame slots and assigning their values based on either a question within the q-a set, or a user utterance during dialogue interaction. This emulates the responsibilities of a NLU component within an IQA system as shown in Figure 3.1. Rule-based methods are commonly used for NLU in industrial dialogue systems. However, they require a high amount of manual effort to create rules for mapping the utterances to intents.

Rule based methods on average require thousands of hand-written rules for rule-set creation of a specific domain of application. In order to decrease the amount of manual effort, our NLPF method combines rule-based and linguistic approaches.

SRL parsing was used as a linguistic approach in the proposed NLPF method. This parsing method is based on a statistical model trained with supervised machine learning. The SRL model was pre-trained on annotated text corpora from other domains [15]. When using these models for SRL parsing in a different target domain, we face a decline in parsing accuracy. This decline is caused by the specific terminology or sentence structures of the target domain, which may differ from the training data. To compensate for this decline in accuracy, a domain-aware semantic role labeling method (presented in Section 3.4.4) is used, which allows the system to exploit the domain ontology information. Which in turn eliminates the need for large in-domain SRL training data.

The SRL parsing method results in a number of PAS Frames (described in section 2.6). These PAS frames are then fed to the rule-based portion of the NLPF. The NLPF then applies domain adaptation to the PAS frames, to obtain domain-specific frames (see Section 3.4.4 for a full description of domain adaptation method).

Applying domain adaptation to the PAS frames offers two main advantages. First, the PAS frame structure facilitates the representation of equivalent values for semantic roles despite sentence variation. As an example, if the user states

their request in a passive or active manner, there will be no effect on the semantic roles which are interpreted. Second, rules defined based on a PAS structure are applicable to more cases than rules defined based on text. This flexibility of the rules, decreases the effects of low recall faced in rule-based approaches to NLU.

3.3 Creating the QUKB and Domain Adaptation Rules

3.3.1 Why Domain Adaptation of SRL Frames is Required?

When using SRL for frame creation, we observed two main issues.

First, errors in SRL parsing and insufficient accuracy occur when utilizing SRL models pre-trained on other domain corpora. This happens because in-domain training data is unavailable for training purposes, therefore the model cannot efficiently detect domain specific terminology. If proper encoding of domain terms occurs from a domain-specific ontology, SRL models from other domains should correctly parse the required sentences.

Second, although PAS frames provide a good generic semantic representation of a sentence, they are not suitable for domain specific IQA systems. The slot labels may need to be renamed, for better usability within the IQA dialogue. It is also possible that slot values could be very long, or out of context.

These issues can be better illustrated with an example as follows. In a car

manual domain, if the system is presented with the question q = *'What happens if the gas cap is not tightened properly?'*, the SRL parsing generates the PAS in Figure 3.4. In this question, the argument AM-ADV for the predicate **happen.01** ([if the gas cap is not tightened properly (AM-ADV)]) is very long and there is multiple information points combined in its value. Also, the argument [what (A1)] of the same predicate is not very informative as a slot-value for the semantic frame representation of this question.

PAS(q) = {

happen.01: [What (A1)] [happens (PRED)]
[if the gas cap is not tightened properly (AM-ADV)]?

tighten.01: What happens if [the gas cap (A1)] is [not (AM-NEG)]
[tightened (PRED)] [properly (AM-MNR)]?

}

Figure 3.4: Showing an example PAS from SRL parsing of a question

By applying domain adaptation through mapping rules, we were able to solve the semantic representation problem discussed above. Applying domain adaptation to SRL representations, has proven promising in previous works [14].

3.3.2 How Domain Adaptation is Addressed in Our Framework

The aforementioned problems for PAS frame construction were addressed with two solutions within our NLPF and QUKB generation framework.

Through extracting a *domain ontology* from the target domain corpus the SRL parsing accuracy will improve. A domain ontology consists of domain-specific concepts, and a lexicon specifying members of each concept (as described in Section 3.4). Domain ontology can be used in the NLPF component for Domain-specific Entity Recognition (DER), which takes care of tagging the domain terms in the input sentences. By incorporating DER tags into the SRL, a Domain-aware SRL (DSRL) model can be built (see Section 3.4.4 for more details). DSRL tags help to prevent incorrect parsing of the input sentence.

Domain adaptation of PAS frames can be completed through rule-based mapping. PAS frames are annotated by a domain expert. These annotations are used to generate the mapping rules. The annotation process involves marking priority slots and values from a PAS frame. In order to fill a frame slot from the user's utterance, The IQA system asks the user for the value of the domain slot by showing its assigned label. Therefore, the final rule-set represents the domain knowledge in form of mappings from linguistic semantic frames (PASs) to domain-specific frames.

3.4 Domain Ontology Building

3.4.1 Domain Ontology Definition

The domain ontology is a lexicon of terms related to a specific domain. Domain terms are extracted from a corpus which is also the source of our q-a set (for this research, car manuals were chosen). The ontology can be viewed as a set of *concept*s relating to the application domain. For example, in the domain of car manuals we dealt with questions including concepts for car-parts, driving-states, etc. Each concept has a collection of *concept member*s. Concept members are the instances of the concept occurring in the corpus. Multiple phrases may refer to the same concept member in questions or utterances. Thus, each concept member has a set of acceptable synonym terms. The synonyms are used in Domain Entity Recognition (DER) for identifying the concept members from input text. DER maps the synonyms into canonical forms for all concept members using the ontology.

The domain ontology can be denoted as $\mathcal{O} = \{c_1, c_2, \ldots, c_{N_O}\}$. Where each c_i is a concept with a set of concept members shown as $M(c_i) = \{m_1, m_2, \ldots, m_{N_i}\}$. Each concept member has a set of synonym terms, shown as $T(m_j) = \{t_{1,j}, t_{2,j}, \ldots, t_{N_j,j}\}$, where each $t_{k,j}$ is a synonym term for a concept member m_j.

For example, in the ontology for car manuals, *Car Part* is an example of a concept with members such as *Engine*, *Tire*, and *Door*. There exist synonyms for

Tire such as *'tire'*, *'tyre'*, and *'wheel'* in the ontology. The concept member *Engine* also has synonym terms such as *'engine'* and *'motor'*.

3.4.2 Framework for Domain Ontology Building

In order to build the domain ontology, according to the three-level structure mentioned above, we propose the following process.

1. Domain term extraction using the Topical Phrase Mining method
2. Identifying domain concepts with clustering terms using Word Embedding similarity
3. Manually labeling concepts, ontology adjustments, and adding synonym terms

Figure 3.5 illustrates the aforementioned steps for domain ontology generation, on a corpus.

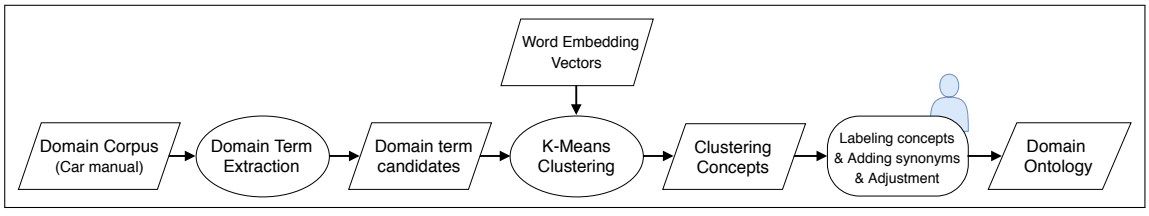


Figure 3.5: The proposed partially automated framework for domain ontology generation

3.4.2.1 Automatic Steps

First, a domain term extraction method is applied to the corpus (car manual). A well-known technique named Topical Phrase Mining (TopMine) was used for term extraction [24]. Then in the next step, the terms are clustered using K-Means clustering based on word embedding similarities [33]. In distributed semantic similarity, each word has a multi-dimensional vector of real values. Each term with multiple words is represented as the average of its word vectors. The similarity of two terms is calculated by the cosine similarity between their corresponding vectors. The clusters from applying K-Means are considered as the initial concepts for building the domain ontology.

3.4.2.2 Manual Steps

The final step requires manual adjustments to the ontology by a knowledge expert. The manual revision of the domain ontology involves removing phrases which do not represent a domain term, moving and re-assigning concept members to better suitable concepts, labeling concepts with appropriate names, and adding synonym terms for the concept members. The result is a domain ontology that covers all the terms in the corpus and user utterances. The final domain ontology can then be used in Domain-aware SRL for creating PAS frames.

3.4.3 Domain-specific Entity Recognition using Ontology

Domain-specific Entity Recognition (DER) creates matching patterns from all the synonym terms in the ontology. The patterns are then compiled into a pattern matching automaton which DER uses to recognize the terms from an input sentence. Once DER recognizes a term, the words in the term are labeled with their associated concept from the ontology. The DER pattern matching automaton detects the term occurrences in $O(n)$ for a sentence with length of n words.

The concepts and members, which are recognized from a question q, are represented as slot-value pairs in its frame $f(q)$. For example, a domain term *'block heater'* from ontology $\mathcal{O}$ appears in the question q = *'What should I do if the block heater does not work?'*. From the ontology, *'block heater'* $\in T($*Block Heater*$)$ and *Block Heater* $\in M($*Car Part*$)$. Thus, we get the following representation by applying DER to q.

$$\text{DER}(q) = \text{What should I do if the [block heater]}_{\mathcal{O}:\textit{Car Part}} \text{ does not work?}$$

3.4.4 Domain-Aware Semantic Role Labeling

Domain-aware SRL (DSRL) provides accurate SRL parsing by interpreting domain entities as noun phrases. The DSRL algorithm processes the text using this pipeline: DER, part-of-speech tagging (POS), dependency parsing, and SRL parsing. The

DER tags are interpreted as noun phrases when applying POS tagging. This information transfers in the pipeline and prevents SRL from breaking the domain terms into separate phrases, or parsing a word within a domain term as a verb.

3.5 Rule-based Mapping for Domain Adaptation

3.5.1 Definition of PAS Frame

Using DSRL each sentence is parsed into a *PAS Frame*. The PAS frame consists of one or more *verb frame*s, each corresponds to one verb from the sentence. In a PAS frame, the slots are the semantic roles, and the values (slot fillers) are the phrases. The domain adaptation applies a set of mapping rules to the slot-value pairs of a PAS frame to map it to the domain slots. Figure 3.6 demonstrates a PAS frame representation for the question "How to reset the transmission while in Limp Home Mode?"

3.5.2 Criteria for Proper Mapping Rule Definition

In this method, we consider a single type of domain frame. Although all domain frames are similar, the slot-value pairs included in a domain frame change based on the question and its PAS frame. Domain slots and values are specified based on a set of mapping rules. The mapping rules themselves are defined based on the PAS

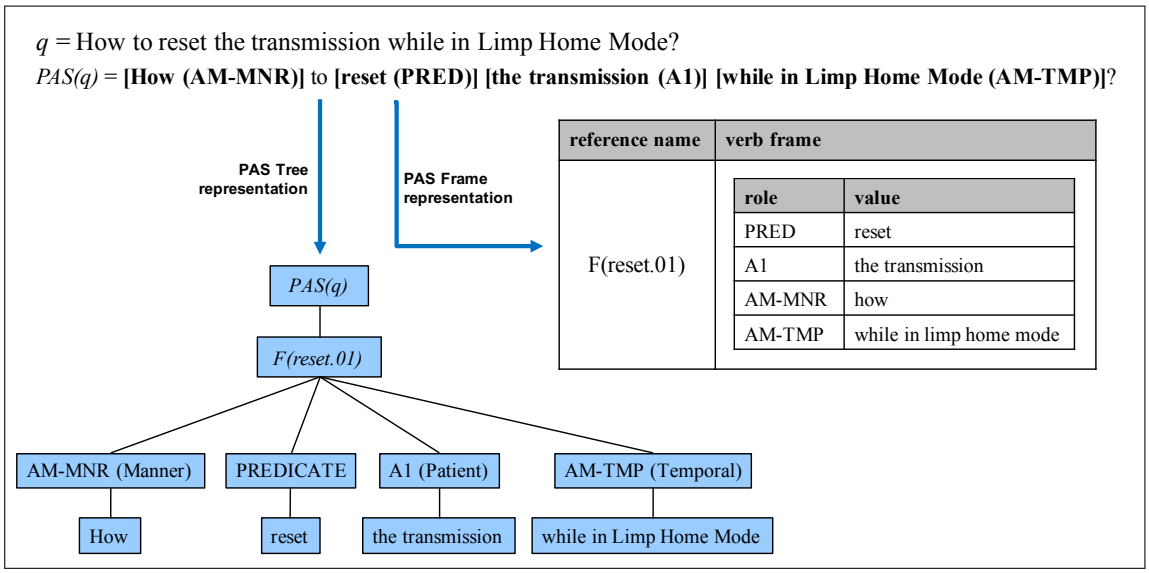


Figure 3.6: PAS Frame Representation Example

frames of questions within the q-a set. The following criteria are considered during *Mapping Rule Definition* for a successful domain adaptation:

- Selecting a *slot name* which is understandable in the context of the current question and the target domain.
- Defining generic mapping rules that extend to similar questions.

3.5.3 Definition of Verb Frame within a PAS Frame

Each PAS frame includes a number of *verb frame*s which are structured as a table with two columns, *Role* and *Value*. The role refers to the semantic role label (slot), and the value refers to the phrase parsed from text (slot filler). The mapping rules

are then defined in accordance to the content of verb frames obtained from the PAS frames. Figure 3.7 shows a PAS representation for the question "What should I do if the block heater does not work?". Figure 3.8 shows the resulting PAS frame generated for the PAS representation of the previous example. Within this example, it can be seen that the generated PAS frame includes two verb frames.

PAS('What should I do if the block heater does not work?') = {

do.01: [What (A1)] [should (AM-MOD)] [I (A0)] [do (PRED)]
[if the block heater does not work (AM-ADV)]?

work.01: What should I do if [the block heater (A0)] does
[not (AM-NEG)] [work (PRED)]?

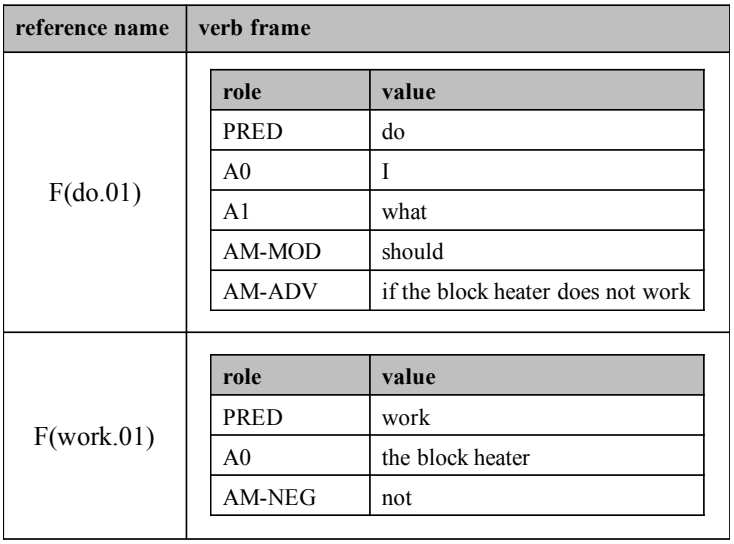
}

Figure 3.7: Example of a PAS representation from SRL parsing of a question

3.5.4 Format of Mapping Rules

A mapping rule specifies what *condition* in a PAS frame leads to *mapping* SRL roles to a domain slot-value pair. Hence, a mapping rule consists of two parts:

- The role-value matching *condition*, which specifies when a rule is applicable to a verb frame.
- The '*mapping template*' produced by the rule, which determines the domain



reference name	verb frame
F(do.01)	role: PRED — value: do role: A0 — value: I role: A1 — value: what role: AM-MOD — value: should role: AM-ADV — value: if the block heater does not work
F(work.01)	role: PRED — value: work role: A0 — value: the block heater role: AM-NEG — value: not

Figure 3.8: PAS frame containing multiple verb frames for the question "What should I do if the block heater does not work?"

slot and its value.

Mapping rules are defined by manual annotations on PAS frames from questions in the q-a set. The condition part of a rule can combine single role-checking conditions using `AND` and `OR` operators. This enables the definition of rules with more specific matchings. The mapping *rule-set* is created from the PAS annotations. Following this process, a set of mapping rules is generated which can be applied to any new PAS frame in order to map it to the domain frame.

The format of mapping rules is defined by a context free grammar (CFG). The CFG for mapping rules is shown in Figure 3.9. The first line of CFG describes that a mapping rule, `RULE`, extends to an `if` statement with the condition and mapping

parts denoted by `CONDS` and `MAPPING` non-terminal symbols. Lines 2 and 3 show how single conditions can be combined by using logical `and`/`or` operators. Line 4 shows `COND` extends to four types of single condition. Lines 5 to 8 shows which aspects of a single role-value is checked in each type of matching condition. A single condition can be based on matching an absolute value (`ABS_CND`), a keyword (`KWD_CND`), a phrase (`PHR_CND`), or a role (`ROL_CND`). The value of `MAPPING` can be assigned from the phrase in a semantic role (`value(ROLE)`), or a phrase explicitly defined in the mapping (`PHRASE`).

RULE → `if (` CONDS `) then` MAPPING
CONDS → COND `and` CONDS
CONDS → COND `or` CONDS
COND → ABS_CND | PHR_CND | KWD_CND | ROL_CND
ABS_CND → `value(`ROLE`) equals` PHRASE
PHR_CND → PHRASE `in value(`ROLE`)`
KWD_CND → WORD `in value(`ROLE`)`
ROL_CND → ROLE `is` ROLE_LABEL
ROLE → PAS.PRED | PAS.A1 | PAS.A2 | PAS.AM-TMP | …
ROLE_LABEL → PRED | A1 | A2 | AM-TMP | …
MAPPING → SLOT = SLOT_VAL
SLOT → PHRASE_UND
SLOT_VAL → `value(`ROLE`)` | PHRASE
PHRASE → WORD PHRASE | WORD | ε
PHRASE_UND → WORD '_' PHRASE_UND | WORD | ε

Figure 3.9: CFG grammar rules for the mapping rule-set.

3.5.5 Mapping Rule-set Development

The mapping rule-set is created from annotations on a PAS frames from a question in the *development set*. The development set can be defined as a collection of questions from the q-a set. In order to complete successful domain adaptation, the development set should cover all the subjects that appear in the q-a set. The PAS frames from the development set are then compiled into a Mapping Rule Template File (MRTF). The MRTF shows the PAS frames in a format which is suitable for adding annotations. The resulting annotations in the MRTF are then transformed into a mapping rule-set. All generated rules are in form of if-statements which follow the CFG grammar defined previously.

The *condition* (`CONDS`) of each rule defines the PAS annotations specifying what type of matching is applied to each semantic role. The mapping (`MAPPING`) of a rule defines the *domain slot* and its *value*. Domain slots within the mapping are explicitly defined as a phrase assigned by the knowledge expert. An example for this would be, assigning the phrases *Problem*, or *State* as domain slots within the car manual domain. The value of the domain slot is declared either by a slot value from the PAS frame, or a phrase directly assigned by the knowledge expert. Deriving the value of a domain slot using PAS slot values is more general, and can generate numerous slot-value templates. Using a directly assigned phrase is more limiting

since it generates only one mapping template.

The steps for annotating a PAS frame of a question are as follows:

1. Selecting the important slots from the PAS frame

2. Defining the matching condition based on the features of selected slots (i.e., role, keyword, etc.)

3. Specifying the domain slot-value pair which is produced by the rule

The aforementioned process is illustrated using two examples as follows. The first example is when the question 'What if tread wear indicators first become visible?' is present in the MRTF file. This question requires a rule for mapping the PAS frame to the domain slot *State*. The associated rule should match the argument [visible (A2)] from the verb frame F(become.01). Hence, the condition part of the rule is defined as below:

`if( PAS.A2 equals` 'visible' `) then ...`

The complete rule with the mapping template is shown below. This rule produces the domain slot-value pair *State* = *'visible'*.

`if( PAS.A2 equals` 'visible' `) then`

State `= value(PAS.A2)`

In another example, the question 'Why the anti-lock brake system does not work?' needs mapping to the domain slot *Problem*. However, the matching condition requires checking two roles from PAS: [work (PRED)] and [not (AM-NEG)] from F(work.01) frame. Hence, the condition part of the rule should be defined as below:

`if(` PAS.PRED `equals` 'work' `AND` AM-NEG `in` PAS `) then ...`

The complete rule for the second example with the mapping template is shown below. This rule produces the domain slot-value pair *Problem = 'not working'*.

`if(` PAS.PRED `equals` 'work' `AND` AM-NEG `in` PAS `) then`

Problem `=` 'not working'

PAS annotations make the rule definition faster by eliminating the need for writing the rules in the above form. Instead, the annotation process involves selecting the roles from the PAS frames and the type of matching (word, phrase, role, etc.) and then, defining the mapping pattern.

3.6 Slot Selection Strategy for Dialogue Management

This section offers an explanation on the SELECT algorithm for slot selection which was mention in Section 3.2.2. This algorithm can be applied to both adding and removing of a slot from the dialogue frame. The goal for the SELECT algorithm

is narrowing down possible matches for a dialogue frame to minimize the number of interactions in order to find K number of matching questions from the q-a set.

This algorithm utilizes the results obtained by matching the current dialogue frame to QUKB frames. A score is calculated for slots to be added/removed based on the distribution of the values of the corresponding slot in the matching result. The highest scoring slots are then selected for addition/removal and the dialogue manager initiates user interaction based on this selection. A formal definition of the SELECT algorithm is offered below.

A set Q can be defined as all the questions from our q-a set with each question shown as q_i:

$$Q = \{q_1, q_2, q_3, \ldots, q_{N_Q}\}$$

In QUKB a slot is represented by S_i, and its set of possible values are V_{S_i}. A filled slot is denoted with a slot-value pair:

$$\forall S_i \ , V_{S_i} = \{v_{i,1}, v_{i,2}, \ldots, v_{i,N_{S_i}}\}$$

$$\langle S_i, v_i \rangle \quad , (v_i \in V_{S_i})$$

A frame is a set of filled slots or slot-value pairs. The frame production for a text t (question or utterance) is shown as $f(t)$. Thus the frame can be shown as below:

$$f(t) = \{\langle S_1, v_1 \rangle, \langle S_2, v_2 \rangle, \ldots, \langle S_k, v_k \rangle\}$$

QUKB, which is denoted as Φ_Q, is created by applying the frame production to all

the questions in Q:

$$\Phi_Q = \{f(q_1), f(q_2), \ldots, f(q_{N_Q})\}$$

The text for user utterances during dialogue is shown as d to differentiate it from questions. The set of questions (their frames) matching the dialogue frame ($f(d)$) is shown as Φ_d. This is a subset of questions from QUKB which the dialogue frame is a subset of the frames of these questions:

$$\Phi_d = \{f(q_i) \mid q_i \in Q \wedge f(d) \subseteq f(q_i)\} \tag{3.1}$$

For a value v_j of a slot S_i, we define the *Containing Subset* of v_j in Φ_d as the subset of the matching questions which have the slot-value $\langle S_i, v_j \rangle$ included in their frame. The containing subset $c(v_j)$ is shown as below:

$$c(v_j) = \{f(q_i) \mid f(q_i) \in \Phi_d \wedge \langle S_i, v_j \rangle \in f(q_i)\} \tag{3.2}$$

Based on the definition of the containing subset for a value and its corresponding slot, we define the *Contribution Weight* for the value v_j as the function $w(v_j)$ below:

$$w(v_j) = \begin{cases} 0 & c(v_j) = \emptyset \\ \\ \frac{1}{|c(v_j)|} & otherwise \end{cases} \tag{3.3}$$

The contribution weight of a particular value is higher if it appears in fewer questions, and can uniquely identify one or a small number of questions. The score of a slot $score(S_i)$ is defined as the sum of the contribution weight from all its

values over the size of the matching subset Φ_d. Equation 3.4 shows how *score* is calculated.

$$score(S_i) = \frac{\sum_{v_j \in V_{S_i}} w(v_j)}{|\Phi_d|} \tag{3.4}$$

This definition prioritizes a slot which appears in a higher number of matched questions and has a higher diversity in its value distribution. Hence, getting the value for this slot narrows down the possible matching questions rapidly. The SELECT algorithm utilizes the *score* to rank the slots missing from the current dialogue frame. In Algorithm 1 the process of selecting a slot to be asked in IQA is shown.

Algorithm 1: $SELECT_{ask}$ algorithm describes choosing a slot to ask

Input : Φ_Q, Φ_d

Output: slot S_i to be asked

1 SLOTS := $\{S_i \mid S_i \in \Phi_Q \wedge S_i \notin \Phi_d\}$

2 SCORES := $\emptyset$ (empty map)

3 **foreach** S_i *in* SLOTS **do**

4 SCORES $\leftarrow \langle S_i, score(S_i) \rangle$

5 **end**

6 **return** $\arg\max_{S_i}$ SCORES(S_i)

When there is no matching question for the current dialogue d, a slot-value should be removed. In this case, the lowest scored slot is the prime candidate for removal from the dialogue frame. This removal is required to revise the user's intent

understanding and broaden the matching options regarding QUKB.

The important difference in calculation is that the complete set of questions is used to find a *Static Score* for slots shown as $score_Q(S_i)$. The score is static since it is calculated based on the complete set of questions and it is not altered based on the dialogue frame. We choose the slot with the lowest $score_Q$ to be removed from the dialogue frame.

$$score_Q(S_i) = \frac{\sum_{v_j \in V_{S_i}} w(v_j)}{|\Phi_Q|} \tag{3.5}$$

In static scores, the contribution weights $w(v_j)$ are calculated according to Q and Φ_Q. Algorithm 2 shows the process for selecting a slot to be removed from the current dialogue frame.

Algorithm 2: $SELECT_{remove}$ algorithm describes choosing a slot to remove

Input : Φ_Q, $f(d)$

Output: slot S_i to be removed from $f(d)$

1 SLOTS := $\{S_i \mid S_i \in f(d)\}$

2 SCORES := $\emptyset$ (empty map)

3 **foreach** S_i *in* SLOTS **do**

4 | SCORES $\leftarrow \langle S_i, score_Q(S_i) \rangle$

5 **end**

6 **return** $\arg\min_{S_i}$ SCORES(S_i)

4 Experiments and Evaluations

The evaluation of IQA systems involves testing the system over a set of queries and calculate the accuracy of the system on answering those queries correctly. However, this evaluation technique is used mostly in IR-based QA systems. Our IQA method includes the knowledge base generation for the set of questions, and the interactive dialogue for answering the user's questions. For evaluating each component in IQA, other experiments included to measure the effectiveness of each component in IQA as well. The *user satisfaction rating* can be directly measured by conducting user studies. Although, using performance evaluation heuristics, which are correlating to user satisfaction, is a more efficient option for measuring the system's performance [57]. The factors for measuring the performance can be classified as following:

Task completion success: the extent of success can be measured by evaluating the percentage of correctly answered questions. Also, the percentage of successfully completed subtasks is a factor which is correlated with the system's success. For example, the ratio of the slots in the frame correctly filled from the

queries is possible to be calculated.

Efficiency cost: represents the efficiency of the system in accomplishing the task described by the user. The number of queries asked by the IQA system to find the proper answer is a good measure. Some works focus on optimizing this number in interactive search problems [61], such optimization is not addressed in this work. However, we use this measure in our experiments for comparing the performance of different interactive approaches.

Quality cost: this measure represents the system's proper understanding which enables proper response to the user's query. One suitable factor showing this quality is the *slot filling error rate*. Other factors include rate of times when the system's following question or answers during the dialogue. (correctness of system's questions and answers) [40]

In addition to performance in question answering (precision), factors similar as above are used for evaluating the knowledge base generation framework and dialogue management.

4.1 Dataset and Specification of the IQA System

Three datasets of question-answer (q-a) pairs are used for building the QUKB knowledge bases. The first q-a set, *QA1-Chrysler* has 700 unique questions manually generated from a Chrysler car manual. The second q-a set, *QA2-Chrysler* has

165 questions from the same car manual, generated by an *Automatic Question & Answer Generation* method and curated to fix any style and grammar issues. The third q-a set, *QA3-Ford* is also generated with the automated process and includes 135 questions from a Ford car manual. These datasets are used throughout the experiments in this chapter.

As described in chapter 3, a dataset of questions is required for developing the mapping rules. The development set of questions, *QA0-Dev* has 306 questions, 86% of these questions are from the Chrysler manual, and 14% from the Ford manual. These questions are parsed using SRL and converted to the MRTF template file for defining the mapping rules. All the rules used in the experiments come from the annotations on this file. The rule-set is used commonly for generating the QUKB for all three sets of questions.

4.2 The Mapping Rule-set for Domain Adaptation

From the annotations on the development set, *QA0-Dev*, a set of rules has been defined and revised for the task of IQA on the q-a set about car manuals. SRL Parsing the questions in *QA0-Dev* created 555 different slots (role-value pairs) from the PAS frames. The result rule-set maps the PAS slots into 11 domain slots for building the domain-specific frames in our QUKB.

As mentioned in Section 3.5.4, the matching part of the rules contains four

types: absolute value (`ABS_CND`), keyword (`KWD_CND`), phrase (`PHR_CND`) and role matching (`ROL_CND`). Also, the mapping part has two types: direct mapping (using the value assigned to the argument) and indirect mapping (explicitly defining the phrase for the domain slot). Based on the PAS annotations from *QA0-Dev* 296 mapping rules are defined. The detailed information about the rule-set is shown in Table 4.1.

Total # rules	Mapping type		Matching condition type		
	direct	indirect	keyword & phrase	absolute value	role
296	**59**	**237**	**226**	**34** (29 predicate, 5 argument)	**36**

Table 4.1: Statistics of mapping rule-set from 306 questions in *QA0-Dev*

A sentence when parsed with SRL into PAS, can have more than one verb frames. In such cases, there is a main verb and other secondary verbs with their frames in the PAS. In the mapping rules, the PAS slots for the root predicates of the sentences (PRED slot in main verb frame) and the secondary predicates of the sentences (PRED slot in extra verb frames) are respectively used 67 and 38 times in the mapping rules.

4.3 Question Answering Performance Evaluation

For measuring the performance of QA, as the factor representing task completion success, we calculate the percentage of queries from a set of questions that are correctly answered by the IQA system. Four sets of questions were used in the IQA system, the three question sets mentioned previously and a set of additional queries which is defined based on the questions in *QA2-Chrysler*.

The set of queries was used for testing the robustness and flexibility of the QA when asking a variation of questions from the q-a set. The query set is generated with rephrasing the questions in three ways:

1. **Synonym**: using a synonym phrase for the entity appeared in the question.

2. **Sentence variation**: using a paraphrased version of the question, or paraphrasing the sentence.

3. **Incomplete query**: creating a question by removing a part of its information. The third type evaluates how effective is the IQA in resolving the missing information in the question using the knowledge base and frame representation.

The number of queries in this set is 100 with equal number of queries for each type of query.

To have a better perspective about how effective is our IQA system in finding the answer to questions, we compared our system with a number of Information Retrieval methods for finding the most relevant questions in the question set. The IR document weighting models which are used in our experiment include TF, TF-IDF, BM25F, and PL2.

In the IQA system one parameter is K, the number of matching question which is shown to the user during the dialogue interactions. We perform our experiments with K values 1, 3. Also, K is used in IR methods to determine how many of the highest ranked questions are presented to the user. For IR systems the average of precision at top K results ($MP@K$) over the given set of queries. The $MP@K$ is automatically calculated given the set of questions and the query set using the equation 4.1. This measure is also used for calculating the success rate in IQA system.

Performing user study for IQA is not feasible for all of the specified settings and parameters. To get the returned set of questions from IQA, an automated interactive dialogue is simulated by giving a sequence of queries in case of IQA's following questions. The simulated user answers the IQA system by providing the next query or utterance in its sequence. Also, it accepts the correct answer and ends the IQA dialogue. According to the CSID dialogue manager, if the number of matching frames are more than K, a new slot should be selected and asked from

the user in the dialogue to narrow the matching options.

$$MP@K = \frac{\text{\# of queries having answer @ top K returned items}}{\text{total \# of queries}} \tag{4.1}$$

The evaluation results, comparing the IR and our IQA approach is shown in Table 4.2. The IR models used the questions as the documents to build their indexing database. Increasing K shows improvement in the Mean Precision and better success rate for question answering. For two types of queries 'Sentence variation' and 'Incomplete query' the IR systems make more significant improvement than IQA system and pass the IQA performance. Therefore, it means the frame matching method can perform better than relevance ranking approaches. However, IQA is better capable of interacting with the user and returning a single result (MP@1), which exactly matches the requested information.

The average number of system's following questions in dialogue (turns) is calculated in this evaluation. This is a metric for the second type of evaluation factors, showing the *efficiency cost* in our IQA dialogue system. These values are shown separately for successfully answered queries (hit) and unsuccessful cases (miss) in Table 4.3. Average number of turns is higher in cases where the answering is unsuccessful (miss cases). The IQA asks more questions to find a match for the user's question. Also, the number of turns when the input question is incomplete is higher overall than first two types of query which is necessary. This is necessary

since the initial query cannot be matched to a question in QUKB without asking for additional information.

		IR with question set				Frame-based IQA		
Query type	MP@K	TF	TF-IDF	BM25F	PL2	Domain	PB+Ont	PropBank
Synonyms	MP@1	18.48	21.74	28.26	26.09	**79.35**	56.52	47.83
	MP@3	75.00	75.00	75.00	75.00	**81.52**	57.61	50.00
Sentence variation	MP@1	28.92	28.92	34.94	33.73	**75.90**	9.64	2.41
	MP@3	83.13	85.54	85.54	**86.75**	78.31	9.64	2.41
Incomplete query	MP@1	21.79	21.79	29.49	29.49	**60.26**	55.13	20.51
	MP@3	73.08	74.36	**76.92**	74.36	73.08	61.54	20.51

Table 4.2: Comparison of question answering performance between IR, using questions as documents and our IQA using QUKB and CSID dialogue management.

If we consider the combination of the *Question & Answer Generation* method with our Knowledge base IQA framework, the combined system is even more effective in answering questions in dialogue form. In Table 4.3 the IR models with the answer paragraphs from the car manual corpus are used for the question answering task and evaluated in comparison to the IQA system. The result shows that using a Q&A Generation with the IQA system can outperform the IR over the car manual corpus for the task of QA. The IQA performs significantly better than the IR equivalent systems even allowing a larger K parameter for the QA response.

		IR with answer set				IQA with QUKB		
Query type	MP@K	TF	TF-IDF	BM25F	PL2	MP@K	#turn(hit)	#turn(miss)
Synonyms	MP@1	18.48	21.74	28.26	26.09	**79.35**	1.123	3.632
	MP@3	50.00	47.83	46.74	43.48	**81.52**	1.053	3.941
Sentence variation	MP@1	28.92	28.92	34.94	33.73	**75.90**	1.159	2.000
	MP@3	57.83	59.04	62.65	62.65	**78.31**	1.108	2.111
Incomplete query	MP@1	21.79	21.79	29.49	29.49	**60.26**	1.638	2.935
	MP@3	51.28	50.00	57.69	56.41	**73.08**	1.368	3.857

Table 4.3: Comparison of question answering performance between IR, using answer paragraphs as documents and our IQA using QUKB and CSID dialogue management.

In Figure 4.1 the average number of steps are compared for our three frame generation methods. As we can see, using the domain information in form of ontology and domain adaptation for generating slot-value pairs related to the domain of application is useful for having a more efficient IQA dialogue. The required number of turns to get to a final result is the minimum for our domain frame generation among all the methods used in this experiment.

Table 4.4 shows the result related to the analysis of the number of turns required to answer the question using different frame generation methods in IQA.

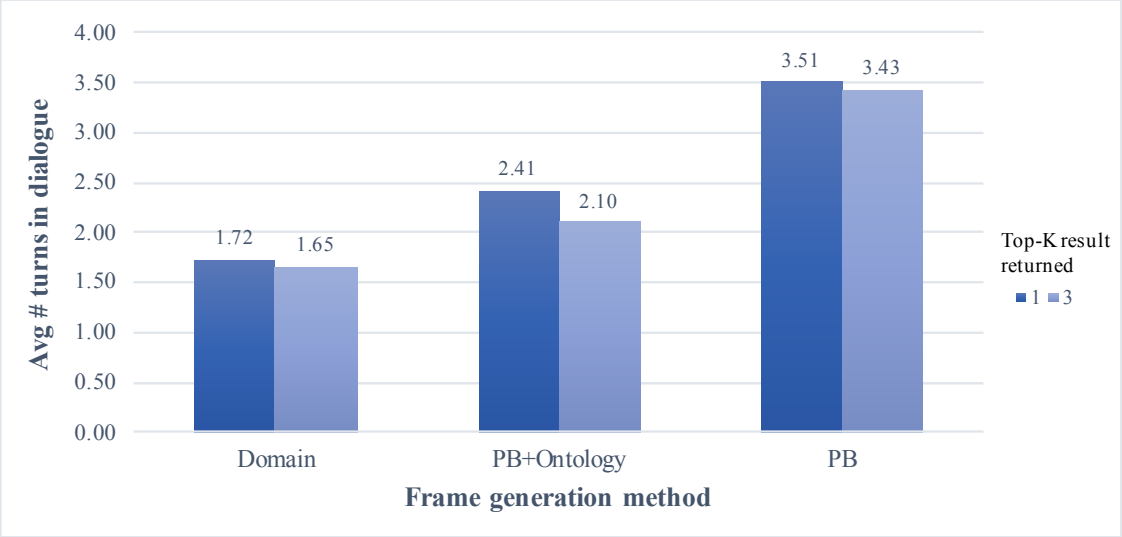


Figure 4.1: Comparison of average number of turns in IQA for different frame generation methods

4.4 Evaluation of Domain Terms Effectiveness in SRL

As mentioned in Section 3.4.4, one step in our method is Domain-aware SRL (DSRL). We use the information from the domain ontology to identify domain terms and use this information for better SRL parsing. Here, our DSRL is compared to normal SRL without the domain terms information to measure the effectiveness of DSRL in creating more accurate PAS frames.

The two version of SRL parsing is used to generate the PAS frames for the all the sets of questions, mentioned before which are used in the experiments. Since there is no ground truth frame annotations for the questions, a randomly selected

		Domain frame			PB+Ontology			PropBank		
Query type	Top-K	hit	miss	all	hit	miss	all	hit	miss	all
Synonyms	1	1.12	3.63	1.64	1.15	2.00	1.52	1.07	1.83	1.47
	3	1.05	3.94	1.59	1.13	2.03	1.51	1.00	1.87	1.44
Sentence variations	1	1.16	2.00	1.36	1.50	2.07	2.01	1.00	1.99	1.96
	3	1.11	2.11	1.33	1.50	2.07	2.01	1.00	1.86	1.84
Partial query	1	1.64	2.94	2.15	1.49	6.43	3.71	1.31	8.58	7.09
	3	1.37	3.86	2.04	1.19	5.33	2.78	1.00	8.58	7.03
All queries	1	1.31	2.86	**1.72**	1.38	3.50	**2.41**	1.13	4.13	**3.51**
	3	1.18	3.30	**1.65**	1.27	3.14	**2.10**	1.00	4.11	**3.43**

Table 4.4: Analysis of number of turns in IQA using different frame generation methods

sample of 100 questions from the set are evaluated manually. From this analysis, the estimated improvement in generated PAS frames is 17.2 % percentage. The improvement is defined as correction of at least one slot or value in the PAS frame.

More detailed evaluation result showing the increase in accuracy of slot-values is shown in Table 4.5.

Although, the trained SRL model used for parsing shows very high accuracy, using domain information is effective in increasing the parsing accuracy for a domain other than the training domain. The results shows 3.6 % and 5 % better accuracy in

		Identifying predicate	Identifying arguments
Accuracy %	SRL	94.26	92.60
	Domain-SRL	97.85	97.66

Table 4.5: Comparing accuracy of Domain-aware SRL and normal SRL in parsing accuracy

identifying respectively predicates and arguments for PAS frames in Domain-aware SRL parsing.

4.5 Evaluation of Domain Adaptation for PAS Frames

For evaluating the frames and slots generated by our method for building the QUKB knowledge base, we compared them to a baseline. The baseline method uses the frameset of PropBank [38] to label the PAS frames to get an understandable frame representation for the questions usable in QUKB and during the IQA task.

The frame is given a passing/failing score based on how complete, useful, and minimally represents the information in the sentence/question. To answer this qualities about a frame we answer the following questions and give a point for each one:

1. **Completeness**: "Are the important information pieces from the question are all captured and represented in the frame?"

2. **Minimal representation**: "Does the frame include few information pieces (less than 2 slots) with redundant or very long phrases of the question?"

3. **Usefulness**: "Does the frame include few slots (less than 2) that are *not useful* in the IQA process?"

For determining if a slot is *useful*, a passing score is given based on asking three questions about the slot to assess its usefulness:

1. **Understandable**: "If the question and the slot label are given, is it easily possible for a non-expert user to assign it the correct value from the question?"

2. **Slot abstraction**: "Is the slot name a suitable abstraction or conceptualization of its associated value?" In another word, "Is the slot name usable as a parent node in an ontology?"

3. **Correct filler value**: "Is the filler value representing a correct meaning for the slot based on the question and its label?" (If it is not filled with the right value then it is not useful for IQA and building the QUKB.)

These questions are answered by a human evaluator to specify the quality of frames apart from their performance in an IQA system experiment. Our three q-a sets are used to generate their frame representation. The frames for both method, PropBank frames and domain adapted frames are assessed according to above set of

quality questions. This evaluation is done manually on a randomly sampled subset of the question set. The summary of the evaluation is shown in Table 4.6 for slot representations and in Table 4.7 for frame representations.

	Slot Evaluation			
	Understandable	Slot abstraction	Correct value	**Acceptable slots**
PropBank	58.70%	7.61%	77.72%	**55.98%**
Domain-adapted	96.30%	98.77%	87.65%	**97.53%**

Table 4.6: Evaluation results for the quality of slot representations for IQA tasks

	Frame Evaluation			
	Completeness	Minimal representation	Usefulness	**Acceptable frames**
PropBank	70.97%	64.52%	19.35%	**64.52%**
Domain-adapted	32.26%	90.32%	90.32%	**90.32%**

Table 4.7: Evaluation results for the quality of frame representations for IQA tasks

The analyzed sample included 31 questions. Overall, the ProbBank and Domain-adapted frames contains respectively 184 and 106 slot-value pairs. Overall, the domain adaptation of frames using the rule-based mapping shows a significant improvement in providing an acceptable frame and slot representation. Using the rule-based mapping increases the precision in the generated slots. This is shown by the increased number for *Correct values* in slot evaluation. Some errors still occur

in the SRL parsing which is not accepted by the mapping rules. Hence, these errors are transferred to the domain frames. As reported in [30, 27] about the limitation of rule-based solutions in dialogue systems, one consequence is the lower coverage of the range of possible utterances (in our case, questions) and low recall problem. The decrease of frame completeness in our results can be explained similarly.

Using the domain ontology for generating slots showing the domain-specific entities, and the manually specified slot names in rules, are very effective in creating meaningful concepts from the slot fillers. The slots from domain frames in our evaluation shows a very high ratio of meaningful concepts and are assigned to meaningful phrases from the sentence, as shown by the numbers under slot evaluation for *Understandable* and *Slot abstraction*.

Also, the slot filling error rate is calculated for the domain slots in our evaluation. On the same subset of randomly selected question, we assessed if any domain slot should be added, removed, or replaced to get a complete domain frame representation. The following formula is used for calculation of the slot filling error rate:

$$\text{Slot filling error rate} = \frac{\text{\# Slot adjustments}}{\text{\# Correct slots}} \tag{4.2}$$

Based on this analysis, the rule-based domain building showed the slot filling error rate of 31.93%.

In our experiment, we used a fixed set of rules defined for the domain adaptation of the frames. The set of questions used in the experiments are different from the set

QA0-Dev and the rules are not tuned for them. Thus, the decrease in accuracy when measuring completeness for the frames and the slot filling error can be improved with adding more rules which address the PAS slots for the cases that the domain frames and slots are not suitable.

4.6 User Study of the IQA Method for Industrial Prototype

The proposed IQA framework in this work was used for building QUKB for the dataset *QA2-Chrysler*. A user study with two participants were conducted with experimental conditions set to emulate a real-world usage of the system by a normal user. The system is compared against an existing IQA framework from the company which involves higher amount of manual work and subjective decisions for knowledge base generation. The same search engine and dialogue manager were used for both knowledge bases to perform knowledge-based IQA. The user study was focused on queries related to 85 questions from *QA2-Chrysler*. The test cases included 11 different categories with 263 test cases in total. Each test case category was used to test one aspect of the possible variation which may appear in user's question. The accuracy of the systems measured for each category of test cases and overall for both systems using this formula:

$$\text{QA Accuracy} = \frac{\text{\# Correctly answered cases}}{\text{\# All test cases}} \tag{4.3}$$

The user study showed that the automatic KB was more successful in the tests by 17.01%, comparing to the manual KB. The performance in particular test types was improved up to 66.7% for 'non-registered synonym' and 41.7% for 'verb tense' categories. Also, the required manual effort for ontology and rule-set adjustments is reported to decrease by 45%, when a new set of q-a set from a different car manual is added to the knowledge base. The manual effort is calculated as the estimated number of manual tasks for adjusting or adding new rules and updating the domain ontology. The difficulty and time required for each task is defined equally for both systems. Moreover, the results showed the effectiveness of our knowledge base generation method when used for building frame-based dialogue systems for IQA. This method can be more efficient in terms of required manual effort for defining and adjusting the knowledge base. Also, the result IQA system improved the task completion success rate.

5 Conclusion

In this work, we addressed the problem of Question Answering in a restricted domain using frame-based dialogue for interactive answering. We used existing NLU techniques such as domain term extraction, named entity recognition, and semantic role labeling to build a frame representation from questions and user utterance. We proposed a framework for building the QUKB knowledge base from a set of questions and a dynamic control structure for dialogue management (CSID) using the dialogue frame and the QUKB information for IQA.

The knowledge base generation includes a procedure to create a domain ontology for entities. The ontology enables the domain-aware SRL algorithm for accurate parsing in text from a domain different from the domain that our SRL model is trained on. As shown by our experimental results in Section 4.4 of previous chapter, DSRL creates parsing with higher accuracy and is effective way of having a better performance from the SRL applied to a new domain.

Another important part of the knowledge base generation is our method for

domain adaptation of the generic frames from SRL parsing. A rule-based mapping of frames is used for domain adaptation in our framework. This method for domain adaptation is designed as an alternative to a previous subjective way of slot-values generation. Our method defines rules for the PAS structure instead of directly defining them for the text. Hence, the rules are applicable in more situations and more capable of capturing the semantic meaning from the PAS structure. Also, the process for rule definition is easier and removes some of the subjectivity from slot-value generation process. It is made possible because the PAS structure provides a general representation of the meaning in the text.

In our evaluations, we showed the effectiveness of our rule-based domain adaptation by measuring factors related to usefulness of the slot-value representation. The domain frames are more suitable for building the knowledge base and perform better in the task of IQA. Also, as expected the domain adaptation creates frames which represent the meaning in questions of our domain properly. This point is the result of the quality comparison of the domain frames and the generic PropBank frames. The proposed framework is also compared to a completely manual method for frame generation based on user study on the IQA task. The result of the user study showed improvement in the task success rate.

Our evaluation results showed that methods for question search like IR, can be effective in finding questions from a set which are related to the user's information

need. However, it is not as effective as conversational IQA when a single result is returned as the answer. However, the question search performance improves as larger number of result is considered as the answer (when $K > 1$). Hence, a system combining the question search and the frame matching for conversational interaction can improve the task performance especially when the frame information is not sufficient for matching the user query to the set of questions.

The contributions of this work can be summarized in the following points:

- We formalized a real-world problem in industry about creating a IQA system with conversational interface using slot-values. The problem is broken down into sub-systems and practical a solution for each part of the problem is specified. The methodology is designed using the available techniques from the research in Dialogue Systems and NLP. Also, we designed our unique solution for rule-based domain adaptation.

- We proposed a more objective method for knowledge base generation in comparison to the ordinary rule-based methods for NLU and frame generation. Suggesting ontology of terms and domain adaptation techniques for the generic SRL frames for achieving more meaningful and effective frames for a domain-specific IQA task.

- Designing an evaluation process for measuring the quality of slot-values and

frames. These evaluation metrics enable comparison of different semantic representations for any domain of application.

In the following part discusses the strong points about this work; the things that are useful and can be adapted for solving other similar problems:

In our framework, the amount of manual work for annotations decreased for adding a new q-a set to the QUKB. The reason of reduced work is that some of the existing rules are about the general semantic structure of questions. Thus, They are also applicable to similar questions. Defining the rules for the PAS structure results in more generalizable rules, when comparing it to directly defining slot-value from the text.

There is no training involved in our method for understanding questions. Therefore, the question understanding is possible without a large dataset of labeled questions for training. The rules are defined on a subset of questions are immediately applicable to new text without any training. However, rule-based methods suffer from low coverage (recall) of the space of possible utterance/questions. This means their ability for understanding all the paraphrased versions of a question is weaker than supervised models trained on large datasets. Our evaluation of the completeness of frames also agrees with this issue. One possible solution to improve the recall is combining this work with supervised machine learning.

Our unsupervised rule-based model can be helpful for creating a large volume

of labeled data required for training supervised models. The intuitions from few questions can be easily encoded into SRL mapping rules by a dialogue system expert who is familiar with the QA domain. After defining the rules and generating the knowledge base, the rule-based frame generation can be used for creating a larger set of questions, labeled with their domain slot-values. The supervised model learns more general patterns (instead of mapping rules) by training on the labeled q-a set. Without using our rule-based framework, creating all the labeled training examples is a more labour-intensive process.

5.1 Future Work

In Linguistics, there are other paradigms of meaning representation. In this work, we focused on Propositional semantics and PropBank form of annotation for building an unsupervised NLU for dialogue system. The reason for this decision was availability of trained models on large corpora from various domains for PropBank. These tools were capable of generating the SRL with desired level of accuracy which was not available for other linguistic approaches like FrameNet. A possible extension of this work may study generating the frame representation using SRL from other approaches with improved models for parsing text and compare the results to PropBank frames and domain-adapted frames.

The process described in this work for building the ontology involves manual

steps for concept labeling and ontology adjustment. One of the future directions is focusing on this process and improving it by eliminating the manual steps. One possible improvement is automatic assignment of labels to the concepts. The proposed concept identification is based on word embedding similarity. Another possible way for improvement is eliminating the manual adjustments by changing the concept identification. For example, using more sophisticated similarity metrics and clustering algorithm; introducing mechanisms for automatic filtering and pruning of concepts by measuring the coherency of concepts.

Simplification and elimination of the rules for domain adaptation is another priority in future works. This is because of the scalability issue with rule-based methods. Maintaining the rule-set and its efficiency and consistency becomes more difficult with larger, more complex set of rules. One way to achieve this is grouping semantically equivalent questions and selecting one or very few samples from each group for the rule definition and manual annotation to minimize the manual effort. Other suggestions include active learning for faster completion of the rule definitions (formulating the rule definition as a supervised learning task); a mechanisms for automatic validation of domain frames and suggest possible addition of the rules for modification of the frames according to the knowledge base of questions.

Bibliography

[1] Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. Finding high-quality content in social media. In *Proceedings of the 2008 international conference on web search and data mining* (2008), ACM, pp. 183–194.

[2] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. Dbpedia: A nucleus for a web of open data. *The semantic web* (2007), 722–735.

[3] Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[4] Baker, C. F., Fillmore, C. J., and Lowe, J. B. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1* (1998), Association for Computational Linguistics, pp. 86–90.

[5] Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., and Schneider, N. Abstract meaning representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse* (2013), pp. 178–186.

[6] Berant, J., Chou, A., Frostig, R., and Liang, P. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (2013), pp. 1533–1544.

[7] Berger, A., Caruana, R., Cohn, D., Freitag, D., and Mittal, V. Bridging the lexical chasm: statistical approaches to answer-finding. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (2000), ACM, pp. 192–199.

[8] Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., and Winograd, T. Gus, a frame-driven dialog system. *Artificial intelligence 8*, 2 (1977), 155–173.

[9] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data* (2008), AcM, pp. 1247–1250.

[10] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* (2015).

[11] Burke, R. D., Hammond, K. J., Kulyukin, V., Lytinen, S. L., Tomuro, N., and Schoenberg, S. Question answering from frequently asked question files: Experiences with the faq finder system. *AI magazine 18*, 2 (1997), 57.

[12] Chen, Q., Hu, Q., Huang, J. X., and He, L. Ca-rnn: Using context-aligned recurrent neural networks for modeling sentence similarity. In *AAAI* (2018).

[13] Chen, Q., Hu, Q., Huang, J. X., and He, L. Can: Enhancing sentence similarity modeling with collaborative and adversarial network. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), ACM, pp. 815–824.

[14] Chen, Y.-N., Wang, W. Y., and Rudnicky, A. I. Leveraging frame semantics and distributional semantics for unsupervised semantic slot induction in spoken dialogue systems. In *Spoken Language Technology Workshop (SLT), 2014 IEEE* (2014), IEEE, pp. 584–589.

[15] Choi, J. D. *Optimization of natural language processing components for robustness and scalability*. PhD thesis, University of Colorado at Boulder, 2012.

[16] Choi, J. D., and McCallum, A. Transition-based dependency parsing with selectional branching. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2013), vol. 1, pp. 1052–1062.

[17] Chotimongkol, A. *Learning the structure of task-oriented conversations from the corpus of in-domain dialogs*. PhD thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science, 2008.

[18] Cimiano, P. *Ontology Learning and Population from Text: Algorithms, Evaluation and Applications*. Springer, 2006.

[19] Collins, M., and Duffy, N. Convolution Kernels for Natural Language. *Proceedings of Neural Information Processing Systems 2001 - Advances in Neural Information Processing Systems 14* (2002), 625–632.

[20] Dang, H. T., Kelly, D., and Lin, J. J. Overview of the trec 2007 question answering track. In *Trec* (2007), vol. 7, p. 63.

[21] Dowty, D. Thematic proto-roles and argument selection. *language* (1991), 547–619.

[22] Duan, H., Cao, Y., Lin, C.-Y., and Yu, Y. Searching questions by identifying question topic and question focus. *Proceedings of ACL-08: HLT* (2008), 156–164.

[23] Dumais, S. T., and Belkin, N. J. The trec interactive tracks: Putting the user into search. *TREC: Experiment and evaluation in information retrieval* (2005), 123–152.

[24] El-Kishky, A., Song, Y., Wang, C., Voss, C. R., and Han, J. Scalable topical phrase mining from text corpora. *Proceedings of the VLDB Endowment 8*, 3 (2014), 305–316.

[25] Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence 194* (2013), 28–61.

[26] Jeon, J., Croft, W. B., and Lee, J. H. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM international conference on Information and knowledge management* (2005), ACM, pp. 84–90.

[27] Jurafsky, D., and Martin, J. H. *Speech and language processing*, vol. 3. Pearson London, 2014.

[28] Kolomiyets, O., and Moens, M.-F. A survey on question answering technology from an information retrieval perspective. *Information Sciences 181*, 24 (2011), 5412–5434.

[29] Kwok, C., Etzioni, O., and Weld, D. S. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS) 19*, 3 (2001), 242–262.

[30] Ligeza, A., and Nalepa, G. J. A study of methodological issues in design and development of rule-based systems: proposal of a new approach. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 1*, 2 (2011), 117–137.

[31] Litman, D. J., and Silliman, S. Itspoke: An intelligent tutoring spoken dialogue system. In *Demonstration papers at HLT-NAACL 2004* (2004), Association for Computational Linguistics, pp. 5–8.

[32] Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. Building a large annotated corpus of english: The penn treebank. *Computational linguistics 19*, 2 (1993), 313–330.

[33] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (2013), pp. 3111–3119.

[34] Molino, P., and Aiello, L. M. Distributed representations for semantic matching in non-factoid question answering. In *SMIR@ SIGIR* (2014), pp. 38–45.

[35] Mollá, D., and Vicedo, J. L. Question answering in restricted domains: An overview. *Computational Linguistics 33*, 1 (2007), 41–61.

[36] Munteanu, C., and Boldea, M. Mdwoz: A wizard of oz environment for dialog systems development. In *LREC* (2000), Citeseer.

[37] Noy, N. F., McGuinness, D. L., et al. Ontology development 101: A guide to creating your first ontology, 2001.

[38] Palmer, M., Gildea, D., and Kingsbury, P. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics 31*, 1 (2005), 71–106.

[39] Pieraccini, R., Levin, E., and Lee, C.-H. Stochastic representation of conceptual structure in the atis task. In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19-22, 1991* (1991).

[40] Polifroni, J., Hirschman, L., Seneff, S., and Zue, V. Experiments in evaluating interactive spoken language systems. In *Proceedings of the workshop on Speech and Natural Language* (1992), Association for Computational Linguistics, pp. 28–33.

[41] Ponte, J. M., and Croft, W. B. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (1998), ACM, pp. 275–281.

[42] Quarteroni, S., and Manandhar, S. Designing an interactive open-domain question answering system. *Natural Language Engineering 15*, 01 (2009), 73.

[43] Quillian, M. Semantic memory in m. minsky (ed), semantec informataon processing, 1968.

[44] Salton, G. *The SMART retrieval systemexperiments in automatic document processing*. Prentice-Hall, Inc., 1971.

[45] Salton, G., and Buckley, C. Term weighting approaches in automatic text retrieval. Tech. rep., Cornell University, 1987.

[46] Seneff, S., and Polifroni, J. Dialogue management in the mercury flight reservation system. In *Proceedings of the 2000 ANLP/NAACL Workshop on Conversational systems-Volume 3* (2000), Association for Computational Linguistics, pp. 11–16.

[47] Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603* (2016).

[48] Shen, D., and Lapata, M. Using semantic roles to improve question answering. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)* (2007).

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

[50] Song, L., Wang, Z., and Hamza, W. A unified query-based generative model for question generation and question answering. *arXiv preprint arXiv:1709.01058* (2017).

[51] Suendermann, D., Evanini, K., Liscombe, J., Hunter, P., Dayanidhi, K., and Pieraccini, R. From rule-based to statistical grammars: Continuous improvement of large-scale spoken dialog systems. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on* (2009), IEEE, pp. 4713–4716.

[52] Surdeanu, M., Ciaramita, M., and Zaragoza, H. Learning to rank answers to non-factoid questions from web collections. *Computational linguistics 37*, 2 (2011), 351–383.

[53] Tang, D., Duan, N., Yan, Z., Zhang, Z., Sun, Y., Liu, S., Lv, Y., and Zhou, M. Learning to collaborate for question answering and asking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), vol. 1, pp. 1564–1574.

[54] Traum, D. R., and Larsson, S. The information state approach to dialogue management. In *Current and new directions in discourse and dialogue*. Springer, 2003, pp. 325–353.

[55] Turney, P. D., and Pantel, P. From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research 37* (2010), 141–188.

[56] Voorhees, E. M., et al. The trec-8 question answering track report. In *Trec* (1999), vol. 99, pp. 77–82.

[57] Walker, M. A., Litman, D. J., Kamm, C. A., and Abella, A. Paradise: A framework for evaluating spoken dialogue agents. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics* (1997), Association for Computational Linguistics, pp. 271–280.

[58] Wang, K., Ming, Z., and Chua, T.-S. A syntactic tree matching approach to finding similar questions in community-based qa services. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval - SIGIR '09* (2009), 187.

[59] Ward, W., and Issar, S. Recent improvements in the cmu spoken language understanding system. In *Proceedings of the workshop on Human Language Technology* (1994), Association for Computational Linguistics, pp. 213–216.

[60] Webb, N., and Webber, B. Special issue on interactive question answering: Introduction. *Natural Language Engineering 15*, 1 (2009), 1–8.

[61] Yang, Y., and Tang, J. Beyond query: Interactive user intention understanding. In *Data Mining (ICDM), 2015 IEEE International Conference on* (2015), IEEE, pp. 519–528.

[62] Yang, Z., Hu, J., Salakhutdinov, R., and Cohen, W. W. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206* (2017).

[63] Yih, W.-t., He, X., and Meek, C. Semantic parsing for single-relation question answering. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (2014), vol. 2, pp. 643–648.

[64] Zelle, J. M., and Mooney, R. J. Learning to parse database queries using inductive logic programming. In *Proceedings of the national conference on artificial intelligence* (1996), pp. 1050–1055.

[65] Zhang, Y., Gan, Z., Fan, K., Chen, Z., Henao, R., Shen, D., and Carin, L. Adversarial feature matching for text generation. *arXiv preprint arXiv:1706.03850* (2017).

[66] Zhou, G., and Huang, J. X. Modeling and learning distributed word representation with metadata for question retrieval. *IEEE Transactions on Knowledge and Data Engineering 29*, 6 (2017), 1226–1239.