A

Ph.D Thesis

on

Evidence Gathering and Language Alignment in Question Answering

Submitted for partial fulfillment for the degree of

Doctor of Philosophy

(Computer Science and Engineering)

in

Department of Computer Science and Engineering

(February 2017)

Supervisor:
Dr. Namita Mittal

Submitted by:
Lokesh Kumar Sharma
(2013RCP9007)

MALAVIYA NATIONAL INSTITUTE OF TECHNOLOGY, JAIPUR

Declaration

I, Lokesh Kumar Sharma, declare that this thesis titled, "Evidence Gathering and Language Alignment in Question Answering" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Ph.D. degree at MNIT.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at MNIT or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this Dissertation is entirely my own work.
- I have acknowledged all main sources of help.

Lokesh

Date: 12.06.2017

Declaration Certificate

It is certified that:

1. I am satisfied that the thesis presented by **Lokesh Kumar Sharma**, worthy of consideration for the award of the degree of Doctor of Philosophy and is a record of the original bonafide research work carried out by him under my guidance and supervision and that the results contained in it have not been submitted in part or full to any other university or Institute for award of any degree/diploma.

2. I certify that he has pursued the prescribed course of research.

3. Plagiarism report enclosed.

(Dr. Namita Mittal)
Supervisor of the student
Department of Computer Sc. & Engineering
Malaviya National Institute of Technology, Jaipur
Date: 12.06.2017

Abstract

Currently, search engines are required to be intelligent as users expect an exact answer rather than a list of documents that probably contain the answer. State-of-the-art search engines employ inbuilt Question Answering (QA) systems which aim to extract an exact information than a list of relevant documents. The current QA systems suffer from the poor performance on complicated open-domain questions. To improve the performance of a QA system is a challenge. In this work, particular issues of QA systems have been focused to improve the overall performance of the QA systems. The complete study of QA systems pulled attention towards two phases of any QA system. First is question analysis phase and the other is answer extraction phase. The performance of a QA system is improved by Natural Language Question Alignment (NLQA) before question analysis phase and by a proper Evidence Gathering (EG) at answering extraction phase.

An evidence is a text snippet in the large corpus which supports the answer. Gathering proper evidence give a support to the extracted answer. The search engines or QA systems with EG part in it provides an answer along with a confidence of the answer. EG systems employ an Evidence Scoring (ES) algorithm to score the evidence and support the answer. For these many features are required to be extracted. These features include various lexical, syntactic, semantic and proposed structural features. The structural features are extracted from the dependency features of the question and supported documents.

Further, In Natural Language Alignment (NLA) a link is established between closely related words of two sentences. Two similar words like ‘plane’ and ‘airplane’ can be aligned but two aligned words like ‘wife of’ and ‘married’ are not similar. A user can ask questions in different manners and expect the same answer. Various alignment approaches have been proposed so far, and most of these are attempted on a word to word alignment. To increase the performance of QA systems it is required to align a natural language question with another question that is able to extract the final answer. In this work, NLA is used for two similarly asked questions which are different in their structure. For NLQA, various features and relations are extracted. These features are used to calculate a feature form score which is useful for learning algorithms. Furthermore, a Topic Word (TW) of the question is extracted to improve aligner accuracy. Results prove that for NLQA new structural features are better, and the accuracy of NLQA is increased when these are combined with TW and other features.

For an existing question-answer pair, to represent the question in the intermediate form, feature form score of the question is generated. An algorithm is designed to learn a particular question feature form and answer pair.

The contributions of this thesis are summarized as follows.

1) The features (includes structural features) have been proposed from the dependency parse and named entities are added to these features. The algorithms have been designed to extract the basic features (e.g. L_e, S_y, S_e).

2) The combined feature-based (with the combination of basic and proposed features) and reference-based (with the indirect co-reference in the text) algorithms have been proposed to gather promising evidence from the unstructured text.

3) An algorithm to calculate the distance among indirect question referents has been proposed for evidence gathering approach.

4) The feature-based and topic-based (with Topic Word of the question) algorithms have been proposed for question alignment of complicated questions.

5) Topic words are extracted using the Topic Word Identification algorithm and a domain word related to a topic word is retrieved from the Domain Word Identification algorithm. These are more useful for knowledge-based QA systems.

6) A named entity based recognition system is designed to enhance Gazetteer performance, and for Indian domain questions (e.g. KBC questions). These named entities are added to the structural features to improve the performance.

7) A question alignment learning algorithm is designed to learn the possible question alignments.

Keywords: Lexical features; Syntactic features; Semantic features; Structural features; Feature relevance; Question answering; Evidence gathering; Question alignment; Indirect reference; Topic word; Domain Word

Dedicated to my parents

Acknowledgements

I would express my heartily gratitude to my supervisor **Dr. Namita Mittal** for her enormous contribution in the realization of this thesis. She stood by me not only as a supervisor but also as a mentor. She taught me the art of writing research papers since I joined the institute. Apart from academia, she was always supportive during my Ph.D.; as many time I feel stressed and demotivated.

I am very much grateful to Malaviya National Institute of Technology Jaipur for providing me the opportunity to carry out my Ph.D. work in the Department of Computer Science and Engineering. I am thankful to **Prof. M.S. Gaur**, Head, Department of Computer Science & Engineering, all faculty members and technical staff, for the fruitful conversations we had all these years.

I am also thankful to all the members of my Department Research Evaluation Committee (DREC) **Prof. M.C. Govil**, **Dr. Girdhari Singh** and **Dr. E.S. Pilli** for giving their valuable time and suggestions in every semester during my progress presentations.

My special gratitude also goes to all the persons from whom, I have received critical reviews, **Prof. R.K. Agarwal** (Professor, JNU New Delhi), **Prof. Ela Kumar** (Professor, IGDTUW, New Delhi) and **Dr. Basant Agarwal** (Associate Professor, SKIT Jaipur).

I would like to thank the research scholars, especially Mr. Chandra Prakash, Mr. Vijay Kumar Sharma, Mr. Gaurav Takhar, Ms. Anubha Aggarwal, Mr. Abhinav and Mr. Vivek for helping me in my thesis proofreading. I would also like to thank all my colleagues Mrs. Namrata Saxena, Ms. Shobha Sharma, Mr. Amit and Mr. Vinay for helping me at my work place and providing me some free time to work.

I would like to thank my family for believing in me and supporting me with all possible means.

Amesh

Date: 12.06.2017

Glossary and Abbreviations

CA - Correct Answer

CD - Correct Document

DFF - Document Feature Form

DWI - Domain Word Identification

EAT - Expected Answer Type

EG - Evidence Gathering

IA - Incorrect Answer

ID - Incorrect Document

IFF - Intermediate Feature Form

IG - Information Gain

IR - Information Retrieval

JSON - Java Script Object Notation

KB - Knowledge Base

KBC - Kaun Banega Krorepati

MQL - Meta Query Language

NLA - Natural Language Alignment

NLP - Natural Language Processing

NLQA - Natural Language Question Alignment

ODQA - Open Domain Question Answering

QA - Question Answering

QFF - Question Feature Form

QFS - Question Feature Score

SQL - Structured Query Language

SVM - Support Vector Machine

TD - Textual Dependency

TREC - Text REtrieval Conference

TWI - Topic Word Identification

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Question Answering (QA) research (Ferrucci et al. 2010; Chu-Carroll et al. 2006; Pedoe 2014) is concerned with the retrieval of exact answers to the natural language questions from an unstructured textual corpus. QA systems (Ferrucci 2012; Mingyu et al. 2007; Quarteroni et al. 2009) have open-domain and domain specific problems. QA systems are not similar to document retrieval (Ponte 1998) systems as in QA systems a question is asked without language restrictions and the answers are specific and exact. Factoid questions are asked for concise answers, usually named entities (e.g. *'Which state of India is the largest by area?'*). List questions attempt to get a list of factoid answers (e.g. *'Who are the characters in the movie Bobby jasoos?'*). Definitional questions asking for relevant information on a provided topic (e.g. *'What is cloud computing?'*). The state-of-the-art QA systems (Moldovan et al. 2003; Lally et al. 2012) address complicated questions that require retrieval of an answer from multiple sources, for instance, *'What is the name of the Indian prime minister who took the decision over demonetization in 2016?'*.

IBM Watson (Ferrucci 2012; Lally et al. 2012) represents the state-of-the-art factoid QA system, which defeated two human champions in the Jeopardy! Quiz show in 2011. The Jeopardy! quiz show is similar to the very famous Indian quiz show KBC. By defeating the human champion, Watson draws the most prominent success for factoid QA systems so far. Many questions typically arose after Watson's success:

1. Did Watson solve the problem of all QA systems?

2. Can Watson really think for a question?
3. What has left for QA research after Watson has defeated humans?

The brief answers to these are: Watson did not solve the problem of all QA systems, and it has a limited capability to reason among evidence, but not to the level of thinking process as of the human brain.

QA research has been increasing with the Text REtrieval Conference (TREC) (Voorhees et al. 1999) which organizes an annual evaluation to encourage research in this field. The first stage in Open Domain Question Anaswering (ODQA) systems (Prager 2006; Pedoe 2014) is question analysis that requires a deep understating of natural language text. Some robust algorithms are needed to enhance the performance of QA systems. There are many phases in ODQA systems mainly; question analysis (Bishop et al. 1990; Yao et al. 2013(a)), document retrieval (Tao et al. 2010), and answer extraction (Severyn et al. 2013; Yao et al. 2013(b)). Initially, basic and the proposed QA features are selected from the unstructured text document. Basic features (Loni 2011) include many lexical features, syntactic features, semantic features and, proposed structural features are discussed in chapter 3.

Two parts of an existing QA system shown in Figure 1.1 have been focused and improved. One is before the question analysis stage and another is at document retrieval stage. The proposed Natural Language Question Alignment (NLQA) before question analysis stage and Evidence Gathering (EG) at document retrieval (Xing et al. 2006) stage uses the features available in a natural language question and relevant document. Proposed NLQA approach aligns two similarly asked questions with an already existing database of questions. Proposed EG approach provides a set of evidence features for scoring the most prominent option among several available options of a question. Thus, both of the proposed approaches, NLQA and EG improve the performance of an overall ODQA system (Peng et al. 2005; Tellez-Valero et al. 2010; Fader et al. 2013; Yao et al. 2014; Moldovan et al. 2013).

In Section 1.1 the motivation behind the need for NLQA and EG approaches is represented using a set of open-domain questions. The research gap in existing QA system is mentioned in 1.2, and the objectives and contributions of this thesis are compiled in

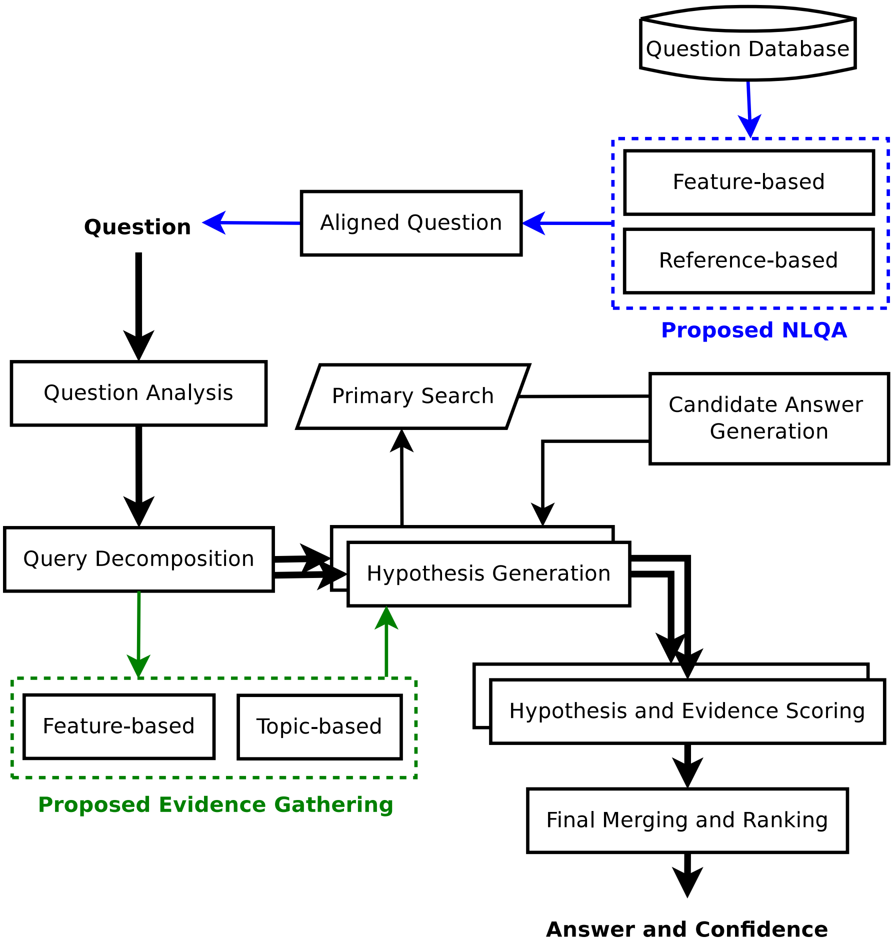


Figure 1.1: Existing question answering system with the addition of two proposed methods NLQA and EG

Section 1.3 and 1.4 respectively. Lastly, in section 1.5 the thesis structure is presented and the flow of chapters is shown in Figure 1.2.

1.1 Motivation

An ODQA system aims to find a concise answer to a natural language question. QA systems retrieve only the asked information (exact answer), unlike the search engines which pass on to the list of relevant documents. For example, given the question *'Which sportswoman was made the brand ambassador of the newly formed state of Telangana?'*, An ideal QA system would answer *Saniya Mirza'*. Therefore, QA system saves time and provides the required information which is accessible on all devices. There are several state-of-the-art QA systems which perform well but still not able to answer all of the open-domain questions. The performance of these modern QA systems can be improved by adding some efficient algorithms at any of the stages of QA, question processing, document retrieval and answer processing.

Question alignment and evidence gathering are the critical phases in question answering (QA), these enhance the performance of retrieving an answer to the natural language questions from an unstructured text. In QA, it is frequent to locally store and tag texts and questions in databases that give a large coverage of the similar questions required for a given question. For instance, knowledge bases are valuable resources for answering very frequent questions, and stored corpora give the relevant information about a domain-specific question. However, these resources may not always include the questions which can be mapped with all the natural language questions asked, and it may be hard to obtain the answers even with the availability of stored resources. For example, the answer in the text corpus may be available in a strange form of the keywords than the question. To improve the coverage of resources and to promote the answer extraction, the QA systems can be aligned automatically with additional related questions found in large, external corpora such as the Web.

Valuable datasets are extracted from the KBC quiz show and TREC evaluations. NLQA and EG are effective strategies for improving QA performance because for complicated questions state-of-the-art QA systems reach to the following types of failures:

1. **Question Analysis Failures**, i.e. the resources contain the answer but the question asked is very complicated to analyze. A similar question from the stored database can be aligned and processed further towards the question analysis phase. Proposed NLQA approaches can address these failures by aligning the complicated question with the similar simple question from the database.

2. **Evidence and scoring Failures**, i.e. the sources contain the answer but does not have proper evidence to justify the answer. Often, the possible reason behind these failures is insufficient evidence or the lack of algorithms to map evidence with the supporting answers.

1.2 Research Gap

Many QA systems (Pedoe 2014; Ravichandran et al. 2002; Reder et al. 1987; Burke et al. 1997) perform well if the top-ranked (top- 3 documents) relevant documents having a possible answer are available. Open-domain QA systems face following problems.

Learning-based methods (Cortes et al. 1995; Christophe et al. 2003; Goldberg et al. 1998; Rasmussen 2006) in QA express the question by creating a feature vector with the basic language features. The size of such a feature vector is large and holds parts of noisy and unnecessary features. Feature vector designed for classification is very sparse; therefore it degrades the performance of the overall QA systems. QA algorithms are helpful for appropriate wh-words (e.g. when, where and who). The probable reason for this is people frequently use more questions seeking for a person, location and time or date. Although, these wh-words help in finding the correct Expected Answer Type (EAT) (Razvan et al. 2010) for a question, therefore it is very much helpful.

A proper question analysis is a critical part of every QA system. The complicated question asked to a QA system is a problem to be considered and can be resolved by several ways. Knowledge-based approaches (Mihalcea et al. 2006) can be used which generate an intermediate representation of the question to be performed with already existing knowledge bases (e.g. Freebase) (Jonathan et al. 2013; Jonathan et al. 2014). The main difficulty with these approaches is their coverage because it is not possible to produce such an extensive knowledge base which can store information related to every domain. Recently available knowledge bases do not contain useful and sufficient knowledge which is lacking to get the final answer. Further corpus-based methods (Islam et al. 2008) essentially rely on the tagging of unstructured text. These approaches do not work well because the open domain documents are kept together and liked to each other. Besides, the methods based on methods need large tagging corpus to perform well. Knowledge-based methods (Mihalcea et al. 2006) work well only when the knowledge base is huge and contain accurate information. Knowledge-based methods have a significant benefit of not requiring extensive tagging corpus. It not only requires a one-time exercise of producing a broad knowledge base but require an extensive Topic Word tagging. However, feature-based algorithms with added Topic Words and indirect references enhance the performance of QA systems.

Therefore, two approaches have been proposed for improving the current QA performance. One is NLQA, and other is EG; the problems with the existing QA models are as follows.

1. Researchers have proposed many approaches for question analysis and most of these be based on the word to word alignment and finding the proper term replacement to reform the question for analysis. The existing approaches do not consider all of this information (e.g. alignment and representation in intermediate form) together. The proposed method in this thesis includes the intermediate feature form score using lexical, syntactic, semantic, and new structural features for QA.
2. The existing feature selection methods do not recognize the repetition in the features. These methods select the important features based on their use of the question property. Proposed feature selection focus on choosing the features for NLQA considering relevancy to the question and repetition.
3. Most of the existing evidence gathering approaches for QA focus either on event-based or temporal features. Long distance relations with the semantics, have not been analyzed together.
4. As most of the existing evidence gathering models use the supported evidence retrieval, which includes passage term match and skip-bigram methods to capture long distance relations. The proposed structural features designed from the dependency rules resolve the issue of relation extraction in the question (e.g. *'character'* and *'know'* are aligned).
5. Most of the existing QA systems do not consider the effect of aligning the question at question processing stage and also do not consider the effect of a proper evidence gathering approaches which have been done in this work.

1.3 Objectives

The problem with the performance of the existing natural language question answering systems is the improper representation of the question to be aligned and gathering irrelevant evidence to support the answer. The objective is to enhance the performance of the existing QA systems by combining the basic (semantic, syntactic, likewise) and proposed structural features. Also, this thesis aims to propose a feature relevance technique which assigns the relevance score to the features and chooses the prominent features by dropping the unnecessary features. This thesis proposes the natural language question alignment model for aligning two similarly asked questions and evidence gathering methods to get supported evidence.

The objectives of this thesis are,

1. To extract new features from the dependency relation with an addition of named entities and further designing the algorithms to extract the basic features and the proposed features.
2. To propose feature-based (based on combination of both basic and proposed features) and reference-based (i.e. indirect reference) evidence gathering approaches.
3. To design an algorithm to calculate the distance among indirect question referents used for reference-based evidence gathering approach.
4. To propose natural language question alignment methods using proposed features and Topic Word of the question, and an algorithm is designed to learn possible question alignments.
5. To design algorithms to extract Topic Word (using Topic Word Identification algorithm) and domain word (using Domain Word Identification algorithm) for KB based QA systems.
6. To design a named entity based answer extraction system to enhance the Gazetteer performance for Indian domain questions for example KBC questions.

1.4 Contributions

Find below the summary of the contributions from this work,

1. The structural features have been proposed, which are constructed using dependency rules of the question and document. The accuracy is improved by adding named entities to these proposed features.

2. Two Evidence Gathering approaches have been proposed; the first approach is based on the combination of basic and proposed features and the second approach relies on the indirect reference available in the questions (very rare) and in the documents.

3. The feature extraction algorithms for the basic and proposed features and also calculated the relevance of each feature by their individual answer extraction accuracy on QA systems.

4. Two Natural Language Question Alignment approaches have been proposed; the first is a feature-based approach and the second is a topic-based alignment approach. For the topic-based NLQA approach, two algorithms have been also proposed i) Topic Word Identification (TWI) and, ii) Domain Word Identification (DWI) algorithm.

5. A question alignment learning algorithm have been proposed for similarly asked questions using alignment index. A valuable KBC question dataset for alignment is also generated which include at least two similarly asked questions along with four options for a particular question.

1.5 Thesis Structure

The thesis structure and the flow of chapters are shown in Figure 1.2. Chapter 1 presents the issues in question answering systems. It further presents the motivation behind this research work. This chapter interprets the possible benefits and fundamental challenges in the area of question answering. A considerable number of researchers have shown their interest to work on question answering (Hovy et al. 2000; Hermjakob et al. 2001; li et al. 2006) problem because of its variety of applications in the information extraction and state-of-the-art search engines. Various kind of methods has been proposed in the literature, although, there are several reasons so that the performance of the question answering systems on open-domain questions is not up to the mark. These reasons are also discussed in this chapter. Finally, in this chapter, a proposal on the research objective and contributions are presented.

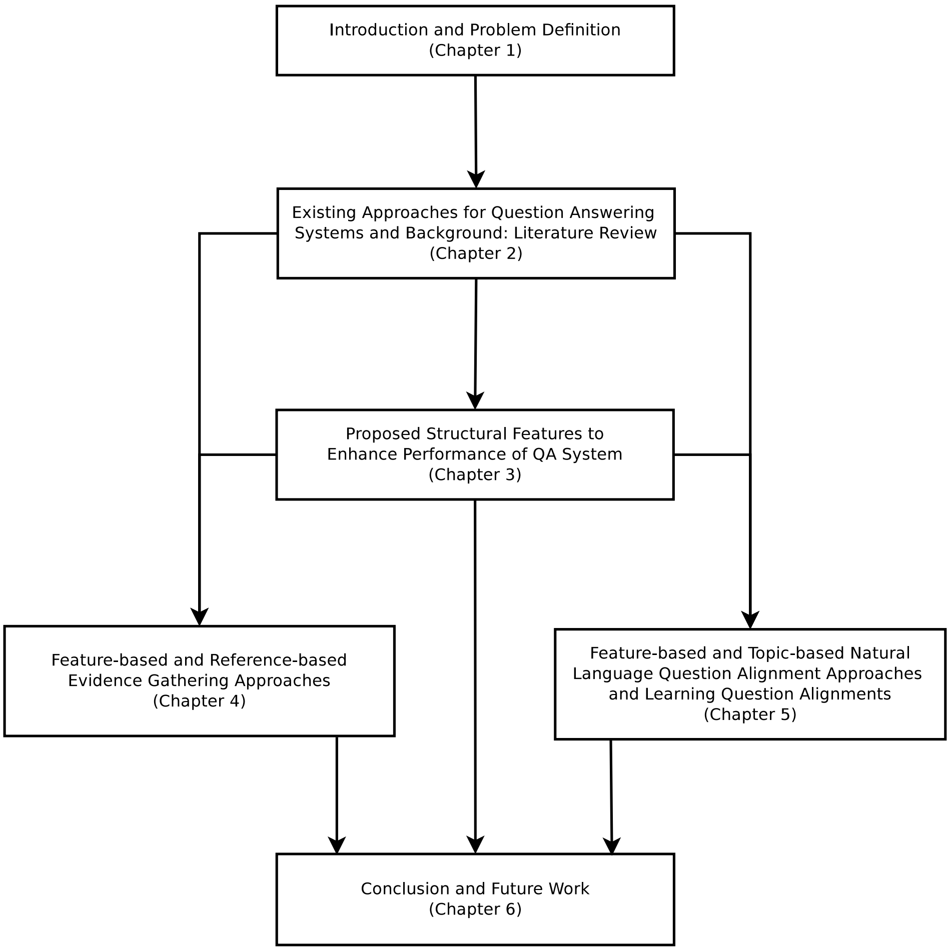

Introduction and Problem Definition (Chapter 1)
Existing Approaches for Question Answering Systems and Background: Literature Review (Chapter 2)
Proposed Structural Features to Enhance Performance of QA System (Chapter 3)
Feature-based and Reference-based Evidence Gathering Approaches (Chapter 4)
Feature-based and Topic-based Natural Language Question Alignment Approaches and Learning Question Alignments (Chapter 5)
Conclusion and Future Work (Chapter 6)

FIGURE 1.2: Thesis structure and the flow of chapters

Chapter 2 presents the background and literature of question answering systems are more than 50-years long and focused more on the start of TREC for question answering problems; various researchers initiate the research by using the semantic parsers in question answering.

Chapter 3 discusses the basic features used for answer extraction in question answering. In this chapter, we designed many new feature extraction algorithms for basic features and used feature selection methods and relevance techniques to measure the individual feature relevance. We propose novel structural features from the dependency rules of the text. Proposed design principles and named entities are added to these features to enhance the answer extraction performance. Later, in this chapter, we collected a valuable KBC dataset and experimented with different datasets (e.g. WebQuestions and TREC dataset) with the various categories of features. These include basic features (e.g. lexical, syntactic and semantic features) and proposed structural features and a combination of these features.

After finding the reason behind the poor performance of the question answering systems on open-domain, in chapter 4 we proposed two novel approaches for evidence gathering to enhance the performance of QA systems. The first approach is feature-based evidence gathering which uses the basic and proposed features of chapter 3. The second approach is reference-based evidence gathering which uses the existing pronominal anaphora resolution and proposed indirect reference based question referents. Later, in this chapter, we explain the experimental results with different categories of features including basic features, proposed structural features, and a combination of these features with the reference-based evidence gathering approach.

Furthermore, in this chapter, we consider how optimal structural evidence features are extracted. This chapter also describes the experiments with baseline supported evidence retrieval approach. Chapter 5 presents the second finding in which we proposed natural language question alignment approaches to enhance the performance of the QA systems. In this chapter, we used features discussed in chapter 3 and also proposed a topic-based question alignment approach. The combination of proposed feature-based and topic based approach increase the accuracy of question alignment which enhances the performance of the question answering systems. In topic-based approach, two methods have been introduced, includes Topic Word Identification and Domain Word Identification.

Finally, the thesis is concluded in chapter 6; this chapter compiles the findings and contributions of the work proposed in this thesis also indicate towards possible future work in the QA domain.

Chapter 2

Background of Question Answering: Literature Survey

Currently, the scope of mobile computing is increasing, and users prefer to use mobile as a searching device than the laptop and other devices.

The current organizations are using some state-of-the-art QA systems including, Apple's Siri, Google's Google Now, IBM's Watson. These organizations are user focused and therefore presents an excellent example of information extraction (Agarwal et al. 2014(a); William 1992; Salton et al. 1986; Chowdhury et al. 2003).

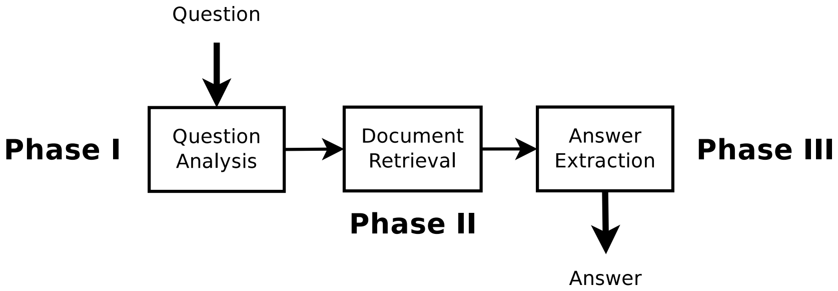


FIGURE 2.1: General question answering system

The Figure 2.1 is showing a brief architecture of a general QA system. A QA system has three stages i) question analysis, ii) document retrieval, and iii) answer extraction. Many researchers have focused these three steps as per their interest. The three stages have to be explored in the next section where we are discussing the ODQA system.

2.1 Open-domain Question Answering

The success of Watson noted the improvement in performance of QA (w.r.t. AI, IR, and NLP) systems. Watson has become the “showstopper” with great confidence, in the Jeopardy! Quiz show. Watson has thousands of algorithms for making hypothesis and gathering pieces of evidence (William et al. 2012; Chatterji et al. 2014). It has been discussed later in this chapter.

In the QA survey, three types of systems have been surveyed:

1. **IR based QA Systems:** These are focused on collecting the most relevant passage or document for answer extraction. Retrieving a relevant document is the first move for getting an answer. Besides search engines, systems that have been employed majority in the literature of QA are TREC QA: START and Indri (Strohman et al. 2005).
2. **NLP based QA:** In this, extracting the answer fragments from retrieved snippets (IR methods), with many semantic measures (Huang et al. 2011; Courtney et al. 2005) and machine learning methods (Yih et al. 2011).
3. **KB QA:** It is the task of getting answers from the knowledge base (structured data) (Hermjakob et al. 2000) than unstructured text. Intermediate query forms can be applied for such retrievals.

A complete QA systems and its three phase are discussed in next sections.

2.1.1 Question Analysis

The question processing starts with a question normalizer and analyzer (Durante et al. 1996; Swan et al. 2000; Scharpf et al. 1996). After the two-phase, a query based on the question is generated. Most of the terms in the Figure 2.2 are clear, and query generator sends the queries to the phase- II (document retrieval).

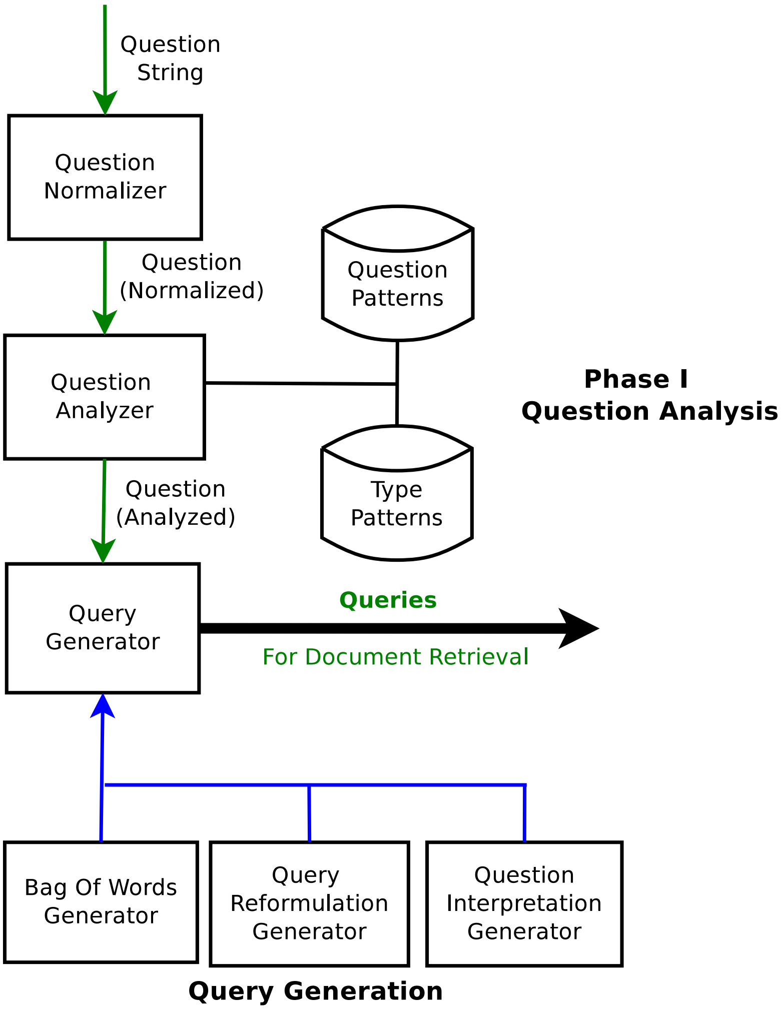


FIGURE 2.2: The complete question processing phase in a QA system

2.1.2 Document Retrieval

In Figure 2.3 it is shown that queries are coming from the Phase- I, and searched for top-ranked documents (Rocchio et al. 1971). The searching is to be done using Google and Indri knowledge miners (Strohman et al. 2005). The outcome of this is sent to the phase- III (answer extraction).

2.1.3 Answer Extraction & Selection

To extract the answer the output of document analysis (i.e. top-ranked relevant documents) becomes the input for answer analysis. These documents are used for answer extraction (Severyn et al. 2013). The Figure 2.4 is showing the filters used in answer extraction. There are some two significant findings from this:

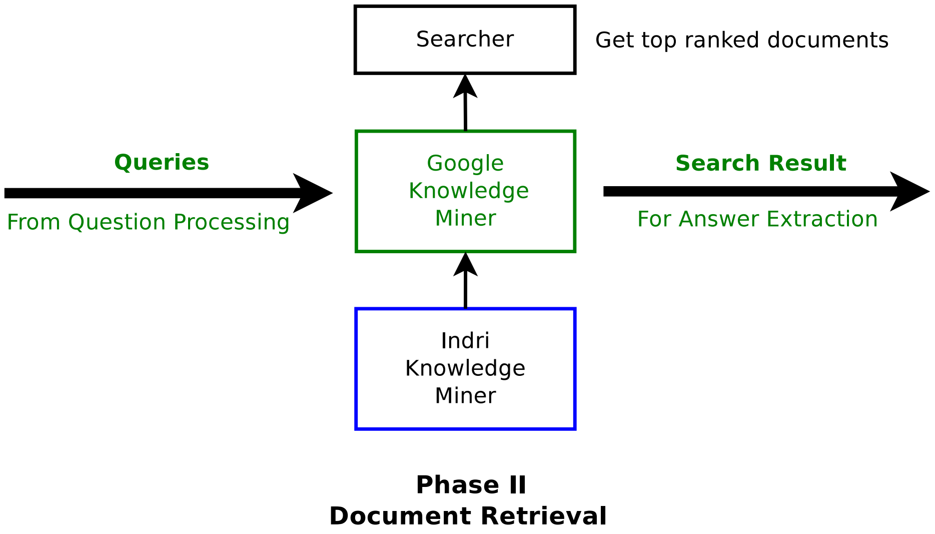


FIGURE 2.3: Document retrieval using google and indri in QA system

1. **Answer Extraction-** It has answer projection, answer type, answer pattern. Answer Type Detection (ATD) is impoertant.

2. **Answer Selection-** For this selection first; stopword can be removed. Question keyword is selected .

Most of the terms in the Figure 2.4 are very clear. Thus can omit the exact definition but use an example for illustration. Given the question: *"What is the height of Guru Shikhar?"* The corresponding *part of question* properties are:

Question phrase/word: what

Question focus: height

Question topic: Guru Shikhar

The outcome of Figure 2.3 can vary by selected knowledge miner technique. The Google techniques will supply the top ranked docs and can have a parameter of most relevance. So the document open more frequently can be chosen. Whereas, the Indri knowledge miner gives the document list from locally stored documents.

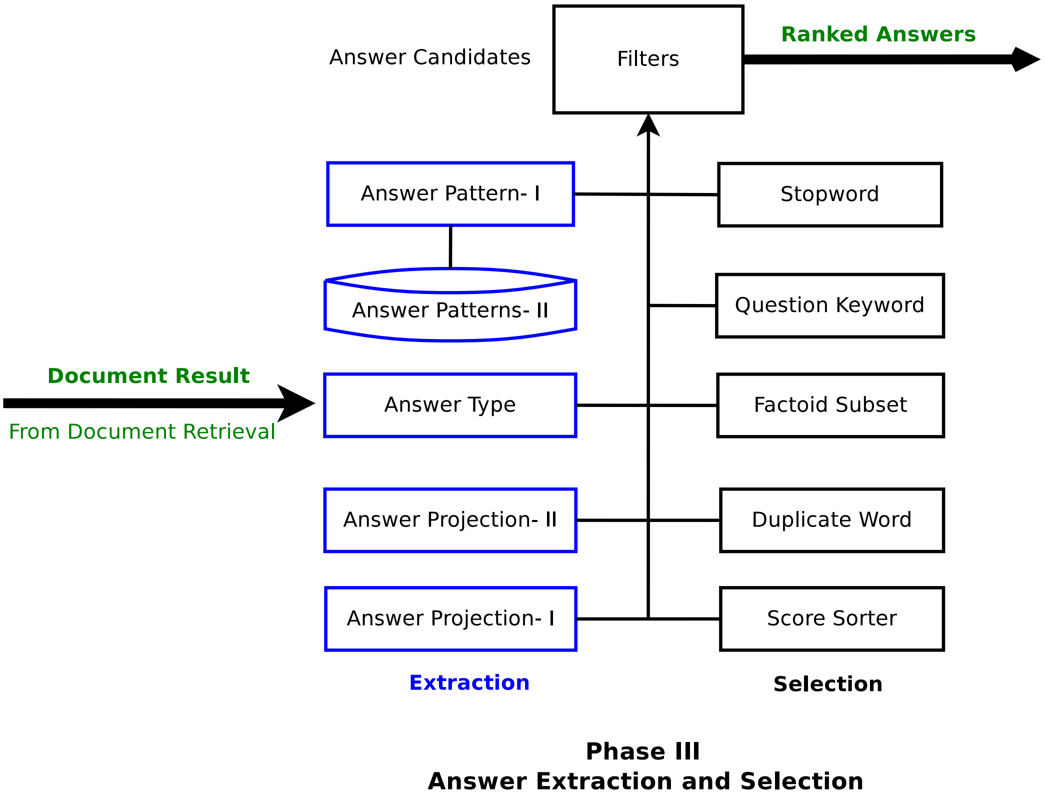


FIGURE 2.4: Answer extraction in QA

2.1.4 State-of-the-art QA Systems

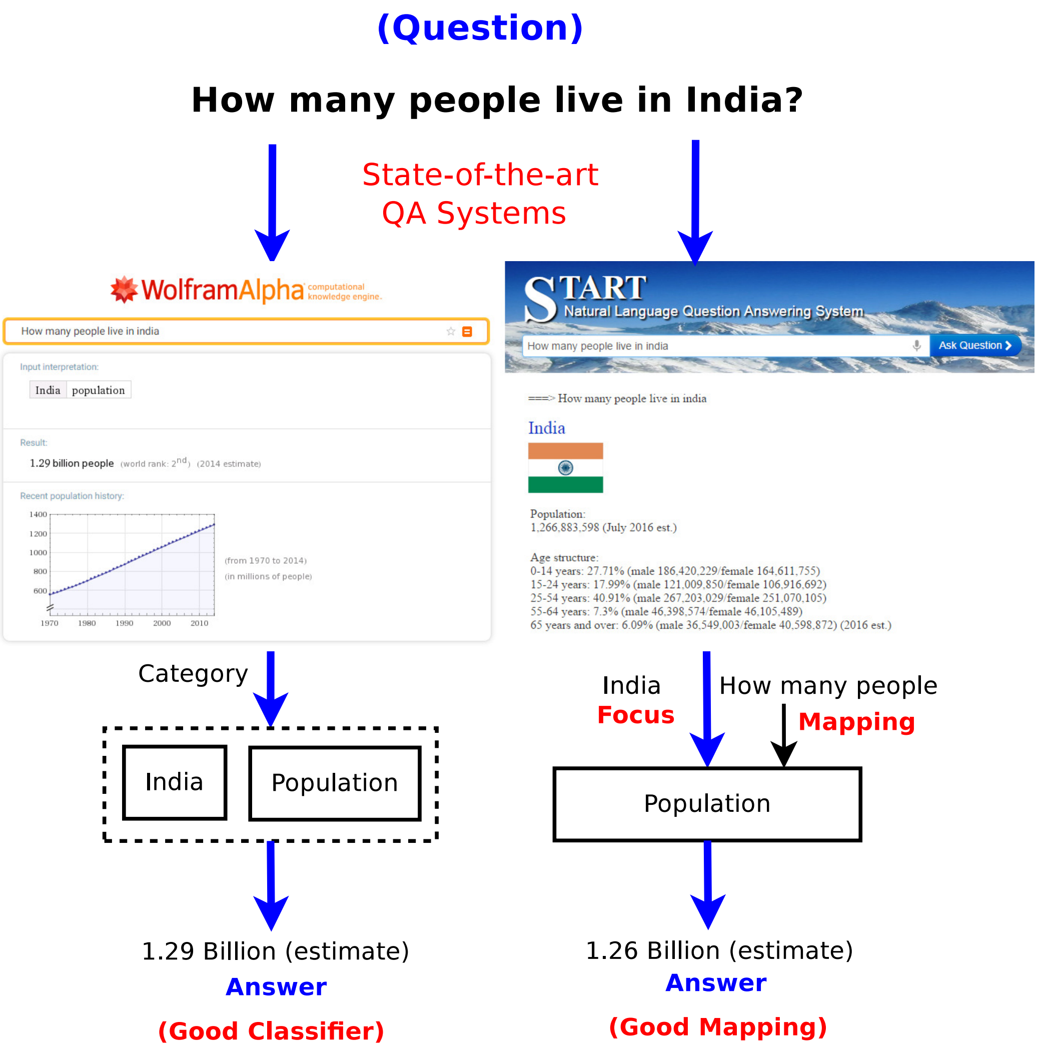
2.1.4.1 START and Wolfram Alpha

START (Katz et al. 2006) is the world's first online QA system is running fine on the web. It is supplying very accurate and exact result to user queries. START is aiming to provide the piece of right information to the user, instead of a relevant list of documents. This system can answer all questions related to cities, countries, etc. In Figure 2.5, the comparision of these START and Wolfram Alpha (Wolfram Alpha LLC 2017) is shown.

From Figure 2.5, it is clear that both the QA systems are able to answer a simple question, *"How many people live in India?"*. The Wolfram Alpha is a good classifier whereas, START gives a good mapping to the question to reach to the answer.

2.1.4.2 IBM Watson

After winning the Jeopardy! Watson has represented the most successful QA system so far. The proposed evidence gathering approaches in next chapter is compared with the



How many people live in india

Input interpretation:

India | population

Result:

1.29 billion people (world rank: 2nd) (2014 estimate)

Recent population history:

(from 1970 to 2014)

(in millions of people)

How many people live in india

===> How many people live in india

India

Population:
1,266,883,598 (July 2016 est.)

Age structure:
0-14 years: 27.71% (male 186,420,229/female 164,611,755)
15-24 years: 17.99% (male 121,009,850/female 106,916,692)
25-54 years: 40.91% (male 267,203,029/female 251,070,105)
55-64 years: 7.3% (male 46,398,574/female 46,105,489)
65 years and over: 6.09% (male 36,549,003/female 40,598,872) (2016 est.)

Figure 2.5: Comparing START with Wolfram Alpha QA system

Watson EG baselines.

An introduction of state-of-the-art QA system (Watson) is given in AI Magazine on "Building Watson" (Ferrucci et al. 2010). Later, the IBM Journal of Research and Development published a dedicated issue (Issue 3.4, May-June 2012) with about 20 articles themed "This is Watson". This section overviews Watson, with a focus of what Watson has done differently. The Watson project started in 2006 with adapting IBM's own in-house QA system, Piquant (Practical Intelligent Question Answering Technology). Piquant participated in TREC and was among the top 3 or 5 for several years.

This result led to a complete overhaul of their technical approach and architecture. The outcome was the extensible DeepQA architecture and the AdaptWatson methodology for rapid advancement and integration of core algorithms. By the end of 2007, the

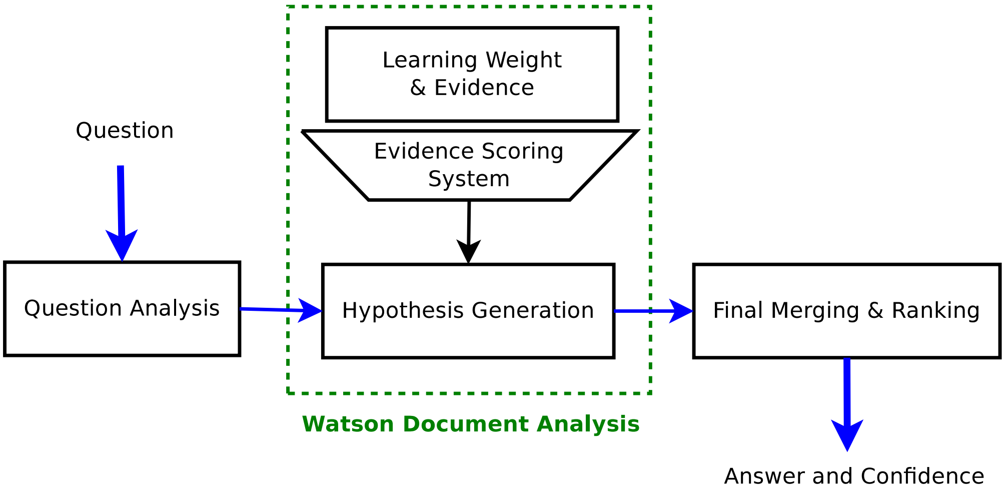


FIGURE 2.6: Thesis structure and the flow of chapters

DeepQA framework (Ferrucci et al. 2010) was implemented and reconstructed as the v0.1 version of Watson. In the next 5 years, the AdaptWatson method was employed over thousands of iterations of development, gradually pushing the confidence curve to more than 85%.

The DeepQA framework consists of more than 100 core algorithmic classifiers (experts), each of which was effective on some certain types of questions. Given the huge size of test data, it was hard to draw insight based on statistical significance when, say, a new classifier (Loni 2011) was added on top of 99 other classifiers. Thus there existed an internal baseline system, called Watson answer-scoring baseline (WASB). WASB includes most components in the DeepQA framework, such as question analysis, passage retrieval, and candidate generation, but only one evidence-scoring component based on answer typing. Answer typing was the most used and intuitive technology employed in most TREC QA systems. That is why it was included in WASB. Usually one expert was able to improve the accuracy of 2% to 5% over WASB. Then with a couple hundred of different exports, the full-fledged system was human-level competitive.

Knowledge Extraction- The output of ESG contains frames and slot fillers, which can be aggregated as strong evidence for certain tasks, such similar to that of Agarwal et al. (2014). All LATs detected from different components were in the end fed into a suite of type coercion algorithms (introduced later).

Deep Parsing- Watson used an augmented version of the English Slot Grammar. The Watson-specific parser produces a parse. In text search, it employed both Indri and Lucene for document and passage retrieval. One distinction between the Jeopardy questions and TREC QA questions (Hermjakob et al. 2002) is that some Jeopardy questions are very complicated with many constraints. Then it turned out the evidence had to be gathered from all over the documents, and thus document title serves as the answer. For instance, consider the question this country singer was imprisoned for robbery and in 1972 was pardoned, favors document retrieval over passage retrieval.

2.2 Evidence Gathering and Scoring in QA

The primary search in Watson generates a list of answer candidates (Aguado et al. 1998; Stuntz 1993; Brenner et al. 2001). But, they are not ranked at this point. Each of the answer candidates spawns a separate parallel process that includes this candidate in the search to retrieve further evidence. When evidence-bearing passages are returned, a passage-scoring component with four algorithms ranks them. The four algorithms are (Murdock et al. 2012; Markless et al. 2006; Chatterji et al. 2014):

1. **Passage Term Match-** This assigns a score by matching question terms to passage terms regardless of the corrections in its grammatical relationship or word order. There are no semantics in this technique.

2. **Skip-Bigram-** This assigns a score by matching pairs of terms that are connected or nearly connected. Important nodes are skipped in this technique.

3. **Textual Alignment-** This assigns a score by comparing the words and word order of the passage to those of the question with the focus replaced by the candidate answer. In this technique parsing is shallow.

4. **LFACS-** This assigns a score on the basis of how well the structure of the question matches with that of the passage, aligning focus to candidate answer. Dependencies are missing in this technique.

After all four algorithms are applied to each passage, the scores from each passage are merged with respect to each individual algorithm. Common ways of merging includes using maximum, sum, or decaying sum of scores from the passages. The best fit found by the Watson team was: summing for Skip-Bigram, decaying summing for Textual Alignment and Passage Term Match, and maximization for Logical Form.

A QA system aims to find a concise answer to a natural language question. QA systems retrieve only the exact information, unlike search engines that provide the list of relevant documents. For example, given the question “Who was the first Indian criketer to hit the double century in one day international?”, an ideal QA system would answer *‘Sachin Tendulkar’*. Therefore, QA system saves time and provides the required information which is accessible on all devices. To improve performance, recently QA has been drawing attention by using knowledge-bases. True Knowledge is a web-based QA system, and IBM’s Watson is state-of-the-art QA system which defeated the human champions in Jeopardy game show. Wolfram Alpha gives access to the world’s facts and data and calculates answers across a range of topics. Start is a system of recent era designed to answer to the questions asked in natural language.

After, Question Analysis and Candidate Answers Generation, Evidence Gathering and Scoring Component (Chatterji et al. 2014) is employed.

1. Search for the provided Answer Type (Answer Type Match Algorithm)
2. Search for keywords: (Passage Term Match (PTM) Algorithm Filter the Classified Passages)
3. Similar to Definition, Math, Factoid, List etc. Questions
4. Search the Focus Replacement word or Phrase
5. Relation Extraction (**wife** ← **married**)
6. POS Matching
7. Passage’s Dependency Parsing
8. Passage’s Named Entity Extraction and NE Matching
9. Headword Extraction Detection and Match

Gathering evidence which supports the answer to a particular question is an important part of QA (Jurczyk et al. 2007; Kanaiaupuni et al. 2000; Brill et al. 2002). Among three stages of QA as shown in chapter 2 in Figure 2.1 i) question analysis, ii) document retrieval and, iii) answer analysis. After question analysis and document retrieval QA systems employs a broad set of evidence and scoring section to compute the collected pieces of evidence. A useful dataset for this purpose, it has been discussed later in the section 6.5.1. The dataset has question-option-answer pairs of a famous Indian game show Kaun Banega Crorepati (KBC) which is similar to very famous game show Jeopardy. Supporting target documents for this dataset are collected manually from the web.

In the evidence retrieval, Supporting Evidence Retrieval (SER) (Murdock et al. 2012) is a system which put the probable answer into the primary question to make a proposition; and then it uses the DeepQA search techniques to retrieve the passages which are most closely related to the proposition. The scores (including all evidence-scoring components) during this phase are calculated and combined using a statistical model which is later used in answer ranking. The Indri passage retrieval algorithm (Strohman et al. 2005) is also used for finding and generating candidates. It uses a predicate-argument structure (PAS) for the syntactic portions of the graph using an English Slot Grammar (ESG) parse. The Skip-Bigram algorithm for evidence gathering was first introduced in for machine translation, in which the system translations are matched with gold standards. Both, the question and the passage text are mapped with their graph representation using a similar mapping procedure.

2.3 Natural Language Question Alignment in QA

In natural language, a same question can be asked in different ways but the answer for all of them should be same. For example, given the original question *'In medicine, which of these is usually denoted by 120/80 for an adult?'* and Similar question *'In medical science, 120/80 used for an adult?'*. An ideal QA system would answer *'Normal Blood Pressure'* for both questions. Therefore, in a QA system questions can be asked in many different ways which are having the same meaning directing these to the same answer. To enhance question alignment performance, QA systems are pulling attention by using Knowledge Bases (KBs). As shown in the Figure 5.1 in chapter 5, a user's query is converted into an intermediate form (e.g. MQL, SPARQL) which is further used to get the answer from knowledge bases. User's query can be different but their intermediate form should be same so that the query's answer remain same.

Many authors (Yao et al. 2013; Heng et al. 2009; Edgar et al. 2004; DeNero et al. 2008; Imamura et al. 2001; Taskar et al. 2005 Liu et al. 2005) have attempted the text alignment among these the MANLI aligner (Yao et al. 2013) is proposed to align P and H sentences for the task of natural language inference. It applies perceptron learning and handles phrase-based alignment of arbitrary phrase lengths. There are issues in the aligner after the lexical alignment phase, and with additional syntactic constraints, the specific alignment match rate for sentence pairs is significantly improved.

The semantic problems (Samuelsson et al. 2006; Robert et al. 2005; Altschul et al. 1990) are examined with the release by Microsoft Research (MSR) of human-generated alignment annotations (Cherry et al. 2003) for inferring from the Recognizing Textual Entailment (RTE) challenge (Bayer et al. 2005). This work is the first to utilize the generated KBC data for training and evaluation of NLA models (Yao et al. 2013; Heng et al. 2009; Edgar et al. 2004). The KBC data consists of a question set containing 500 aligned questions having at least two pairs for each.

The QA systems have a very long history. Many QA systems have been designed. Table 2.1 is showing the QA systems since they have been started in the history to the state-of-the-art QA systems. The accuracy of an ODQA system is not be seen as there dataset is open-domain. ODQA systems has an infinite range of possible questions and therefore it is very hard for these to predict the question structure and to find possible answer from an unlimited unstructured source.

Table 2.2 is showing the accuracy of question classification (Loni 2011) on various basic features used in this work. Later, in this work the accuracy of answer extraction with individual feature and features combined with proposed features is shown.

U: Unigrams, **B:** Bigrams, **T:** Trigrams, **NG:** N-grams, **WH:** Wh-word, **WS:** Word-Shapes, **L:** Question-Length, **P:** POS-tags, **H:** Headword, **HC:** Head-Chunk, **IS:** Informer-Span, **HY:** Hypernyms, **IH:** Indirect-Hypernyms, **S:** Synonyms, **NE:** NameEntities, **R:** Related-Words

We presented most of the state-of-art techniques employed for question answering. Two phases of QA systems have been explored for namely Evidence Gathering and Natural

QA System	**Authors (Year)**	Description
Baseball	Green et al. 1961	It is the first QA system in history. It read simple questions about baseball games from punched cards and printed lookup answers from its internal dictionary representation of knowledge
Lunar	Woods et al. 1977	It had operators for the words every and average and semantic scope, which looked like the first-order logic commonly used nowadays
Ladder	Hendrix et al. 1978	Language Access to Distributed Data with Error Recovery). What is the length of the constellation? Now, "of the nautilus". so, what is the length of the nautilus.
Planes	Waltz et al. 1978	Programmed LANguage-based Enquiry System) Question Answering system for a large relational database
BUC	Wilensky et al. 1988	Berkeley Unix Consultant) It is an intelligent, natural-language interface that allows naive users to learn about the UNIX operating system
Janus	Bobrow et al. 1990	Natural language interface that translates intentional logic expressions representing the meaning of a request into executable code for each application program, in order to provide an answer
State-of-the-art QA System		
Watson	David Ferrucci (2010)	DeepQA project. Thomas J. Watson Research Center, IBM
START	Boris Katz (1993)	Natural language question answering system, InfoLab Group, MIT
Wolfram Alpha	Wolfram Alpha LLC (2010)	Wolfram Research

Table 2.1: Brief overview of all QA systems available since the evalution of QA in NLP.

Features	**Author**	**Classifier**	**Accuracy**	
			Coarse	Fine
U + P + HC + NE + R	Li and Roth (2002)	SNoW	91.0 %	84.2 %
U + NG	Zhang and Lee (2003)	Tree kernel SVM	90.0 %	-
U + P + HC + NE + R + S	Li and Roth (2004)	SNoW	-	89.3 %
U + B + H + HY	Metzler et al. (2005)	RBF kernel SVM	90.2 %	83.6 %
U + B + T + IS + HY	Krishnan et al. (2005)	Linear SVM	94.2 %	88.0 %
U + B + T + P + H + NE +more	Blunsom et al. (2006)	Maximum Entropy	92.6 %	86.6 %
U + B	Merkel et al. (2007)	Language Modeling	-	80.8 %
U + L + P + H + HY + NE + S	Li et al. (2008)	SVM+CRF U	-	85.6 %
U + NE + S + IH	Pan et al. (2008)	Semantic tree SVM	94.0 %	-
U + WH + WS + H + HY + IH	Huang et al. (2008)	Maximum Entropy	93.6 %	89.0 %
U + WH + WS + H + HY + IH	Huang et al. (2008)	Linear SVM	93.4 %	89.2 %
U + H + HY + IH	Silva et al. (2011)	Linear SVM	95.0 %	90.8 %
U + H + HY + IH	Loni et al. (2011)	Linear SVM	93.6 %	89.0 %

Table 2.2: Brief overview of all features used in question answering are shown with their their accuracy of classification (on a similar dataset of TREC).

Language Question Alignment. Both the phases have their benefits in improving the performance of QA systems.The approaches perform well because structure of tokens changes with domain and context, and the available corpus which can provide prominant evidence and the tokens depending on the domain and context.

In Chapter 3, we will extract all the features used for EG and NLQA in question answering systems. We experiment many new feature extraction techniques and prominent structural features in addition to unigrams, bigrams, dependency features.

Chapter 3

Proposed Features and their Relevance in Question Answering

This chapter illustrates the idea of automatically collecting basic and new features from an unstructured text or a question. Firstly, few algorithms are designed to extract basic features. Further, some design principles and extraction algorithms are designed to extract new features from dependency parse of the question and the document. Prominent features are selected using feature selection techniques, and their relevance is decided using feature relevance techniques. These prominent features are further useful in Evidence Gathering (EG) and Natural Language Question Alignment (NLQA). In question answering task, in vector space model, a question (Q) is represented as (Equation 3.1):

$$Q = (f_1, v_1), (f_2, v_2), ..., (f_N, v_N) \tag{3.1}$$

Where, (f_i, v_i) is defined as i^{th} feature and value of the question Q whereas, N $\in$ total number of features in Q. The value v_i is calculated using the algorithms discussed in the section 3.2. Due to the large size of feature vector only non-zero valued features are kept in the feature vector. Therefore, the size of individual features is pretty small despite the large size of feature space. These features are categorized into i) Basic features, and ii) Proposed features. Feature extraction algorithms are designed for both basic and proposed features. The basic features including all the lexical features, semantic features, and syntactic features are added to feature space.

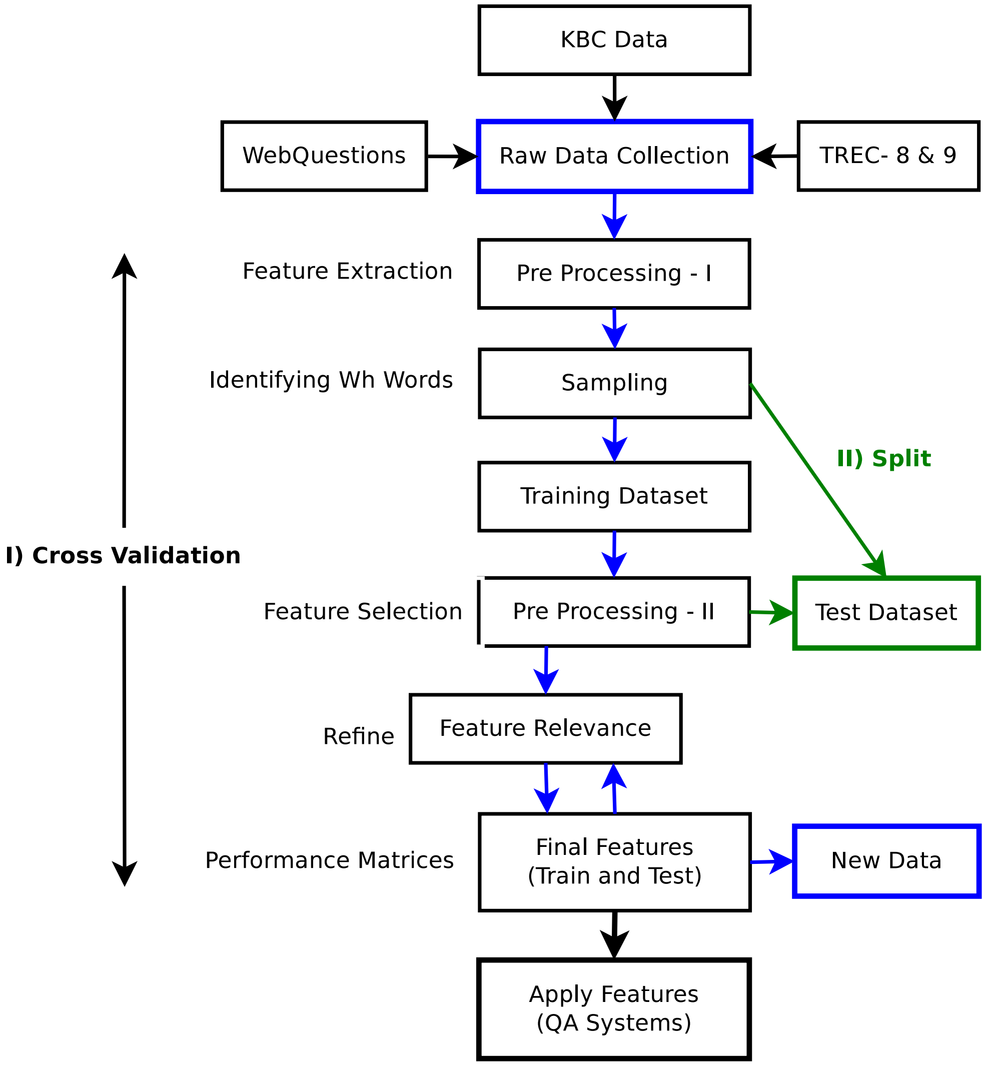


FIGURE 3.1: Raw dataset collection, features extraction, selection and new data generation for QA systems

The origin of these features and their extraction and selection procedure to create new dataset is shown in Figure 3.1. The Figure 3.1 shows that the data is taken from the KBC game show questions of a particular episode (especially season-5). Apart from the KBC, the TREC (8 and 9) (Voorhees 1998; Singhal et al. 2000) and WebQuestions (WQ) (Wang et al. 2014) datasets are also selected. In the first stage, preprocessing is done, and features are extracted using feature extraction algorithms, and after the sampling process the dataset it is split into training and test question dataset. These datasets are further processed to select the relevant features and scaling is performed on these features. After this, relevant features are selected for training and testing to produce the final model. These features are applied for a successful answer extraction in QA. In the next sections, the two categories of features are discussed in details.

3.1 Basic Features

Lexical Features- These are usually selected based on the words presented in the question. Just analyzing the single word as features is called unigram feature. Unigram is a particular case of the n-gram features. To extract n-gram features, a sequence of n-words in a question is counted as a feature. Consider for example the question *'Which sportsperson was made the brand ambassador of the newly formed state of Telangana?'* from KBC dataset. Basic features of the lexical category are shown in Figure 3.2.

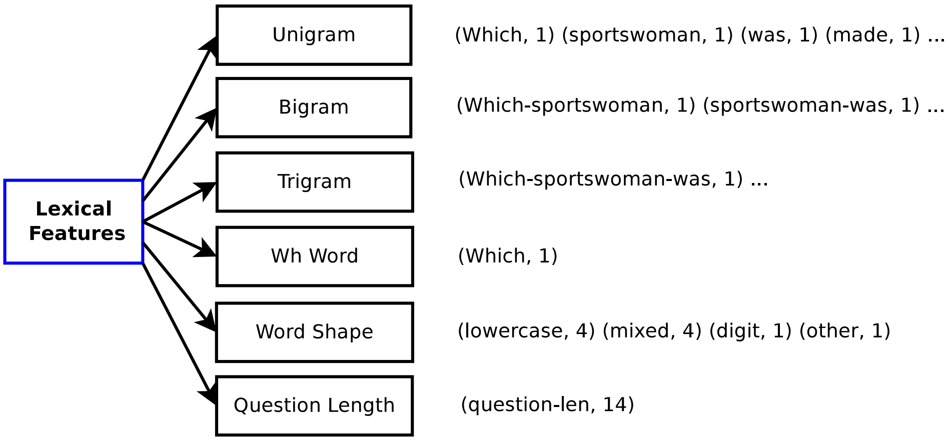

Lexical Features
Unigram
Bigram
Trigram
Wh Word
Word Shape
Question Length
(Which, 1) (sportswoman, 1) (was, 1) (made, 1) ...
(Which-sportswoman, 1) (sportswoman-was, 1) ...
(Which-sportswoman-was, 1) ...
(Which, 1)
(lowercase, 4) (mixed, 4) (digit, 1) (other, 1)
(question-len, 14)

Figure 3.2: Lexical features present in a KBC question

Feature space for unigram is: Q = (Which, 1), (sportsperson, 1), (was, 1), (made, 1), (the, 1), (brand, 1), (ambassador, 1), (of, 1), (newly, 1), (formed, 1), (state, 1), (of, 1), (Telangana, 1), (?, 1). The pair is in the form *(feature, value)*, only the features with non-zero values are kept in the feature vector. The frequency of the words in question *(feature values)* can be viewed as a weight value. It utilized this aspect to weight the features based on their importance. They joined different feature spaces with different weights. In their approach, the weight value of a feature space and the feature values (term frequencies) are multiplied. If any two consecutive words are considered as a different feature, then the feature space is extremely larger compared to unigram feature space and that demands larger training size. Therefore with same training set, unigrams perform better than bigrams or trigrams. In most of our experiments for answer extraction bigrams give better results than unigrams or other features.

Huang et al. (2008, 2009) examine a separate feature that is question's wh-words. They modified wh-words, namely which, how, where, what, why, when, who and remaining. For example, this feature of the question *'What is the deepest ocean of the world?'* is *'what'*. Considering the wh-words as a separate feature improves the performance of QA according to the experimental studies. The other kind of lexical feature is Word Shapes (W_s). It refers to possible shapes of the word: upper case, all digit, lower case, and other. Using word shapes alone is not a reliable feature set for question answering, but their combination with another feature improve the performance of QA.

The another lexical feature is question's length; it is a total number of words in the question. The features are represented in a similar way to the Equation 3.1.

Syntactical Features- The most basic syntactical features are Part of Speech (POS) tags and headwords. POS tags indicate such as NP (Noun Phrase), JJ (adjective), etc. The above mentioned the pos tags: *Which/WDT sportsperson/NN was/VBD made/VBN the/DT brand/NN ambassador/NN of/IN newly/RB formed/VBN state/NN of/IN Telangana/NNP*. A POS tagger obtains the pos tags of a question. In QA, all the pos tags of a question in feature vector can be added applied as bag-of-pos tags.

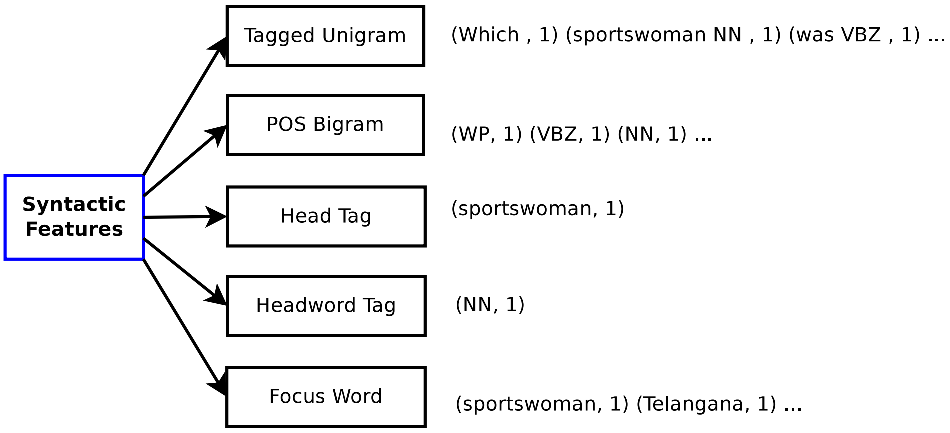

Syntactic Features
Tagged Unigram
(Which , 1) (sportswoman NN , 1) (was VBZ , 1) ...
POS Bigram
(WP, 1) (VBZ, 1) (NN, 1) ...
Head Tag
(sportswoman, 1)
Headword Tag
(NN, 1)
Focus Word
(sportswoman, 1) (Telangana, 1) ...

Figure 3.3: Syntactic features present in a KBC question

Some more feature namely tagged unigram which is a unigram expanded with part-of-speech tags. Instead of using common unigrams, tagged unigrams can help to identify a word with different tags as two separate features.

In syntactic features, headword is the most edifying word in a question or a word that represents the object that question attempts. Identifying a headword can improve the efficiency of a QA system. For example for the question *'Which is the newly formed state of India?'*, *'state'* is the headword. The word *'state'* majorly contribute to classifier to tag *LOC:state*. Extracting question's headword is challenging. The headword of a question frequently selected based on the syntax tree of the question. To extract the headword, it is required to parse the question to form the syntax tree. The syntax (parse) tree is a tree that represents the syntactical structure of a sentence base on some grammar rules. Basic syntactic features are shown in Figure 3.3.

Semantic Features- These are extracted from the question on the basis of the meaning of the words in a question. Semantic features require third party resources such as WordNet (Miller 1995) to get the semantic knowledge of questions. The most commonly using semantic features are hypernyms, related words, and named entities.

Hypernyms are the lexical hierarchy with important semantic notions using the Wordnet. For example, a hypernym of the word *'school'* is *'college'* of which the hypernym is *'university'* and so on. As hypernyms provide abstract over particular words, they can be useful features for QA. Extracting hypernyms is not easy as,

1. It is difficult to know the word(s) for which one need to find the hypernyms?
2. Which part-of-speech should be counted for focus word selection?
3. The focus word(s) expanded may have several meanings in WordNet. Which meaning is to be used in the given question?
4. Which level can one go to the hypernym tree to achieve the prominent set?

To overcome the problem of obtaining a proper focus word. The question can consider the headword as the focus word and it can be expanded for its hypernyms. All nouns in a question are considered as candidate words. If the focus word and the hypernym are same, this word can be expanded further. Consider the question again *'What is the most populated city in India?'*. The headword of this question is *'city'*. The hypernym features of the word with value six as the maximum depth will be as follow: (area, 1) (seat, 1) (locality, 1) (city, 1) (region, 1) (location, 1). The word *'location'* features, can contribute the classifier to categorize this question to *LOC*.

Named entities are the predefined categories of name, place, time etc. The available methods are applied to achieved an accuracy of more than 92.0% on determining named entities. For example for the question, *'Who was the second person to reach at the moon surface?'*, their NER system identifies the following named entities: *'Who was the [number second] person to reach at the [location moon surface]?'* In question answering

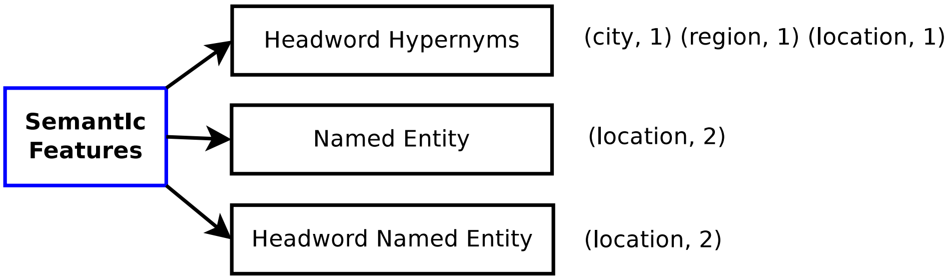


Figure 3.4: Semantic features in a KBC question

the identified named entities improves the performance when added to the feature vector. Basic features of a lexical category are shown in Figure 3.1. Apart from these basic features, proposed features for answer extraction are discussed in the next section. Figure 3.4 shows the basic semantic features.

3.2 Proposed Features and Feature Extraction Algorithms

The proposed structural features in a question are extracted from its dependency parse with additional Design Principles (DP), discussed later in details. These new features equally contribute in feature vector which is used for EG, NLQA and answer extraction in QA systems. Before going into details of the proposed features and their feature extraction rules, the feature extraction algorithms for basic features is discussed.

3.2.1 Algorithm to Extract Basic Features

Lexical features are easy to extract because these are obtained from the question, and no third party software (e.g. WordNet) is required. For a given question set (e.g. KBC) all lexical features are extracted from a features vector called lexical feature vector (Le_{fv}).

Algorithm 1 is showing the combined feature extraction algorithm for lexical features, and the accuracy of each is shown in Table 3.1 in the end of this section. Total 500 KBC questions are used to examine the feature extraction accuracy, and the algorithm attains 100% feature extraction accuracy for all lexical features.

Algorithm 1 Lexical feature extraction pseudo-code

```
INPUT: Question set (Q)
OUTPUT: $Le_{fv}$ : Lexical feature vector from Q
Variables used
$(Q_f, V) : (QuestionFeature, FeatureValue)$
$TF : TermFrequency$
$DL : DocumentLength$
$Q_{ui} : i^{th} Unigram$
$Q_{bi} : i^{th} Bigram$
$Q_{ti} : i^{th} Trigram$
$Q_i^{wh} : i^{th} whword$
$Q_i^{ws} : i^{th} wordshape$

for questions in dataset $Q$ do
    if $Q_f \neq$ '?' then
        extract lexical features of the question
    else
        repeat procedure
    end if
end for
if $L_e$ extract a unigram then
    $(Q_{ui}, V_{ui}) \leftarrow Unigram_i$
    $V_{ui} = \frac{TF}{DL}$
    $V_{ui} \leftarrow FeatureValueofi^{th}Unigram$
    if $L_e$ extract a bigram then
        $(Q_{bi}, V_{bi}) \leftarrow Bigram_i$
        $V_{bi} = \frac{TF}{DL}$
        $V_{bi} \leftarrow FeatureValueofi^{th}Bigram$
        if $L_e$ extract a trigram then
            $(Q_{ti}, V_{ti}) \leftarrow Trigram_i$
            $V_{ti} = \frac{TF}{DL}$
            $V_{ti} \leftarrow FeatureValueofi^{th}trigram$
        end if
    end if
end if
if input is $Q_i$ $(i^{th}question)$ then
    extract $Wh - word$ and $Wordshape$
    $W_w \leftarrow WhWordList$
    $(Q_i^{wh}, V_i^{wh}) \leftarrow \sum WhWord_i$
    $(Q_i^{ws}, V_i^{ws}) \leftarrow WordShape$
    $V_i^{ws} = WordShape$
    $V_i^{ws} \leftarrow FeatureValueofi^{th}WordShape$
    if $QL \neq 0$ then
        $QL \leftarrow QuestionLength$
    end if
end if
Return $\rightarrow$ $LexicalFeatureVector$ $(Le_{fv})$
```

1. *check_Q* (*Question_Termination*) (Lines 1 to 7 in Algorithm 1)

2. *extract_NG* (*N_grams*) (Lines 8 to 22 in Algorithm 1)

3. *extract_WhWs* (*Whword_Wordshape*) (Lines 24 to 29 in Algorithm 1)

4. *extract_QL* (*Question_Length*) (Lines 30 to 32 in Algorithm 1)

Syntactic features are quite difficult to extract because these features are extracted from the question and also require a third party software (e.g. WordNet). For a given question set (e.g. KBC) all syntactic features are extracted and placed into the features vector, it is called a syntactic feature vector (Sy_{fv}).

Algorithm 2 Syntactic feature extraction pseudo-code

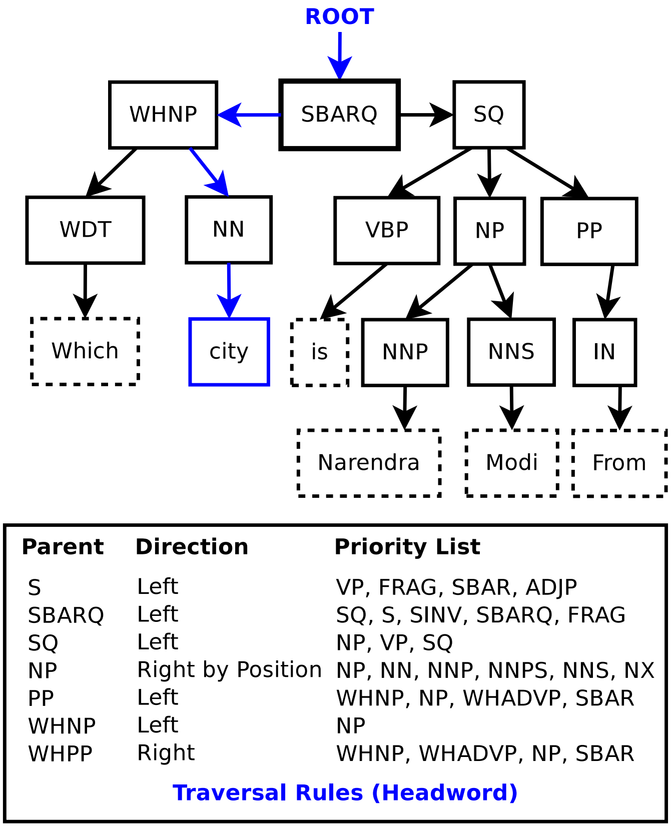
INPUT: Question set (Q)
OUTPUT: Sy_{fv} : Syntactic feature vector from Q
Variables used
Similar to lexical algorithm

```
for questions in dataset $Q$ do
    if $Q_w \neq$ '?' then
        extract syntactic features of the question
    else
        repeat procedure
    end if
end for
if $S_y$ extract a tagged_unigram then
    $(Q^t_{ui}, V^t_{ui}) \leftarrow TaggedUnigram_i$
    $V^t_{ui} = \frac{TF^{tu}}{DL}$
    $V^t_{ui} : FeatureValueofi^{th}TaggedUnigram$
    if $S_y$ extract a POS_tags then
        $(Q_{pi}, V_{pi}) \leftarrow Stanford\_tagger$
        if $S_y$ extract a Headword then
            $(Q_{hi}, V_{hi}) \leftarrow Headword$ (using headword extraction algorithm)
            $(Q^t_{hi}, V^t_{hi}) \leftarrow HeadwordTag$
            $(Q^h_{hi}, V^h_{hi}) \leftarrow Wordnet$ ($Q^h_{hi} \in$ headword hypernym of i$^{th}$ word )
        end if
    end if
end if
if input has multiple headword) then
    $(Q^w_{fi}, V^w_{fi}) \leftarrow focusword_i$
end if
Return $\rightarrow$ SyntacticFeatureVector ($Sy_{fv}$)
```

Algorithm 2 is showing the combined feature extraction algorithm for syntactic features, and the accuracy of each of these features is shown in Table 3.1 in the end of this section.

1. *check_Q* (*Question_Termination*) (Lines 1 to 7 in Algorithm 2)
2. *extract_TP* (*Taggedunigram_Postag*) (Lines 8 to 20 in Algorithm 2)
3. *extract_H* (*Headword*) (Lines 14 to 17 in Algorithm 2)
4. *extract_FW* (*Focus_Words*) (Lines 21 to 23 in Algorithm 2)

The tree traversal rules shown in Figure 3.5 are implemented to get a headword which is used as a significant syntactic feature of the question. The accuracy of this headword extraction algorithm is 94.3% on KBC questions as the traverse rules are formulated manually.



Parent	Direction	Priority List
S	Left	VP, FRAG, SBAR, ADJP
SBARQ	Left	SQ, S, SINV, SBARQ, FRAG
SQ	Left	NP, VP, SQ
NP	Right by Position	NP, NN, NNP, NNPS, NNS, NX
PP	Left	WHNP, NP, WHADVP, SBAR
WHNP	Left	NP
WHPP	Right	WHNP, WHADVP, NP, SBAR

Traversal Rules (Headword)

Figure 3.5: Tree traversal rules for headword

The headword extraction algorithm in Algorithm 3 is using the traversal rules shown in Figure 3.5. Babak loni (2011) uses these traversal rules for headword extraction for question classification. For example, for the question *"Who was the first man to reach at the moon?"* the headword is *"man"*. The word *"man"* will contribute for getting the Expected Answer Type (EAT). Few examples are showing the headword of a question. The words in bold are the possible headwords,

Algorithm 3 Headword extraction algorithm

```
procedure Extract_Tree
if $isTerminal(tree)$ then
    return tree
else
    root_node ← apply-traversal-rules (tree)
    return Extract-Question-Headword
end if
end procedure
```

What is the nation **flower** of India?

What is the name of the **company** launched JIO 4G in 2016?

What is the name of world's second longest **river**?

Who was the first **man** to reach at the moon?

Now, Table 3.1 is showing the accuracy of feature extraction on basic (lexical) features. The feature extraction accuracy of all the syntactic and semantic features is 100%. The headword is only syntactic feature headword having an feature extraction accuracy 94.3% (as discussed in Figure 3.5) and it can be improved using machine learning methods.

TABLE 3.1: Accuracy of lexical feature extraction algorithms

Accuracy of Lexical Feature Extraction Algorithm Total No. of Question = 500			
Lexical Feature	**Features Extracted**		**Accuracy**
	Correct	**Incorrect**	
Unigram (U_n)	500	0	100%
Bigram (B_i)	500	0	100%
Trigram (T_r)	500	0	100%
Wh-Word (W_w)	500	0	100%
Word Shape (W_s)	500	0	100%
Question Length (Q_l)	500	0	100%

3.2.2 Proposed Structural Features

The proposed structural features are obtained from the features in the yield of dependency parse. These structural features (say, S_t) are employed for complicated relations presented in similar questions and used for the uniqueness of efficient constants available in parsing results. The Question Feature Form (QFF) produced for a question contains one composite feature function. Structural features allow the model to adapt for all questions used for alignment using the question structure. Figure 3.6 is showing the structural features available in a KBC question.

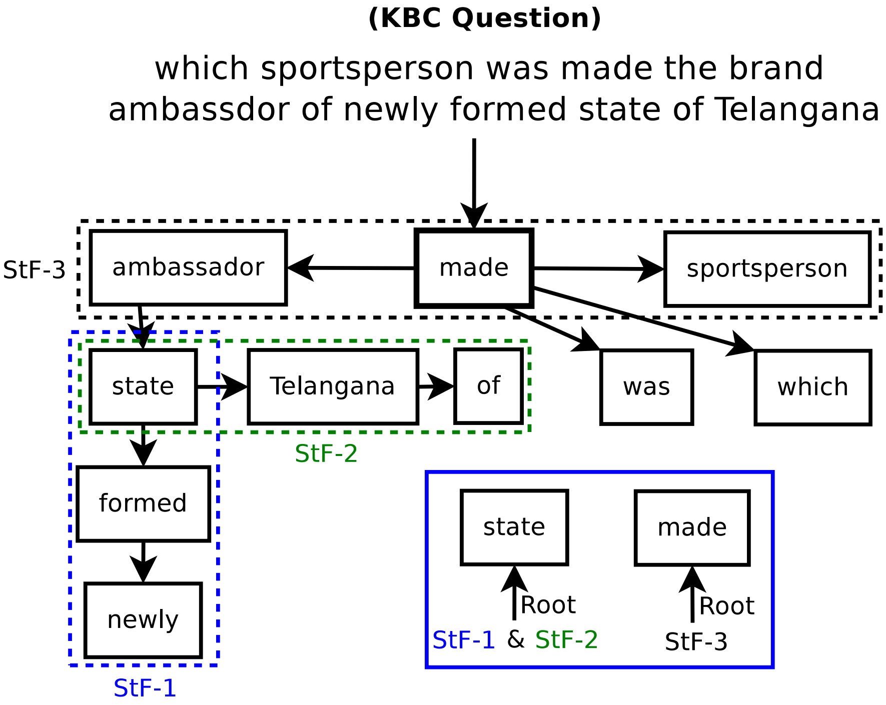


Figure 3.6: Structural features in a question used to align two words

There are some relations where *'state'* can be aligned with *'newly formed'* and *'of Telangana'* and another structural feature where *'made'* is aligned with *'ambassador'* and *'sportsperson'*. The link between *newly-formed* → *Telangana* and *state* → *made* cannot be identified directly. The connection provides a structural confirmation which has been described in details later in this section.

3.2.2.1 Dependencies Rules for Structural Features

Researchers in different domains have successfully used dependency Rules (DR) or Textual Dependencies (TD). In Recognizing Textual Entailment the increase in the application TD is distinctly apparent.

It is found that the rules are designed from the dependencies in the extraction of a relation between question and document, a system with DR semantics considerably outperforms the previous basic features on KBC dataset (by a 9% average improvement in F-measure). The tree-based approach uses a dependency path to form a graph of dependencies. The system those uses TD demonstrates improved performance for the feature-based techniques.

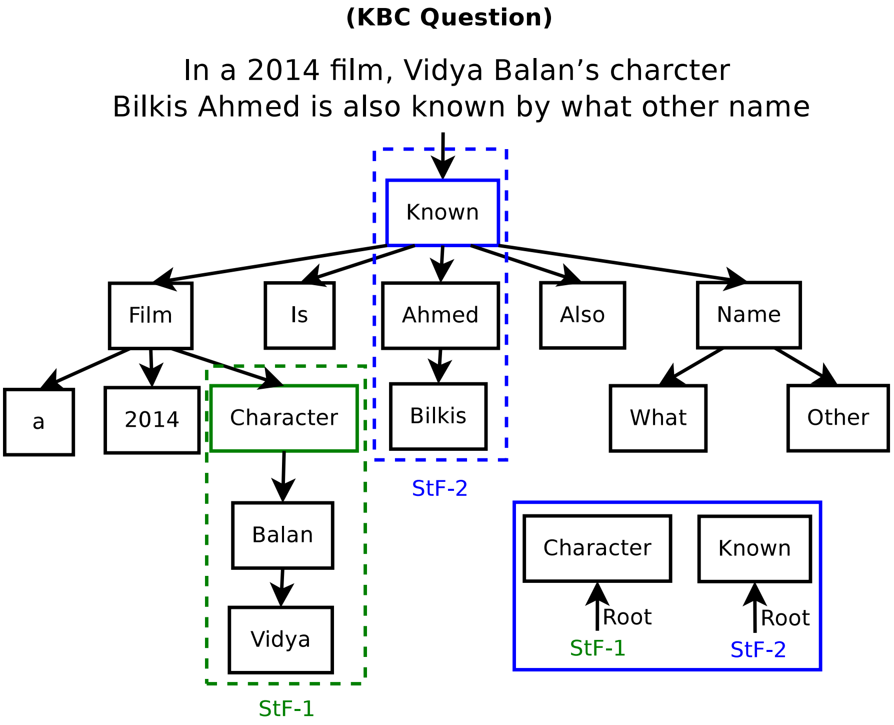
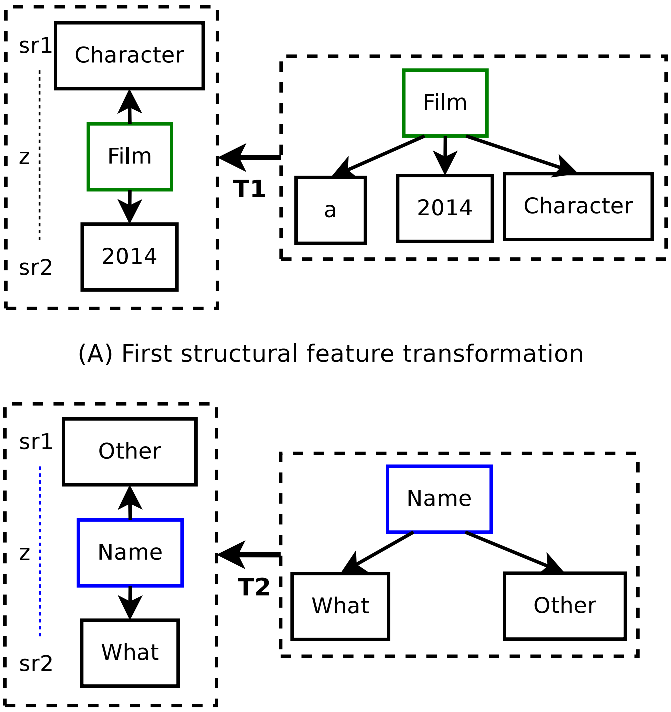


Figure 3.7: Structural features in a KBC question

In the Figure 3.7 structural features of a KBC questions are highlighted. The parsing technique uses the relation *'Vidhya Balan, a film character, has worked as Ahmed Bilkis in 2014'* separated by commas on the NP. The Parser uses a diverse dependency presented in questions and relevant document. Another example is the PP where many relations mean an alternative attachment with structures. By targeting semantic dependencies TD, provides an adequate representation for a question.

The structural features are transformed into a binary relation by removing the non-contributing words (i.e. stopwords). Figure 3.8 show such a design for two structured features T_1 and T_2 of a question which is shown in Figure 3.8. There can be more than two structural features in the question so there can be more than two structural transformations.



(A) First structural feature transformation

(B) Second structural feature transformation

FIGURE 3.8: Transforming structural features into binary relations

KBC dataset has manually annotated data for information retrieval in the open domain version to be tagged with the TD scheme. These conversion rules that are used to transform TD tree into a binary structural relation.

3.2.2.2 Design Principals for structural features

The structural feature representation bears a strong representation of feature vector space, and, more directly, it describes the grammatical relations. These design principals are used as a starting point for extracting the structural features. For obtaining SF, the TD helps in structural sentence representation, especially in relation extraction. SF makes available two options: in one, relations between it and other nodes, whereas in the second, making changes and adding prepositions into relations.

The intended use of structural extraction SF attempt to adhere to these six design principles (DP_1 to DP_6):

DP_1: Every dependency is expressed as a binary relation obtained after the structural transformation.

DP$_2$: Dependencies should be meaningful and valuable to EG and NLQA.

DP$_3$: The structural relations should use concepts of traditional grammar to link the most frequently related word.

DP$_4$: The relations with a maximum number of branches should be available to deal with resolving the complexities of indirect relations helpful for word alignment.

DP$_5$: There should be the maximum possibility of relations to be in NP words and should not be indirectly mentioned via non-contributing words.

DP$_6$: Initially the is the longest meaningful connection on which minimum non-contributing words than linguistically expressed relations.

From dependency rules and design principals for structural features the feature extraction algorithm aims to extract all possible structural features of the question. This structural feature extraction algorithm considering these design principles is discussed in the next section.

3.2.2.3 Structural Feature Extraction Algorithm and Score

The proposed structural features which are obtained from the dependency structure of a question on the basis of DPs. Consider a rule $X \rightarrow Y_n$ for $n = 1...Q_l$ (Q_l question length) in which X and Y_i are non-terminals in a dependency tree. Consider X to be the root for this relation $(X \rightarrow Y)$ Y_i will participate in structural features.

The relations of dependency tree are used for extracting SFs to capture long-distance connections in the question and text. For the sentence: *'Which sportsperson was made the brand ambassador of the newly formed state of Telangana'*, dependency relations are as follows. dobj(made-4, Which-1), nsubjpass(made-4, sportsperson-2), auxpass(made-4, was-3), root (ROOT-0, made-4), det(ambassdor-7, the-5), compound(ambassdor-7, brand-6), dobj(made-4, ambassdor-7), case(stat e-11, of-8), advmod(formed-10, newly-9), amod(state-11, formed-10), nmod:of(amb assdor-7, state-11), case(Telangana-13, of-12), nmod:of(state-11, Telangana-13). The structural features are designed using the dependency principals of the proposed algorithm that is shown in Algorithm 3.4. The root word and its siblings are expanded to measure the design principles.

The TD includes many relations which are considered as structural features: For instance, in the sentence *'Indian diplomat Devyani Khobragade posted where, when she* *was arrested in a visa case in 2013'*, The following relations under the TD representation are obtained:

TD_1 : *amod*(khobragade-4, Indian-1)
TD_2 : *det*(case-15, a-13)
TD_3 : *compound*(case-15, visa-14)
TD_4 : *nmod*(arrested-11, case-15)

The algorithm extracts four structural relations numeric modifier relation between *'arrested'* and *'case'*. Algorithm also provides an apposition relation between *'posted'* and *'arrested'*. The relation between these words represent the best possible link available in the text and it captures the significant features. For example, the adjectival modifier gleeful in the sentence, relation of verb to have textual dependecy is shown:

S_tF_1 : *dep*(posted-5, where-6)
S_tF_2 : *advmod*(arrested-11, when-8)
S_tF_3 : *advcl*(posted-5, arrested-11)
S_tF_4 : *nmod*(arrested-11, case-15)

The connection between these outcomes shows that SF proposes a wider set of dependencies, catching relation distance which can contribute to evidence gathering and question alignment. The parallel structural representations help in linking two words which can not be linked otherwise, and this is the reason for choosing NP words as root.

The TD scheme offers the option prepositiona dependencies involvement. In the example *'Name the first deaf-blind person who receive a bachelor of arts degree?'* instead of having two relations *case*(degree-12, of-10) and *dobj*(receive-7, bachelor-9) or *nmod*(bachelor-9, degree-12) and *acl:relcl*(person-5, receive-7), SF gives a relation between the pharses: *case*(degree-12, person-5). These links are used later in this work in EG & NLQA. Some more useful structural extractions such as, e.g. *'Which sport uses these terms reverse swing and reverse sweep?'*. TD gives direct links between *'swing'* and *'swap'* for (*dobj*),

TD_5 :*dobj* such as (reverse-6, swing-7)
TD_6 :*dobj* such as (reverse-9, sweep-10)
SF_5 : (reverse, sweep, *dobj*)

The information in SF_5 is not apparent in the TD which follows *dobj* in a similar way, have relations with three parameters such as, $(SF_i,\ SF_j,\ TD)$.

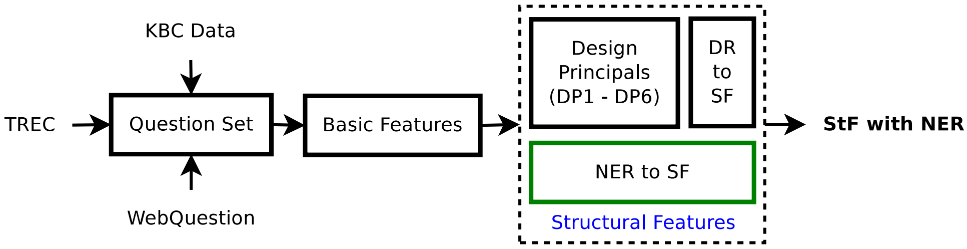


FIGURE 3.9: Adding named entities to structural features

SF representation is enhanced with the addition of named entities, for the sentence in Figure 3.9. In Figure 3.9 structural features are extracted from design principles, dependency rules and named entities which give an outcome as structural features with NER. The information available for the word *Telangana* in the SF scheme:

SF_{ne} : (Telangana5, location)

The structural information becomes valuable with the use of named entities and, SF provides the root to relate the words from the named entities. The structural feature extraction algorithm using extraction rules is shown in Algorithm 4.

3.2.3 Proposed Feature Relevance Technique

Individual features are tested on the different dataset to get the final answer (features are used in QA). The feature which is contributing in attaining the highest accuracy by QA system is marked as the most relevant feature. The accuracy of answer correctness after including these individual features is shown in Table 3.2.

The feature relevance is calculated by the Equation 3.2, where $\sum QC_i$ is the sum of correctly answered questions (i ∈ KBC, WebQuestions and TREC) and $\sum Q_T$ is the total number of questions.

$$F_r = \frac{1}{2} \times \frac{\sum QC_i}{\sum Q_T} \tag{3.2}$$

The relevance score of the features (F_r) is useful where feature vector is redundant and we need to reduce the space. In such situations the features with low relevance score

Algorithm 4 Structural feature extraction algorithm & weight of structural features

INPUT: Question set (Q)
OUTPUT: Sy_{fv} : Structural feature vector from Q
Variables used
(Q_f, V) : $(QuestionFeature, FeatureValue)$

```
for questions in dataset $Q$ do
    if $isTerminal$ = '$leaf$' then
        backtrack tree
    else
        expand_root procedure
    end if
end for
procedure expand_root
if $root$ has child nodes then
    for childs in tree $T$ do
        if $isTerminal \neq$ '$NP$' then
            backtrack tree
        else
            head_child $\leftarrow$ apply_rules from $DP$ (Rule 1 to 6)
            $Weight_{DP} \leftarrow Weight_{DP} + 1$
            head_child $\leftarrow$ apply_rules from $DR$
            $Weight_{DR} \leftarrow Weight_{DR} + 1$
            head_child $\leftarrow$ apply_NER
            $Weight_{NER} \leftarrow Weight_{NER} + 1$
        end if
    end for
end if
$St_{fv} = Weight_{DP} + Weight_{DR} + Weight_{NER}$
Return $\rightarrow$ $StructuralFeatureVector$ $(St_{fv})$
```

can be removed from the feature vector. The feature selection techniques (Agarwal et al. 2014) are also used for selecting the features with relevant information.

3.3 Dataset and Result Analysis

3.3.1 Dataset Used

To measure the correctness of the proposed features and their extraction algorithms, the publicly available KBC dataset is used. This dataset consists of open-domain questions

Table 3.2: Basic and proposed features with their relevance in QA

	Correct Answers (%)			
Basic Features	WebQ	TREC	KBC	**Relevance (Fr: 1-5)**
Unigrams	61	63	67	**4**
Bigrams	82	79	88	**5**
Trigrams	58	55	52	**3**
Wh-word	48	35	32	**3**
Word Shape	51	43	48	**3**
Question Length	28	23	19	**2**
Tagged Unigram	43	42	46	**3**
POS tags	46	51	56	**3**
Headword	87	88	91	**5**
Headword Tag	62	58	52	**4**
Focus Word	76	72	80	**4**
HW Hypernyms	66	54	63	**4**
Named Entity	83	82	77	**5**
Headword NE	57	52	49	**3**
Proposed features (structural features S_t)				
S_t with DP	56	61	65	**4**
S_t with DR	67	68	72	**4**
S_t with NER	92	88	91	**4**

with option and answers. For accurate and more stable experiments, TREC and WebQuestions datasets are also used that consist the relevant documents. Sample questions and documents from each dataset are given in Appendix A.

3.3.2 Performance Metrices

The performance of the feature extraction algorithms on KBC dataset and other datasets is measured by the total number of questions accurately answered by each features and by the combination of features.

Correct Answers (CA): It belongs to the number of correct answers provided by a particular feature.

Incorrect Answers (IA): It belongs to the number of incorrect answers provided by a particular feature.

Correct Documents (CD): It belongs to the number of correct documents selected by a particular feature.

Incorrect Documents (ID): It belongs to the number of incorrect documents selected by a particular feature.

The feature accuracy is employed for estimating the performance of basic and proposed features in QA. The precision of features is the division of the total questions that are correctly expressed by the features and the total number of documents that are to be expressed (it is the summation of TP and FP) as given in Equation 3.4.

$$Precision_{features} = \frac{CA}{CA + CD} \tag{3.3}$$

The recall is the division of the total number of correctly expressed question or documents to the total number of question or documents that are to be expressed (it is the sum of TP and FN) as given in Equation 3.5.

$$Recall_{features} = \frac{CA}{CA + ID} \tag{3.4}$$

F-measure is the aggregate of $Precision_{features}$ and $Recall_{features}$ is given by Equation 3.5.

$$AnswerExtraction_{accuracy} = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3.5}$$

In this analysis, accuracy of answer extraction (A_cAE) (also termed as F-measure) is done to report the performance of feature representation for QA systems and later NLQA and EG algorithms.

3.3.3 Results and Discussions

Ten-fold cross validation method is employed to estimate the accuracy of the proposed methods. The question data and documents are split randomly into 90% training and 10% testing. Wh-words (who, where, when) give an idea of expected answer type of the question, that is why it is important to handle such wh-words in QA. In the experiments, a simple approach is adopted such as to find the Expected Answer Type in the document. For example, *'which is the largest city of the world'*, the *'city'* is the EAT for this question and the document is searched for all the named entities with NE tag *'Location'*. The weighting formula for individual features has been discussed in Algorithm 1, 2 and 3 are used to calculate the weight of the basic features and proposed features.

3.3.3.1 Determination of Prominent Features

In the experiments it can be observed that bigrams (B_i) features are better than any other features on datasets as shown in Table 3.3. The bigram feature set provides the accuracy 69% as compared to 62%, 58%, and 52% for unigrams, trigrams and word-shape feature respectively on KBC dataset as shown in Table 3.3. The probable causes for this can be explained as follows. Unigram feature set contains lots of irrelevant features, which depreciates the accuracy of answer extraction. Also, trigrams feature set scattered than bigrams which demote the accuracy. Word-shape features are not valuable for the question and important mainly for document analysis. Word-shape feature set contains less information that is not enough for answer extraction in QA. Hence these perform worst when used separately. The dependency features are present in both question and document, resulting in more particular features. That is why these features are used to design structural features and contribute much for QA. These standard features are more useful for QA.

Table 3.3 present the accuracy of answer extraction (A_cAE) for all the basic features and Table 3.4 present the accuracy of answer extraction for all the proposed features. The accuracy of bigram features is considered as the baseline for this experiments.

The proposed feature sets features with the addition of DP, DR, and NER increases the accuracy for datasets (WebQuestions, TREC, and KBC). For example, St_{dr} features increased the accuracy from 64% to 78% (+14%) with addition of St_{ner} on KBC dataset. The proposed St_{dp} features also increased the accuracy from 68% to 82% (+14%) with addition of St_{ner} on KBC dataset. It is because adding NER to structural features improve the accuracy by dropping unnecessary and unrelated features. Structural features with design principals St_{dp} in addition to NER attain the accuracy of 82% as compared to its comparable combined basic feature set ($U_n + B_i + H_w + T_u$) i.e. 89.6%. It is 7.4% more than the proposed structural features but still meaningful. Therefore, structural features are very relevant and selective features. It is clear that in basic features, bigram features performed well than others, while used independently. Whereas, proposed

Table 3.3: Accuracy of Answer Extraction (A_cAE) of basic features on KBC dataset

Basic Features on KBC Dataset					
			A_cAE (%)		A_cAE (%)
Lexical	1	Unigram U_n	62	$U_n + B_i$	84
	2	Bigram B_i	69	$B_i + U_n$	84
	3	Trigram T_i	58	$T_r + U_n$	66
	4	Wh-word W_w	38	$W_w + B_i$	72
	5	Word Shape W_s	52	$W_s + B_i$	67
	6	Q^n Length Q_l	18	$Q_l + B_i$	64
	(a)	Le_{avg}	49.5	$U_n + B_i$	84
	(b)	$Le_{max} - Le_{avg}$	+19.5	$Le^*_{max} - Le^*_{avg}$	+34.5
Syntactic	1	Tagged U_n T_u	45	$T_u + H_w$	78
	2	POS Tag P_t	52	$P_t + H_t$	58
	3	Headword H_w	62	$H_w + T_u$	78
	4	H_w Tag H_t	53	$H_t + P_t$	70
	5	Focus Word F_w	61	$F_w + T_u$	72
	(c)	Sy_{avg}	54.6	$U_n + B_i$	71.2
	(d)	$Sy_{max} - Sy_{avg}$	+7.4	$Sy^*_{avg} - Sy^*_{max}$	+6.8
	(e)	$Le_{avg} + Sy_{avg}$	52.1	$U_n + B_i + H_w + T_u$	83.6
	(f)	$Le_{max} - Sy_{avg}$	+14.4	$Le^*_{max} - Sy^*_{avg}$	+12.8
Semantic	1	H_w Hypernym H_h	44	$H_h + N_e$	78
	2	Named Entity N_e	65	$N_e + H_h$	78
	3	H_w NE H_n	62	$H_n + N_e$	67
	(g)	Se_{avg}	57	$H_h + N_e$	74.3
	(h)	$Se_{max} - Sy_{avg}$	+8	$Se^*_{avg} - Se^*_{max}$	+3.7
	(i)	$Se_{max} - Le_{avg}$	+15.5	$Se^*_{avg} - Le^*_{max}$	-27
	(j)	$L_e + S_y + S_e$	53.7	$U_n + B_i + H_w + T_u$	89.6
	(k)	$Le_{max} - Se_{avg}$	+12	$Le^*_{avg} - Se^*_{max}$	+32
	(l)	$Sy_{max} - Se_{avg}$	+5	$Sy^*_{avg} - Se^*_{max}$	+35

Table 3.4: Accuracy of Answer Extraction (A_cAE) of proposed features on KBC dataset

Proposed Features on KBC Dataset					
			A_cAE (%)		A_cAE (%)
Structural	1	S_tF with DR St_{dr}	64	$St_{dr} + St_{ner}$	78
	2	S_tF with DP St_{dp}	68	$St_{dp} + St_{ner}$	82
	3	S_tF with NER St_{ner}	58	$St_{dr} + St_{dr}$	78
	(m)	St_{avg}	63.3	$St_{dp} + St_{ner}$	82
	(n)	$Le_{max} - St_{avg}$	5.7	$Le^*_{avg} - St^*_{max}$	+32.5
	(o)	$Sy_{max} - St_{avg}$	1.3	$Sy^*_{avg} - St^*_{max}$	+27.4
	(p)	$Se_{max} - St_{avg}$	6.3	$Se^*_{avg} - St^*_{max}$	+25

features are concerned dependency based structural features are more prominent than basic features (including bigram features), and the accuracy over datasets is presented

Table 3.5: Basic features on each question datsets and comparisions after adding structural features

Without structural feature (S_tf)						
			Question Dataset (No. of question taken)			
			WebQ (W) (5500)	TREC (T) (2000)	KBC (K) (500)	(W + T + K) (300 + 300 + 300)
Lexical	1	U_n	63.2	60.1	67.3	65.2
	2	B_i	68.1	55.3	62.6	66.1
	3	T_i	59.2	56.3	62.4	58.3
	4	W_w	38.3	53.6	58.4	61.2
	5	W_s	52.1	56.8	64.7	67.1
	6	Q_l	16.1	19.4	21.2	23.2
		Le_{avg}	**49.5**	**50.25**	**56.03**	**56.85**
Syntactic	1	T_u	43.2	55.8	57.5	53.6
	2	P_t	50.8	52.1	61.9	64.3
	3	H_w	60.8	86.5	71.3	70.6
	4	H_t	57.2	58.9	61.4	66.3
	5	F_w	53.8	55.9	59.6	55.4
		Sy_{avg}	**53.16**	**61.84**	**62.34**	**62.02**
Semantic	1	H_h	44.5	55.3	75.8	74.7
	2	N_e	65.1	55.8	73.9	71.6
	3	H_n	60.5	52.4	68.3	72.9
		Se_{avg}	**56.7**	**54.5**	**72.67**	**73.06**
With structural feature (S_tf)						
Lexical	1	U_n	68.2 (+5)	62.4 (+2.3)	72.7 (+5.4)	71.7 (+6.5)
	2	B_i	69.8 (+1.7)	58.9 (+3.6)	71.1 (+8.5)	73.3 (+7.2)
	3	T_i	60.1 (+0.9)	57.7 (+1.4)	63.2 (+0.8)	68.9 (+10.6)
	4	W_w	41.4 (+3.1)	55.5 (+1.9)	61.2 (+2.8)	72.6 (+11.4)
	5	W_s	55.3 (+3.2)	62.4 (+6.1)	57.7 (-7.0)	68.3 (+1.2)
	6	Q_l	15.3 (-0.8)	19.5 (+0.1)	22.1 (+0.9)	23.1 (-0.1)
		Le_{avg}	**51.68 (+2.18)**	**52.73 (+2.48)**	**58.0 (+1.97)**	**62.98 (+6.13)**
Syntactic	1	T_u	61.3 (+18.1)	55.9 (+0.1)	69.7 (+12.2)	75.9 (+18.4)
	2	P_t	55.3 (+4.5)	56.7 (+4.6)	71.4 (+9.5)	74.3 (+10.0)
	3	H_w	65.5 (+4.7)	86.9 (+0.4)	72.2 (+0.9)	66.9 (-3.7)
	4	H_t	58.1 (+0.9)	62.7 (+3.8)	72.3 (+10.9)	73.9 (+7.6)
	5	F_w	61.3 (+7.5)	56.4 (+0.5)	67.5 (+7.9)	71.3 (+15.9)
Semantic		Sy_{avg}	**60.3 (+7.14)**	**63.72 (+1.88)**	**70.62 (+8.28)**	**72.46 (+10.44)**
	1	H_h	79.1 (+34.6)	69.4 (+14.1)	76.2 (+0.4)	78.6 (+3.9)
	2	N_e	81.3 (+16.2)	78.3 (+22.5)	87.6 (+13.7)	82.4 (+10.8)
	3	H_n	66.4 (+5.9)	71.2 (+18.8)	84.3 (+16.0)	79.9 (+7.0)
		Se_{avg}	**75.6 (+18.9)**	**72.9 (+18.4)**	**82.7 (+10.03)**	**80.3 (+7.24)**

in Table 3.5.

The proposed structural features give better results than the basic bigram features with very fewer feature sizes. For example, question dataset after addition of structural features produced an accuracy of up to 82.7% (+10.03%) with 15 basic features for KBC adding the structural feature (S_tf) as shown in Table 3.4. Similarly, all the proposed features are constructed using dependency rules and performed better than similar basic features. For example, Se_{avg} attained the accuracy of 73.06%, whereas addition of structural features in these features produced an accuracy 80.3% (+7.24) on a combination of WebQ, TREC and, KBC dataset as shown in Table 3.4. Structural feature set presents the accuracy of 82.7% (+10.03%) on KBC dataset. It is because by including the dependency rules, design principals, and NER relevant semantic information dependency features contain a large number of long distance relation capture. The proposed features produce an accuracy of 75.6% with a maximum increase in the accuracy of +18.9 as shown in Table 3.5.

The proposed features resolve the issue of hidden features while decreasing the feature space by combining the features with basic features and NER features. Structural features include the noun-phrase dependency distance as per design principals. Structural features are very useful to their ease of extraction, and these reduce the feature vector size significantly. It is followed in the experiments that if structural feature vector is very small then the performance is not well, and as feature vector size is increased the performance increases. It is due to the reason that as vector size is increased, the possibility of a grouping of the root words in the structure performs better. Experimental outcomes state that the proposed structural features with the addition to basic features perform better.

3.4 Summary

The performance of several basic features of relevant documents is examined on three datasets namely WebQuestions, TREC (8 and 9), and KBC. Apart from the basic features new structural features are proposed viz. structural features. The feature extraction algorithms for basic and proposed structural features is proposed.

Proposed structural feature are combined with DP, with DR, and with NER. Further, the features have been assigned a relevance value which is calculated from the accuracy of an individual feature by their answer extraction accuracy on QA systems. It is also examined that addition of proposed structural features to the basic features improve the performance of answer extraction on QA systems.

Furthermore, it is noticed that proposed structural features provide improved results for bigrams (B_i) features and prominent proposed structural with NER (St_{ner}) features provide excellent results than basic features. The accuracy of the question length features was near to the 20% which is the minimum among all features. The main cause for this is the Q_l only gives the idea of the question complexity. It is observed that when two basic features are combined their combination gives better results than the individual feature. The combination of bigrams with question length do not perform well for QA systems.

In chapter 4, we will use all these features to gather evidence and a indirect reference based approach for evidence gathering is proposed and combined with feature-based evidence gathering.

Chapter 4

Evidence Gathering and Scoring Approaches

The motivation behind QA research is the need of user who is using state-of-the-art search engines. The user expects an exact answer rather than a list of documents that probably contain the answer. In this work, we consider a particular issue of QA that is gathering and scoring answer evidence collected from relevant documents. The evidence is a text snippet in the large corpus which supports the answer. For Evidence Scoring (ES) several efficient features and relations are required to extract for machine learning algorithm. These features include various lexical, syntactic and semantic features. Also, new structural features are extracted from the dependency features of the question and supported document. To score the evidence, for an existing question-answer pair, Logical Form Answer Candidate Scorer technique is used.

A QA system aims to find a concise answer to a natural language question. QA systems retrieve only the exact information, unlike search engines that provide the list of relevant documents. For example, given the question *'Who was the first Indian criketer to hit the double century in one day international?'*, an ideal QA system would answer *'Sachin Tendulkar'*. Therefore, QA system saves time and provides the required information which is accessible on all devices. To improve performance, recently QA has been drawing attention by using knowledge-bases. True Knowledge is a web-based QA system, and IBM's Watson is state-of-the-art QA system which defeated the human champions in Jeopardy! Game show. Wolfram Alpha gives access to the world's facts and data and calculates answers across a range of topics. Start is a system of recent era designed to answer to the questions asked in natural language.

Gathering evidence which supports the answer to a particular question is an important part of QA. Among three stages of QA i) question analysis, ii) document retrieval and, iii) answer analysis as shown Figure 2.1 of chapter 2. After question analysis and document retrieval QA systems employs a broad set of evidence and scoring section to compute the collected pieces of evidence. A useful dataset for this purpose, it has been discussed later in the section 5.1. The dataset has question-option-answer pairs of a famous Indian game show Kaun Banega Crorepati (KBC) which is similar to very famous game show Jeopardy. Supporting target documents for this dataset are collected manually from the web.

The focus now goes on option scoring using question and document features. In this, question features are collected to produce an intermediate form of question called Question Feature Form (QFF) and document features are collected to produce an intermediate form of the document called Document Feature Form (DFF). These intermediate forms are mapped with options to to collect supporting evidence. Feature scoring sections use ranked text passages. These passages are associated with a candidate answer which is one among four provided options. Now treating all the passages as evidence an Intermediate Feature Form (IFF) of each passage is produced. IFF finds passages which are closely associated with that question for a given option. The final evidence score is calculated after merging the scores of all features. Figure 4.1 shows the overall architecture of evidence scoring system. It is showing the three parts of the system first is related to feature extraction and IFF score generation. Second is related to gathering answer evidence, and third part is the merging and ranking section.

The calculated score is further used to compare the answer evidence score which is calculated from the document. It is useful for evidence gathering for complex KBC questions having a question-option-answer dataset. The proposed approach is similar to get the score for a semantic parser. Semantic parsers map utterances to their formal meaning representations using a logical form. This approach uses question features to represent intermediate form to map ranked documents. For meaning representations, almost every parser uses a predefined set of constants. In this work, instead of using logical terms we focus on question features. These question features provide a significant

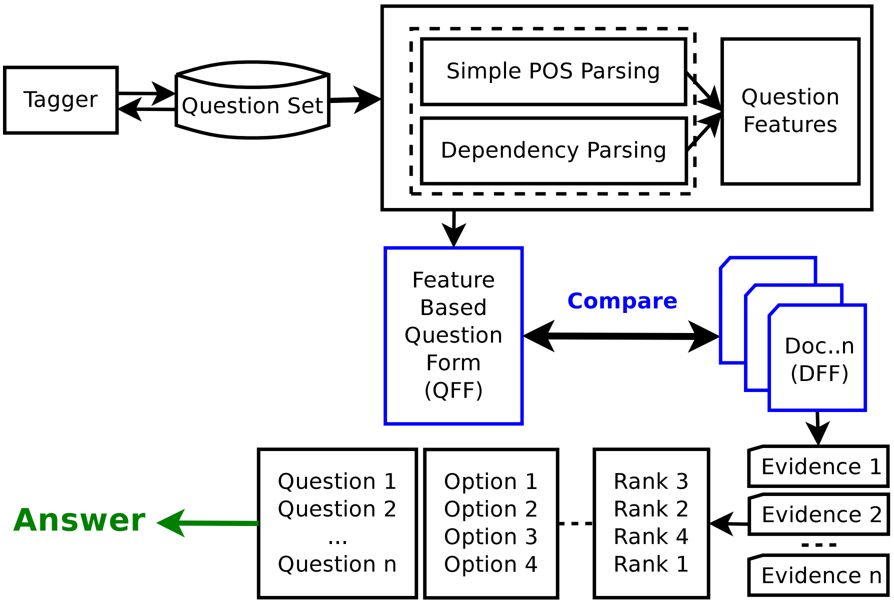


Figure 4.1: Overall architecture of proposed evidence scoring system

indication in gathering supported evidence from the document. Let F be a set of features in the question (Q), and R be a set of ranked documents in the collection of documents (D). Also, let Y be an intermediate feature expression that can be executed against R to return evidence for particular answer E = EXECUTE(Y, R). The purpose is to build a feature function for mapping a natural language question Q to an intermediate feature form Y. We assume access to data containing (Qi), (Oi1, Oi2, Oi3, Oi4) where, i = 1 . . . n question-option pairs and a ranked document set R. The learning algorithm estimates the parameters of a linear model for ranking the possible entries in E. Learning of word meaning is removed in the stages of parsing and validated by some features as part of the learning model.

Information Retrieval Vs Evidence Gathering- In an IR system, the user asks a query, and its results are shown as a list of documents whereas in evidence gathering system a question is answered with a confidence value. IR methods give the most relevant document results. Whereas, EG methods provide a complete scenario behind the retrieval (i.e. that provides a confidence value to the answer); the comparison is shown in Figure 4.2.

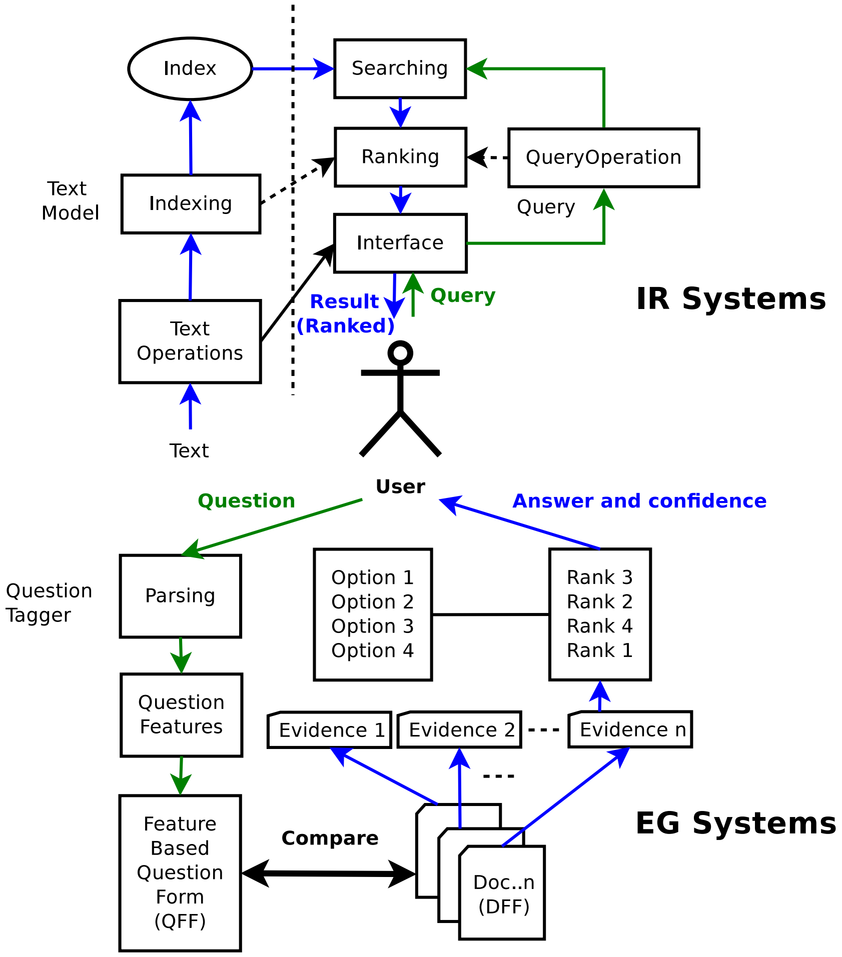


FIGURE 4.2: An architecture of information retrieval system and evidence gathering systems

4.1 Related Work

In the literature of evidence retrieval, Supporting Evidence Retrieval (Murdock et al. 2012) is a system which put the probable answer into the primary question to make a proposition; and then it uses the DeepQA search techniques to retrieve the passages which are most closely related to the proposition. The scores (including all evidence-scoring components) during this phase are calculated and combined using a statistical model which is later used in answer ranking. The Indri passage retrieval algorithm is also used for finding and generating candidates. It uses a predicate-argument structure (Marcus et al. 1994) for the syntactic portions of the graph using an English Slot Grammar (McCord 1990) parse. The Skip-Bigram algorithm (Murdock et al. 2012) for evidence gathering was first introduced in for machine translation, in which the system translations are matched with gold standards. Both, the question and the passage text are mapped with their graph representation using a similar mapping procedure.

Along with the challenges in the existing semantic parsing (Pradhan et al. 2004) methods to find the most appropriate question representation model. The proposed approach is focused on getting a Logical Feature Form (LFF) score from question features used for evidence gathering in ranked documents. There are several efforts done in making a question representation model using Combinatory Categorical Grammar (Steedman et al. 2011) and dependency trees (Baluja et al. 1997). In some approaches, the direct supervision is hidden by latent variables or distant supervision (Mazzetti et al. 2000; Wener 2006). These latent meaning representations are used for training the semantic parser from utterance and denotations. QA research can be broadly categorized as machine learning-based (Wang et al. 2002) approaches and knowledge-based (Yang et al. 2003) approaches. The performance of the machine-learning approaches greatly depends on the effectiveness of the feature extraction process.

In proposed work for EG, the denotation of question and document is replaced by a score of LFF (which includes structural features of both) that is used to retrieve document evidence. Authors use two steps to produce the final logical form, first to map utterance to domain-independent logical form and then ontology matching to get the final logical form. We employed with Dependency Compositional Semantics (DCS) features (mainly with structural features) and produced an intermediate QFF. In another work authors (Yao et al. 2014; Pinto et al. 2002) attempted to map text to structured form and some authors attempted the natural language predicate and argument triples to map with structured RDF triples. Author (Yao et al. 2014) created a semantic parser for structured knowledge-base (KB) relations. Precision and recall are used as performance matrices for semantic parsers to calculate the accuracy of the LF. The system learns from QFF and document evidence. We compare our evidence scoring system with those systems which learn from question-answer pairs. The system employs learning with: morphological features, syntactic parse trees, a set of semantic features and structural features. Feature selection methods are used to obtain optimal features for IFF generation.

For the given question-option-answer dataset of KBC, this work is divided into following subtasks: A) question parsing, B) feature extraction and C) intermediate QFF. Later scoring the evidence and learning these evidence with QFF. Now consider the result of question parsing and feature extraction subtasks. In question parsing phase the dependency features of question and passage are collected.Many features are extracted, and relevant are selected in feature extraction phase. Now the details of intermediate QFF and its score is discussed in details. This will further help in calculating a evidene score for answer evidence.

4.2 Proposed Evidence Gathering Approaches

A feature extraction vector $F \leftarrow f(Q_i, z)$ is defined where Q_i is the provided question, and z is the value obtained from various features. It is the essential part of intermediate QFF model. Features are divided into lexical, syntactic, semantic and appropriate structural features. Structural features are similar to the features used for DCS tree. In DCS, structural features have importance because of the tree representation of the question. In this case, features have importance according to their relevance which is calculated by the feature selection methods discussed in chapter 3.

Open domain QA systems use wide-ranging coverage of parsers. The quickness and correctness of CCG parser are used to parse answer candidates. This is mixing the parser into a QA system for evidence gathering, scoring and learning. Parsers are applied to the questions, for two reasons: i) the use of question features allows the parser to deal with extraction cases, which is the important part in question parsing for intermediate form generation and, ii) comparison of possible answers from the ranked documents with options. Answer extraction component is simplified if the same parser is used for both question and target documents. Parsing is done on the questions of KBC dataset. The proposed approaches using features and references is shown in Figure 4.3. Two proposed approaches generate two proposed models indirect-reference base Evidence Gathering (iEG) model and feature-based Evidence Gathering (fEG) model

In the initial stage of parsing, the results were not as per expectations because the structure of questions was not frequent. For example, there are no “what”questions with the common form of “What is the name of the first president of India?”. In KBC questions, this is a very common form of Wh-question. In KBC question set there is a lesser number of similar question types beginning with “How”or “Who”. As creating

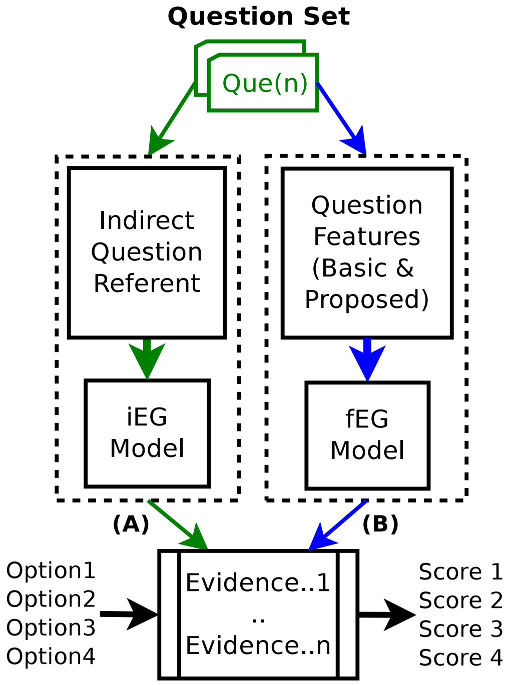


FIGURE 4.3: Proposed evidence gathering approaches: (A) Reference-based approach and (B) Feature-based approach

a new data set is always expensive, so an existing alternative CCG Lexical Category (CLC) annotation is used with KBC dataset. CLC annotation is easier than annotating a question with its derivations. It can be done with the tools and available resources.

In this work, the question is annotated with a super-tagger which uses the output form of dependency parsers. This tagger is sufficient to give high parsing accuracy on complex KBC questions. For example, the dependency parsing of the question "*In a 2014 film, Vidya Balan's character Bilkis Ahmed is also known as what other name?*"is shown chapter 3 in Figure 3.7. In next section the evidence gathered using the basic and structural features are discussed in details.

4.2.1 Feature Based Evidence Gathering Approach

Evidence gathering is done using the features of question and document. For this various alignment features including basic and proposed features are extracted. The basic features include lexical, syntactic and semantic features but not all of the basic features are relevant for alignment purpose. Relevant basic features are selected after their individual feature performance and tested over proposed structural alignment features.

Selection of relevant features is meaningful as the high dimensionality of features is a curse for the performance of machine learning algorithms. Figure 4.4 is giving an overall architecture of proposed feature-based alignment approach.

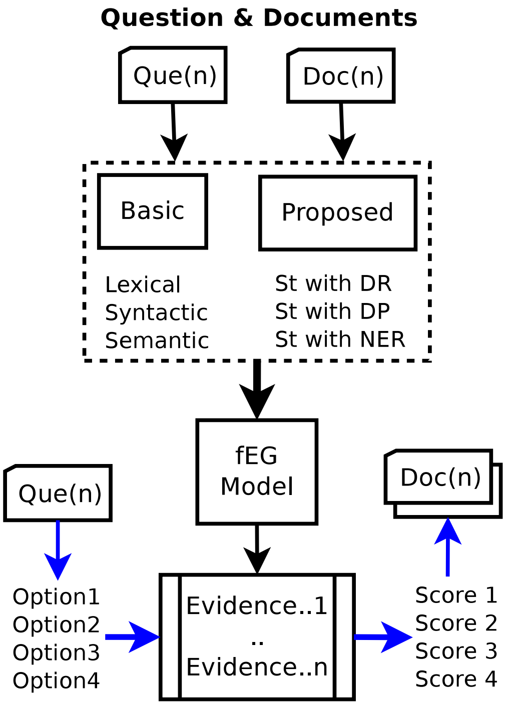


FIGURE 4.4: Feature-based EG approach to get fEG model

4.2.1.1 Basic Evidence Features

Lexical- CCG's lexical or morphological features are the POS category of the word. That is associated with forward and backward CCG operation rule. The important lexical features (say, L_e) according to their relevance are selected. Lexical features tolerate the feature space to reason about the denotation of unobserved words. Unlike the small size domain, large scale and open-domain questions are impossible to demote with limited training data.

As, we have already discused that, n-gram is defined as a sequence of n items in the question. An n-gram for n=1 is referred to a unigram. Similarly, for n=2 and n=3 n-grams referred as bigrams and trigrams. A wh-word can appear anywhere in the question as in our example question it appears at the position 14. In our case a wh-count and position both are used what, 1, 14, this feature identifies whether the wh-word is at starting or somewhere else in the question. Word shape and question length also used

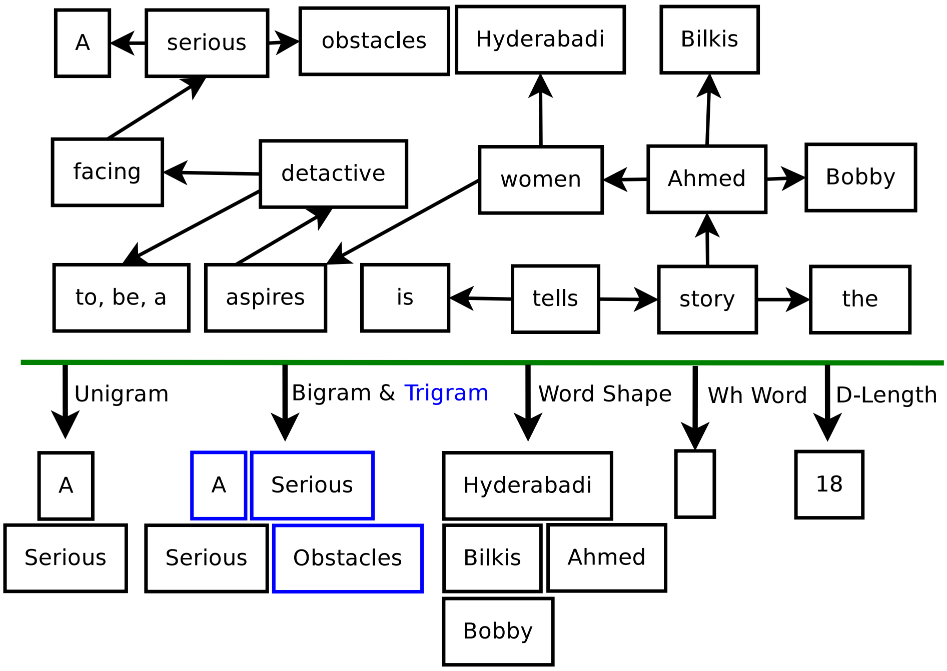


FIGURE 4.5: Lexical evidence features

in lexical features. Each lexical feature is explained in Table 1 along with relevance of these features on other feature of similar category. The feature relevance is measured by feature selection methods. The syntactic features extracted like n-gram, and word shape are shown in Figure 4.5.

Syntactic- The Part of Speech (POS) tags and headwords are the most commonly used Syntactical Features (say, S_y). A headword is usually defined as the most descriptive word for a question or a word that defines the purpose of question. The syntactic features extracted like tagged unigram, headword tag and focus word are shown in Figure 4.6.

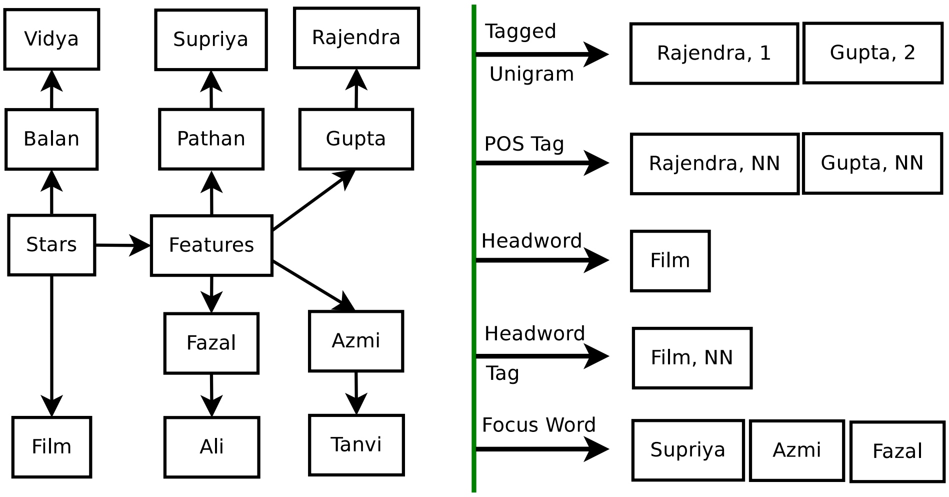


FIGURE 4.6: Syntactic evidence features

Semantic Features- For Semantic Features (say, S_e), we require a third party database such as WordNet, or a dictionary to extract semantic relations of question. The most commonly using semantic features are headword's hypernyms, related words, and named entities. WordNet is a lexical database of English words. It gives a lexical hierarchy that links a word with higher level semantics particularly hypernyms. For example, a hypernym of the word *'city'* is *'municipality'* of which the hypernym is *'urban area'* and so on. There is no major effects of hypernym unavailability. Named entities are another very important semantic feature used in some studies. Named entities are semantic categories which can be assigned to a word in a given sentence. The syntactic features extracted like headword, and named entities are shown in Figure 4.7.

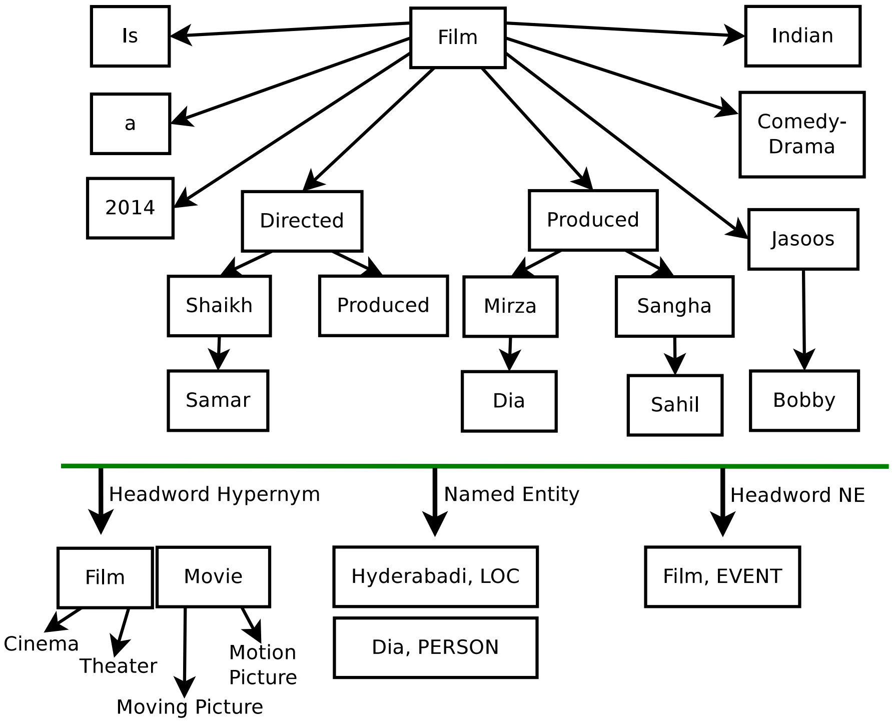
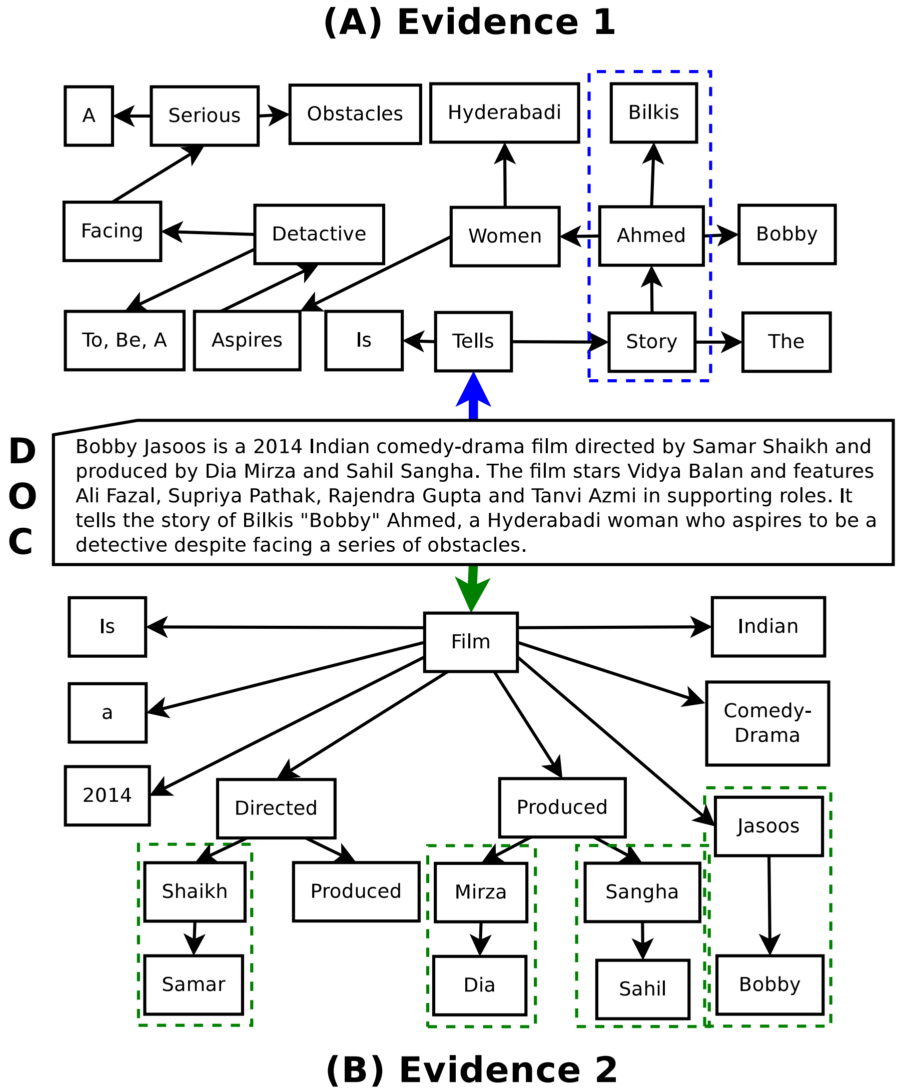


FIGURE 4.7: Semantic evidence features

4.2.1.2 Structural Evidence Features

The structure matching operators in ranked documents (R) is defined, which produce novel feature vector that is represented by the features as the outcome of dependency parse. These structural features (say, S_t) are used for complex relations presented in R and used for the uniqueness of efficient constants available in parsing results. The QFF produced for a question contains one composite feature function. Structural features allow the model to adapt for all ranked documents having evidence support for the question structure. Each feature captures properties about (Q, R) which precise the details of the exact occurrence to a generalize occurrences that share common features among evidence. Figure 4.8 is showing documents structural feature evidence. There are relations where produced has a relation with directed which can not be identified directly, evidence 1 of Figure 4.8 is showing this relation. The relation contributes as a structural evidence. There are two pieces of evidence in Figure 4.8 and one more evidence of same document is shown in Figure 4.9.

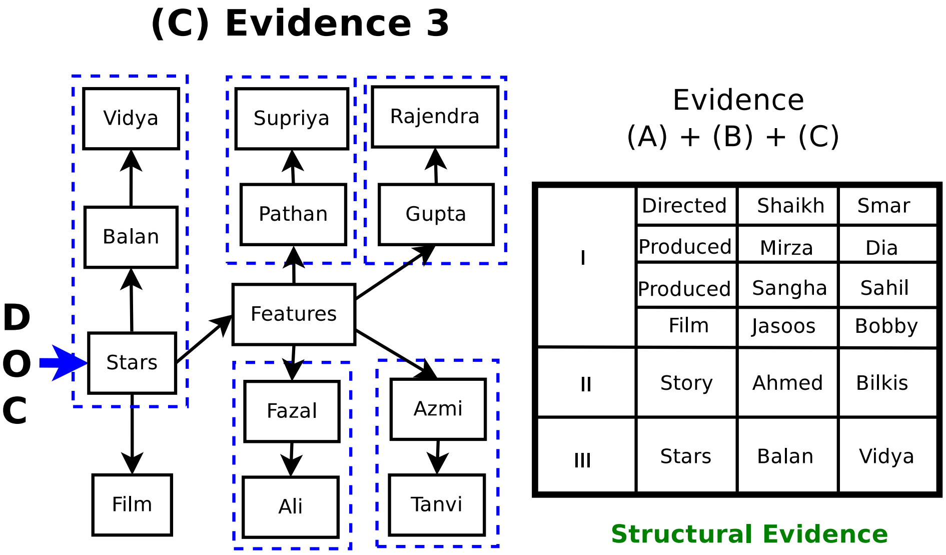

(A) Evidence 1

Bobby Jasoos is a 2014 Indian comedy-drama film directed by Samar Shaikh and produced by Dia Mirza and Sahil Sangha. The film stars Vidya Balan and features Ali Fazal, Supriya Pathak, Rajendra Gupta and Tanvi Azmi in supporting roles. It tells the story of Bilkis "Bobby" Ahmed, a Hyderabadi woman who aspires to be a detective despite facing a series of obstacles.

(B) Evidence 2

FIGURE 4.8: Structural evidence features of two evidence in a document

In the Figure 4.9, a complete structural evidence table is shown which is combining all the available pieces of evidence. A relation of two words, produced and directed



I	Directed	Shaikh	Smar
	Produced	Mirza	Dia
	Produced	Sangha	Sahil
	Film	Jasoos	Bobby
II	Story	Ahmed	Bilkis
III	Stars	Balan	Vidya

FIGURE 4.9: Structural evidence features of third evidence in a document

(*produced* ↔ *directed*) and stars and film (*stars* ↔ *film*) can be extracted from the structural evidence.

4.3 Indirect Reference Based Evidence Gathering Approach

In evidence gathering, reference resolution is an important part while collecting supporting evidence in unstructured text. Indirect reference is defined as the referent-referee relation between two somehow related entities (e.g. car and garage). These referents of indirect referring expressions are unstated and also not mentioned previously in the text. Distance between these co-refers can be calculated which can be used at the time of semantic tagging. Which identifies the most probable hidden word associated with the entity. In the proposed algorithm, semantic distance of two referents is calculated using the depth first search tree traverse algorithm. The calculated distance between two referents is used for suggesting the accurate hidden word to resolve indirect reference in answer source (sharma et al. 2015(a)). Algorithm is discussed for resolving indirect reference and the outcome of the algorithm can be used for resolving all kinds of reference (e.g. one reference, pronominal reference etc.). Figure 4.10 is showing the complete architecture of proposed reference based EG approach. In this approach, two parts are considered one is related to existing anaphora resolution and other is proposed indirect reference based; these two complete the ovelall reference based EG approach.

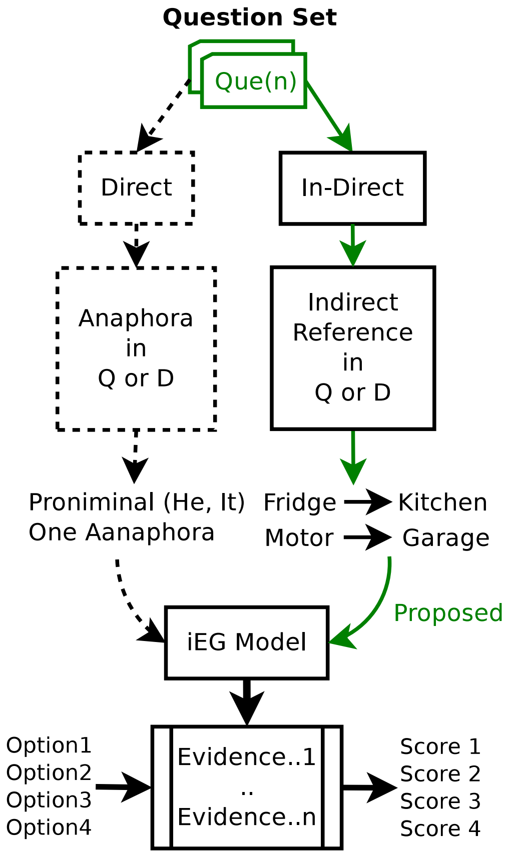


Figure 4.10: Proposed reference-based evidence gathering approach

4.3.1 Indirect Question Referents

Resolving indirect reference is challenging because one try to encapsulate the unstated or background knowledge in the text. The researchers tried to capture such knowledge by maintaining a tree structure starting from root and expands according to the decision of *'has'* and *'is part of'*. Maintaining such a semantic tree having a proper relation is essential in many problems; but these algorithm limit to capture the hidden information of only one entity and later it becomes difficult to relate this information with appropriate referring expression and this referring expression has importance in answer extraction. In this work, the proposed algorithm for identifying the difference in two co-refers (which are related indirectly) by calculating distance between these co-refer to root tree node (referring expression). For this an m-way tree is used and this tree expands at the same time it finds any entity attached with its node. So, the focus is on finding distance of a matched referent with its root node. After finding this distance it is compared with the another tree. Therefore, the best suitable referent can be chosen for provided referring expression and available hidden knowledge. This proposed algorithm store distance in form of positive integers ranges from 0 to +. Hence, this algorithm is fast in comparing the outcomes coming from different tree traversals.

4.3.2 Reference Based Evidence Algorithm

This algorithm captures the indirect references in the text and use these references later as the evidence. These indirect references in the document are promising evidence to supprot the answer. The steps used in this algorithm are explained below, $\mathbf{S}_i \in \mathbf{i}^{th}$ **Step**

S$_1$: A referring expression say fridge door and construct a tree of all possible referents. Assigning the value to E_a according to predefined rules. Starting from E_a= 0. Shown in Figure 4.11.

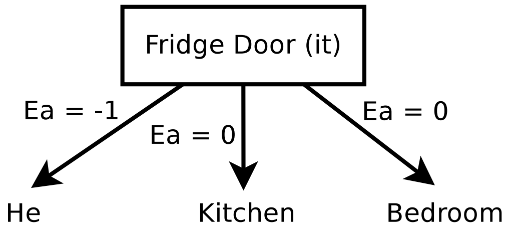


Figure 4.11: Expansion of fridge door

S$_2$: Now traverse tree nodes and assign value + to those leaf nodes having negative values in previous step.

Negative value of any node in previous step indicates no use of the node for this calculation. And assigning positive to indicates that this node is more expandable and can be used for different calculations. For value of $E_a \geq 0$ expend the node further Shown in Figure 4.12.

S$_{3a}$: After tree nodes and assign value + to those leaf nodes having negative values in previous step. the negative value of any node in previous step indicates no use of the node for this calculation.

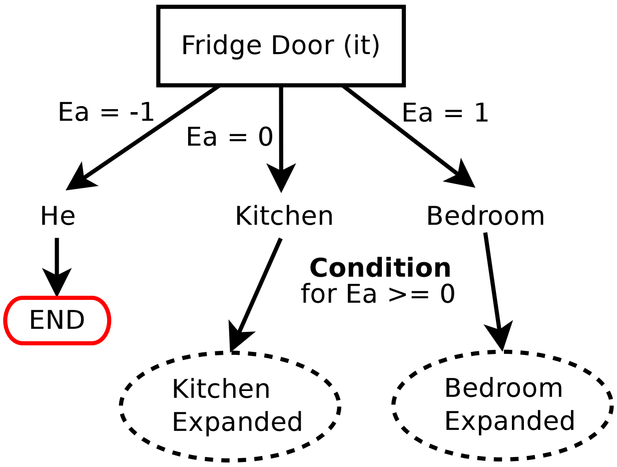


Figure 4.12: Complete node traversal and weight assignment

S$_{3b}$: Take a referring expression say fridge door and construct a tree of all possible referents. Assigning the positive to indicates that this node is more expandable and can be used for different calculations (Figure 4.13). Now traverse tree nodes and assign value + to those leaf nodes having negative values in previous step.

The negative value of any node in previous step indicates no use of the node for this calculation.

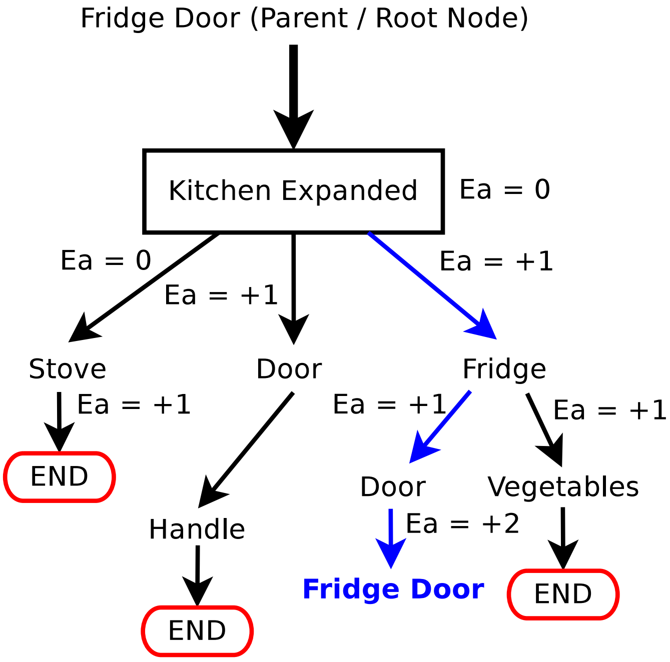


Figure 4.13: Kitchen is expanded and assigned value on tree nodes

And assigning positive to indicates that this node is more expandable and can be used for different calculations. Assign the value to E_a according to predefined rules. Starting from E_a= 0. Shown in Figure 4.14.

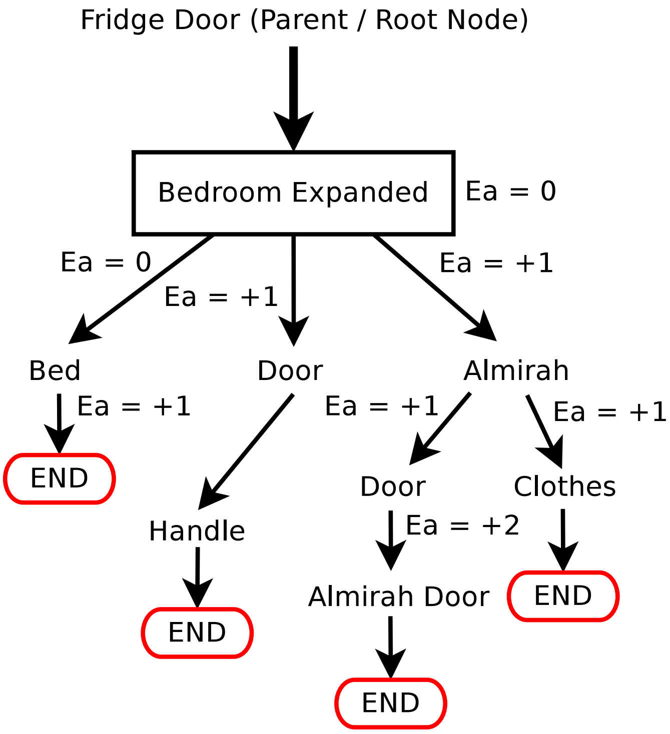


FIGURE 4.14: Bedroom is expanded and assigned value on tree nodes

S$_4$: (Compare) Now comparing different values of step 3(a), Step 3(b), Step 3 (c)... nodes and assign value + to those leaf nodes having negative values in previous step. negative value of any node in previous step indicates no use of the node for this calculation.

S$_4$: (Classification) Classify according to the value E_a. Category 1: As E_a= should be places in table that will be least used for further discussions. Category 2: Table of most promising indirect referents with value of E_a=+1 and E_a= +2.

S$_4$: (Decision) Now on basic of the value of E_a. nodes and assign value + to those leaf nodes having negative values in previous step. negative value of any node in previous step indicates no use of the node for the calculation shown in Table 4.1.

Table 4.1: Sentance tags, word tags and referent distance calculated using the algorithm steps

Sentance Tag	Word Tag	Referent Distance
S_{t1}	W_{t1} : *he*	$E_a = -\infty$
S_{t1}	W_{t2} : *kitchen*	$E_a = +2$
S_{t1}	W_{t3} : *bedroom*	$E_a = +\infty$
S_{t2}	W_{t1} : $fridge$	$E_a = +1$
S_{t3}	W_{t1} : *door*	$E_a = +1$

How this algorithm works: The higher values of E_a is the indication of more promising referent and it is shown in Figure 4.15 that larger the value of E_a more will be the chance to stay in the bucket of prominent referents. Behavior of the E_a is shown in Figure 4.15, which shows that an entity having E_a= - and '0' have no chances to be the prominent referent of the question.

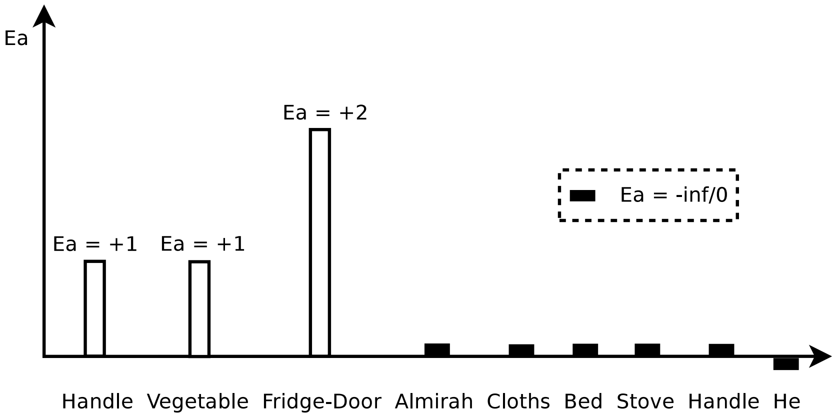


Figure 4.15: Proposed Evidence Gathering Approaches: A) Reference based and B) Feature based

Algorithm is calculating semantic distance for multiple referents in a single discourse for indirect reference used previously in the any of the $Queue_{noun}$ or $Queue_{verb}$. It also extracts the ongoing discourse and semantic relation with the help of discussed parallel queue method. Extracted value of ongoing discourse and semantic relation can be used to identify indirect reference in same sentence. We aimed to present a general model which is similar to the estimation models which estimate one among many possible solutions. The model was trained properly which was portrayed by the overwhelming training results. It was found that systems using indirect reference are better evidence collector than the previous methods. This work gives an idea of hybrid models (using feature and reference) to resolve evidence gathering.

4.4 Evidence Scoring

In this section, scoring algorithm merges all evidence scores to provide a single evidence score. This score is the combination of L_e, S_y, S_e and S_t feature scores. The score of QFF and DFF are calculated from Eq. 4.1 is treated as the individual feature evidence score and used to score the provided options.

4.4.1 Feature Form based Evidence Score

A logical form is generally used to query a knowledge base. The intermediate QFF generated in this work does not query any knowledge base and it is used for representing the question to a QFF score. This QFF score is than used to map with DFF score. Structural features are of more importance in this work and rest features contribute equally. Let, L_e be the lexical feature, S_y be the syntactic feature, S_e be the semantic feature and S_t be the structural feature. The QFF and DFF scores can be represented in Eqation 4.1 (the logical aggregation is generated after regressive analysis of featues). The addition of all the evidence score in Table 4.2 is 22, this is useful in comparing the document evidence.

$$QFF_{score} = DFF_{score} = \sum_{i=1}^{n} [log(L_e \times S_y \times S_e)] \times S_t \tag{4.1}$$

4.4.2 Indirect Reference Score

An indirect form is used to get a the score of reference based evidence. IR_{score} calculated in this work is the individual evidence weight calculated in Table 4.3. but it is used for representing the question to its QFF weight and then to map it with DFF weight. Structural features are of more importance in this work and rest features contribute equally. Let Le be the lexical feature, S_y be the syntactic feature, S_e be the semantic feature and S_t be the structural feature. The QFF and DFF scores can be represented

Table 4.2: Basic features of all question datsets and comparision after adding structural features (for KBC dataset)

Feature-based scoring of a relevant document used to generate DFF						
		Features	Question	Evidence$_1$ (E$_1$)	Evidence$_2$ (E$_2$)	Evidence$_3$ (E$_3$)
Lexical	1	U_n	0.632	0.601	0.673	0.652
	2	B_i	0.684	0.553	0.626	0.663
	3	T_i	0.592	0.563	0.624	0.583
	4	W_w	0.383	0.536	0.584	0.612
	5	W_s	0.523	0.563	0.647	0.671
	6	Q_l	0.161	0.194	0.212	0.232
	Lexical score (Le_s)			**4**	**2**	**3**
Syntactic	1	T_u	0.432	0.558	0.575	0.675
	2	P_t	0.508	0.521	0.619	0.643
	3	H_w	0.608	0.865	0.713	0.706
	4	H_t	0.572	0.589	0.614	0.663
	5	F_w	0.538	0.559	0.596	0.554
	Syntactic score (Sy_s)			**3**	**1**	**1**
Semantic	1	H_h	0.445	0.553	0.758	0.747
	2	N_e	0.651	0.558	0.739	0.716
	3	H_n	0.605	0.524	0.683	0.729
	Semantic score (Se_s)			**0**	**1**	**0**
Structural	1	St_{DR}	0.545	0.553	0.558	0.547
	2	St_{DP}	0.551	0.558	0.539	0.616
	3	St_{NER}	0.605	0.524	0.5683	0.529
	Structural score (St_s)			**2**	**3**	**2**
	$Le_s + Sy_s + Se_s + St_s$			**9**	**7**	**6**
Combined Evidence Score (E$_1$ + E$_2$ + E$_3$)						**22**

Table 4.3: Reference based scoring on evidence E$_1$, E$_2$, E$_3$ (for KBC dataset)

Reference-based scoring of a relevant document (evidence availability)					
		Reference	Evidence$_1$ (E$_1$)	Evidence$_2$ (E$_2$)	Evidence$_3$ (E$_3$)
Promominal	1	Sentance Recency	1	1	1
	2	Subject Emphasis	1	0	1
	3	Existential Emphasis	0	0	1
	4	Accusative Emphasis	0	0	0
	5	Head Noun Emphasis	1	1	1
	6	Non-adverbial Emphasis	1	1	1
	Pronominal score (Pr_s)		**4**	**3**	**5**
Indirect	1	Level$_1$	1	1	1
	2	Level$_2$	0	1	1
	3	Level$_3$	0	0	1
	Indirect score (In_s)		**1**	**2**	**3**
	$Pr_s + In_s$		**5**	**5**	**8**
Combined Evidence Score (E$_1$ + E$_2$ + E$_3$)					**18**

in Eq. 4.2.

$$IR_{score} = \frac{\sum_{i=1}^{n} CWti \times RD_i}{\sum S} \tag{4.2}$$

Combined Evidence Scoring Table and its Significance- The score calculated in Table 4.2 and Table 4.3 is using the combination of evidence score. This combined score is useful to represent the complete document. Therefor, there are two possible evidence scores one, used to score a particular evidence and the combined score is used to score the complete document.

4.5 Data Set, Experimental Setup and Result

4.5.1 Dataset Used

To decide the prominent question features and to produce logical forms, we used publically available TREC Question Classification (TQC) dataset and KBC question dataset. In TQC, questions are tagged with their category which is useful in deciding the answer type. The answer type can be used in gathering answer evidence. To perform our experiment more reliably a more steady dataset has been developed that is KBC dataset. KBC dataset has question-option pairs and relevant documents. The questions in KBC dataset vary in various domains including movie, sports, geography and so on. There are about 1000 questions (500 for question alignment) are collected for KBC dataset. All these questions having four options, one answer and, at least, three relevant documents.

4.5.2 Experimental Setup

Before setting up for experiments, the difference between the intermediate feature score used in this proposed and traditions logical form should be clear. Let the utterance: “What is the highest point in Florida?”From the Geo dataset has the Logicalform:(A,highest(A,(place(A),loc(A,B),const(B, stated(florida))). Now utterance: “In a 2014 film, Vidya Balan’s character Bilkis Ahmed is also known by what other name?”From the KBC dataset, there is Logical Form Score (LFS) not an LF. LFS, in this case, will be a number, not a representation: say, LFS = 1.183.

4.5.3 Results and Discussions

The feature vector (F) formed after question processing is further used for calculating an individual question and document LFS. The LFS is used in evidence learning algorithm. In the experiments, LFACS technique and LFF score (which includes new structural evidence features) is used for scoring the relevant documents. Ten-fold cross-validation does the evaluation of feature extraction. For feature extraction, linear SVM and Naive Bayes Multinomial have been used. In the proposed work, each question is parsed to produce output in a form of dependency features. These dependency features are the backbone of structural features. Wh-words (what, which, where, who, when) are essential to handle at first stage of parsing to give a concrete idea of upcoming document evidence. A headword is also important as it gives the idea about lexical answer type. At this point, the accuracy of headword extraction algorithm is critical.

4.5.3.1 Accuracy of Feature-based Evidence Gathering

In, Passage Term Match (PTM) question terms are matched to passage terms. The grammatical connection or word sequence is not considered. The Skip-Bigram (SB) technique gives the score by matching pairs of words that are related or nearly related. In Textual Alignment (TA) a score is given by comparing the words and word order of the passage. In TA, the question focus is replaced by the candidate answer. In LFACS technique, the score is assigned by how likely the structure of the question can be mapped with the passage. Table 4.4 shows the results of passage scorer, from the support evidence retrieval system that has all four scorers. SB and LFACS (regarding high P and low R) have the excellent outcome. PTM or TA from the system does not show a significant impact on this dataset. LFS is useful for matching the appropriate question with the document having sufficient evidence to support its option. The focus is not only scoring the documents by supported evidence but further to combine these evidence to score a complete document.

Combined evidence score provides a complete scoring technique to this evidence scoring system. This score is further added to the feature-based score with experimental settings which regulates the evidence score. More the training examples are available for this scoring algorithm more accuracy it can attain, so the question is, how many examples

Table 4.4: Basic features of all question datsets and comparision after adding structural features

KBC Dataset (for total 500 questions)				
	SER Scorers (Existing)	**Precision**	**Recall**	**F-Measure**
SER	Passage Term Match	73.7	91.8	81.7
	Skip Bi-gram	81.4	90.4	85.7
	Textual Alignment	75.3	84.9	79.8
	Logical Form Answer Candidate Scorer	86.2	57.5	69.0
	Combined Basic and Proposed Features			
Proposed	L_e	57.2	62.4	59.68
	S_y	63.7	79.8	70.84
	S_e	71.8	84.6	77.67
	S_t	68.6	82.2	74.78
	$L_e + S_y + S_e$	79.5	81.2	80.34
	$L_e + S_y + S_e + S_t$	88.2	90.5	89.33

are mandatory to get a high level of precision in scoring? So the answer is when all training examples are used. Evidence scoring substitutes between updating positive and negative evidence sets and update parameters for I iterations.

For example, use I = 5 (similar to the settings of scoring- SER) as the default value. The calculation of the available evidence is based on search size where every intermediate form has at most J structural evidence features. The default value is J = 20. The proposed work is compared with evidence scoring with SER. Skip bigram technique in the SER has the highest accuracy, and that is 3.63% less than the combined feature accuracy of proposed feature-based approach. A main reason for this accuracy is the inclusion of structural features. Structural features are more informative than other features used for evidence gathering and scoring. Further, parallel queue method has been proposed for resolving indirect reference as a new approach to save time for capturing very far distance relations in text.

4.6 Summary

Proposed work is focused on gathering evidence for question-option from ranked documents. In this approach, initially question is parsed, and lexical, syntactic, semantic and very useful structural features are extracted. These features are used to form an intermediate QFF. Unlike from other logical forms, the QFF is calculated for a single real value. This unique value represents the question. Similarly, document's intermediate form called DFF. DFF is calculated and mapped with question's QFF score. At the end of QFF and DFF, evidence gathering is completed. After evidence gathering using QFF and DFF, the evidence is scored with provided options.

The proposed work is compared with Support Evidence Retrieval (SER). SER uses Passage term-matching, Skip bigram, and Textual alignment. In this work, QFF and DFF model use lexical, syntactic, semantic and structural features. The combination of features gives better results as compared to SER. Although the variation in the result is comparable. The reason is, extracting structural features of a question and relevant document is easy and efficient.

Chapter 5

Natural Language Question Alignment and Learning

In Natural Language Alignment (NLA) a link is established between closely related words of two sentences. Two similar words like *'plane'* and *'airplane'* can be aligned but two aligned words like *'wife of'* and *'married'* are not similar. A user can ask questions in different manners and expect the same answer. Various alignment approaches (li et al. 2009; Hall et al. 1999) have been proposed so far, and most of these are attempted on a word to word alignment. To increase the performance of QA systems it is required to align a natural language question with another question that is able to extract the final answer. In this work, NLA is used for two similarly asked questions which are different in their structure. For Natural Language Question Alignment (NLQA), various features and relations are extracted. These features include useful lexical, syntactic, semantic and proposed structural features. Structural features are extracted from the dependency features of two or more similar questions. Further, these features are used to calculate a feature form score which is useful for learning algorithms. Furthermore, a Topic Word (TW) of the question is extracted to improve aligner accuracy. Experiments prove that for NLQA new structural features are better, and the accuracy of NLQA is increased when these are combined with TW and other features. For an existing question-answer pair, to represent the question in the intermediate form, feature form score of the question is generated. Later in the chapter, an algorithm is designed to learn the particular question feature form and answer pair.

In natural language, a same question can be asked in different ways but the answer for all of them should be same. For example, given the original question *'In medicine, which of these is usually denoted by 120/80 for an adult?'* and Similar question *'In medical science, 120/80 used for an adult?'*. An ideal QA system would answer *'Normal Blood Pressure'* for both questions. Therefore, in a QA system questions can be asked in many different ways which are having the same meaning directing these to the same answer. To enhance question alignment performance, QA systems are pulling attention by using Knowledge Bases (KBs). As shown in the figure 5.1, a user's query is converted into an intermediate form (e.g. MQL, SPARQL) which is further used to get the answer from knowledge bases. User's query can be different but their intermediate form should be same so that the query's answer remain same.

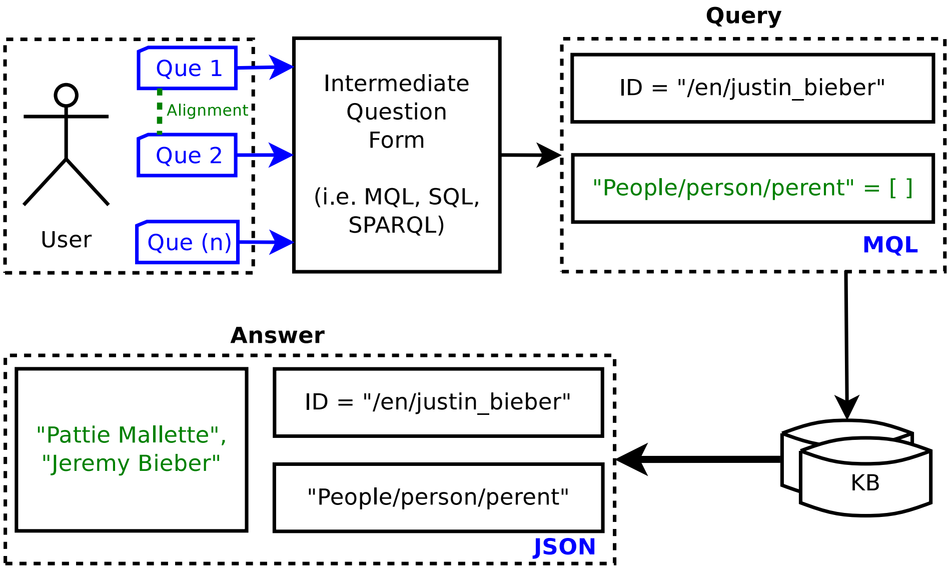


Figure 5.1: The initial stage where question alignment is required

Among the phases of QA schown in chapter 2, namely question analysis, passage retrieval and answer extraction, the proposed question alignment is done at the initial stage of question analysis which is shown in the Figure 5.1.

The intermediate form has a feature score to further map with the similar questions. First of all taking a single question and a similar question then an IFF of all questions is generated. Later in this chapter, the feature forms and topic based alignments are explained in details.

NLQA vs Text Similarity and Machine Translation- Textual similarity (Vasileios et al. 1999; Mohler et al. 2009; Gomaa et al. 2013; Daniel et al. 2013) is the task of deciding if two sentences express a similar or identical meaning and require an in-depth understanding of a sentence and its meaning to achieve high performance. It is used to make document clusters of similar representations as shown in Figure 5.2. Whereas, NLQA decide similarity from the alignment of combined features and Topic Word.

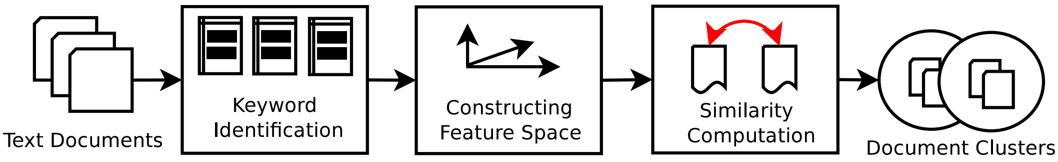


FIGURE 5.2: Text similarity forming the document clusters

The alignment problem is more familiar in Machine Translation (MT) (Koehn et al. 2007), where recognizing that *'Gandhi was killed'* can be inferred from *'MKG was assassinated'*, one must first recognize the correspondence between *'Gandhi and MKG'*, and between *'killed'* and *'assassinated'*. Figure 5.3 is showing that alignment is one of the way of computing similarity, and MT uses alignment to compare or to generate another text in different language.

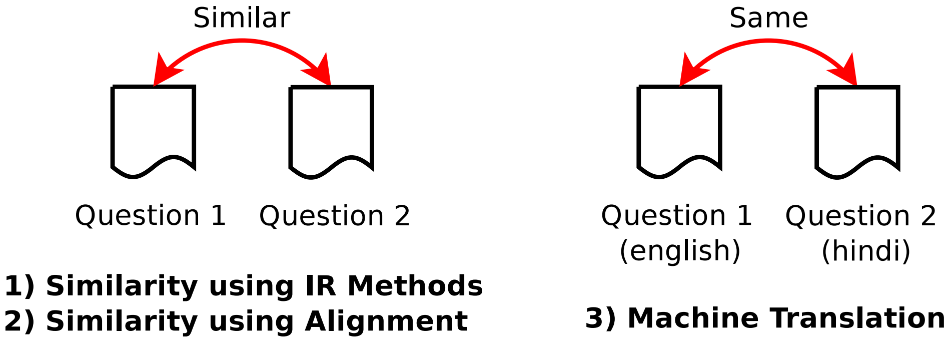


FIGURE 5.3: Difference in text similarity, alignment and machine translation

5.1 Related Work

Many authors (Toutanova et al. 2002; Roth et al. 2012; Yao et al. 2013) have attempted the text alignment among these the MANLI aligner (Yao et al. 2013) is proposed to align P and H sentences for the task of natural language inference. It applies perceptron learning (Ng et al. 1997) and handles phrase-based alignment (MacCartney et al. 2008) of arbitrary phrase lengths. There are issues in the aligner after the lexical alignment phase, and with additional syntactic constraints, the specific alignment match rate for sentence pairs is significantly improved. Besides the above-supervised methods, indirect supervision (Chang et al. 2010) has been examined. Among them, Wang and Manning (2010) continued the work of McCallum et al. (2005) and reduced alignment as latent variables. Heilman and Smith (2010) adopted tree kernels to explore the alignment that produces the lowest tree edit distance. Other tree matching work for alignment includes that of (Punyakanok et al. 2004; Chambers et al. 2007; Roth and Frank 2012).

The semantic problems are examined with the release by Microsoft Research (MSR) of human-generated alignment annotations (Brockett, 2007) for inferring from the Recognizing Textual Entailment (RTE) challenge. This work is the first to utilize the generated KBC data for training and evaluation of NLA models. The KBC data consists of a question set containing 500 aligned questions having at least two pairs for each.

Table 5.1: Various systems with their intermediate form accuracy

System	**Author**	**IF Accuracy**	**Year**
KZGS11	Kwiatkowski et al.	88.6	2011
KZGS10	Kwiatkowski et al.	88.2	2010
LNLZ08	Lu et al.	81.8	2008
λ-WASP	Wong and Mooney	86.6	2007
ZC07	Zettlemoyer and Collins	86.1	2007
WASP	Wong and Mooney	74.8	2006
ZC05	Zettlemoyer and Collins	79.3	2005
PRECISE	Popescu et al.	77.5	2003
COCKTAIL	Tang and Mooney	79.4	2001

The intermediate feature score is produced which is compared with the existing systems shown in table 5.1. This table is showing a comparison of systems with their logical form accuracy. These systems demonstrate that how a question can be converted into its logical form. Later it is shown that how we have calculated the intermediate logical form score to represent a question. In this work, alignment of two questions is done with generated feature form and TW.

5.2 Proposed NLQA Approaches

In this section, two proposed approaches for Natural Language Question Alignment (NLQA) are discussed. The first approach is feature-based alignment and the second approach is topic-based alignment (sharma et al. 2015(c)). For feature-based alignment, various basic features and proposed features are required from the both questions. These features include previously discussed lexical, syntactic, semantic and new structural features. All Structural features are not important for question alignment. Further, these features are used to calculate a feature form score which is useful to measure feature based similarity. In the experiments for NLQA, new structural alignment features are combined with basic features to improve the alignment accuracy. Figure 5.4 gives a brief overview of two proposed alignment approaches; first, feature-based NLQA and second, topic-based NLQA, which are discussed in details in the next sections.

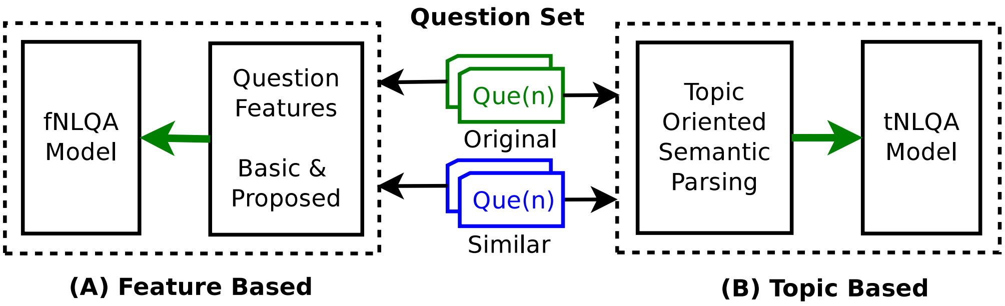


FIGURE 5.4: Overview of two proposed alignment approaches. feature-based alignment and topic-based alignment

5.2.1 Feature Based Alignment Approach

Natural Language Question Alignment is done using the features of two similarly asked questions. For various alignment features including basic and proposed features are extracted. The basic features include lexical, syntactic and semantic features but not all of the basic features are relevant for alignment purpose. Relevant basic features are selected after their individual feature performance and tested over proposed structural alignment features. Selection of relevant features is meaningful as the high dimensionality of features is a curse for the performance of machine learning algorithms. Figure 5.5 is giving an overall architecture of proposed feature-based alignment approach.

In the Figure 5.5, question set has a collection of one original and at least two similar questions. Both original (Q_o) and similar (Q_s) questions are passed through the same parsing stage. Parsing phase has two outcomes one from the part-of-speech parser and other from dependency parser. Dependency parsing gives dependency relations these relations are further used to design structural rules for proposed structural alignment

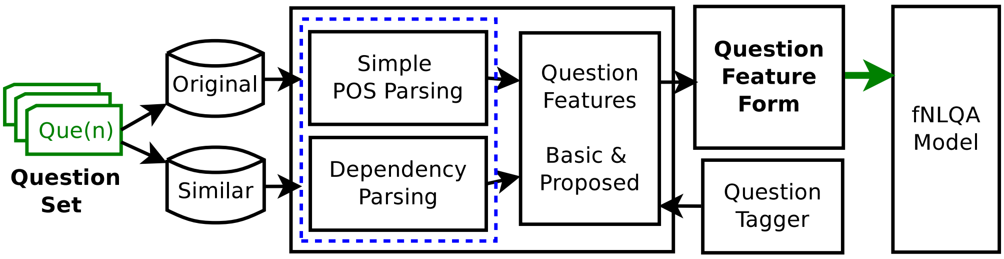


FIGURE 5.5: Feature based alignment approach resulting a fNLQA model

features. The questions in the dataset are tagged and kept in the question tagger to tag the rarely appeared text of the question. These features are kept together to form a question feature form which is used to create a feature-based Natural Language Question Alignment (fNLQA) model.

5.2.1.1 Basic Alignment Features

Lexical Alignment Features- Lexical or morphological features of categorical grammar can be used as the part of speech category of the word. That is attached to front and back by CCG rules. In this work, significant lexical features (say, L_e) are extracted from the question. These lexical features allow the feature space to think about the definition of unobserved words. Unlike the small size domain, large scale and open-domain questions are impossible to notate with limited training data. Figure 5.6 is showing the lexical alignment in the aligned and similar question. This lexical alignment is done by the aligner which matches the lexical words in the both questions.

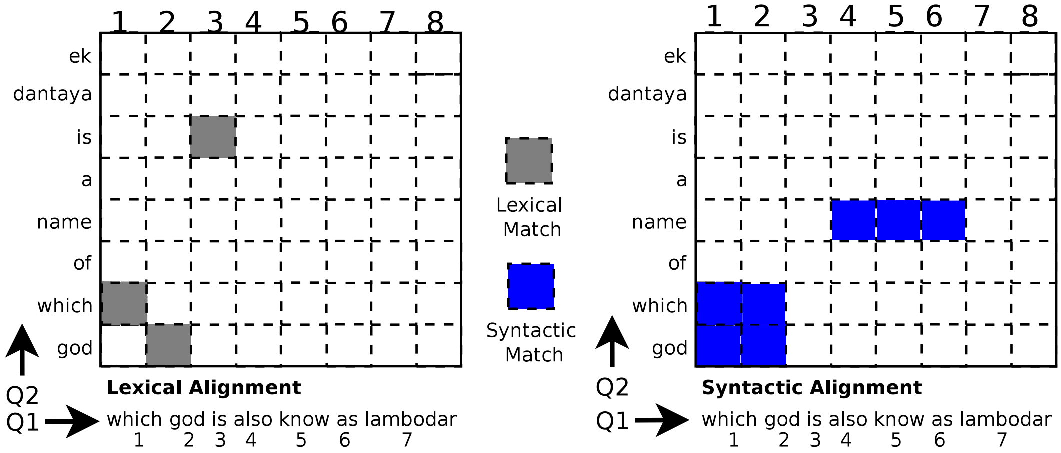


FIGURE 5.6: Lexical and syntactic similarity alignment between two questions.

An n-gram is simply designed as a sequence of n items in the question. An n-gram for n=1 is related to a unigram. Similarly, for n=2 and n=3 n-grams associated as bigrams and trigrams. A wh-word can appear anywhere in the question not just in starting.

Syntactic Alignment Features- The Part of Speech (POS) tags and headwords are the most commonly used Syntactical Features (say, S_y). A headword is usually defined as the most detailed word for a question or a word that specifies the purpose of the question. Some syntactic features are extracted like tagged unigram, headword tag and focus word are shown in Figure 5.6.

Semantic Alignment Features- For Semantic Features (say, S_e), a third party database such as WordNet [24], or vocabulary to extract semantic relations of question is required. The most commonly using semantic features are headword's hypernyms, related words, and named entities. Some important features like Named Entities are also used in some studies. These are shown in the Figure 5.7.

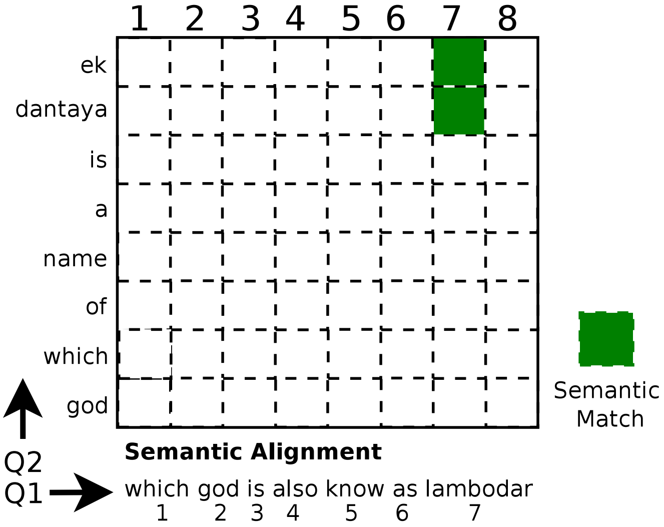


FIGURE 5.7: Semantic similarity alignment between two questions.

5.2.1.2 Structural Alignment Features

It has been discussed that the structural match of two questions is determined and it produces new feature vector that is collected from the features in the yield of dependency parse. These structural features (say, S_t) are employed for complicated relations presented in similar questions and used for the uniqueness of efficient constants available in parsing results. The QFF produced for a question contains one composite feature function. Structural features allow the model to adapt for all questions used for alignment using the question structure.

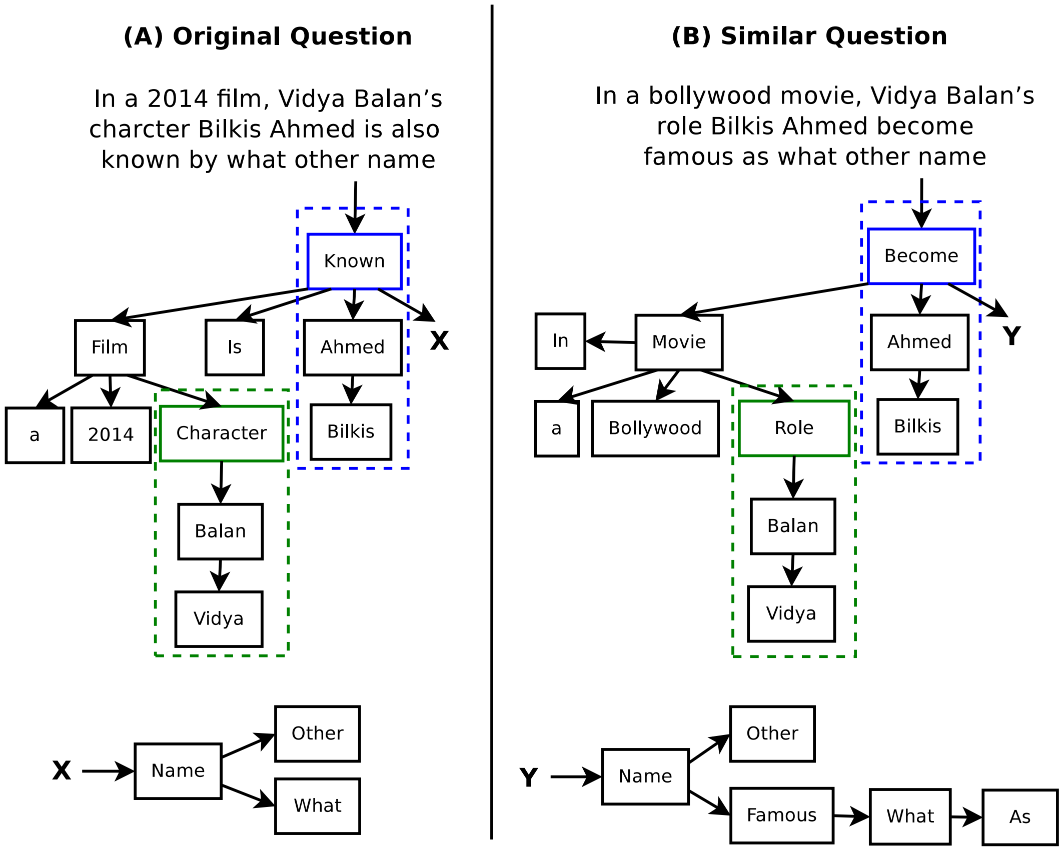


FIGURE 5.8: Sructural features in two simiar questions

Figure 5.8 shows the question's structural features for alignment. There are relations where *'Vidya Balan'* has an alignment link with *'Ahmed Bilkis'* (Vidya Balan → Ahmed Bilkis) which cannot be identified directly. The connection provides a structural confirmation of two root words *'character'* and *'known'*.

Figure 5.9 shows the structural features extracted from the question's dependency tree. These structures can be the longest sequence in the tree so these are transformed into binary relations. There are few rules for extracting structural features in the question those have been discussed in chapter 3.

In the Figure 5.10, it is shown that the transformed structural feature is represented in a two directional feature. In this way, the stop words are eliminated and two feature directions are represented by structural feature-1 (sr1) and structural feature-2 (sr2). The features are from an original and similar question. In the Figure 5.11, one original

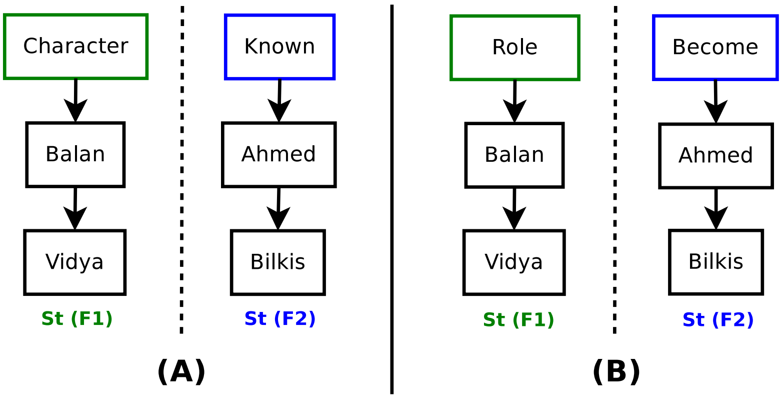


FIGURE 5.9: structural features transformation in original and similar question

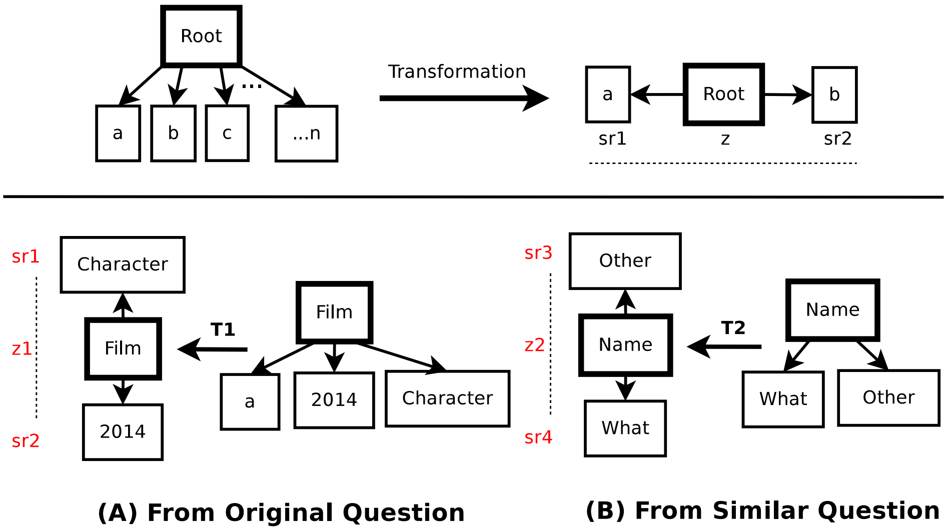


FIGURE 5.10: Structural transformation to get binary relations

question *'who is also known as lambodar'* is going to align with *'ek dantaya is a name of which god'*. *'who'* and *'ek dantaya'* are entities in original and similar question. Two semantic alignments *'also known as'* and *'name'*. In original question lambodar is tagged as *'name'* (*Entity* → *Name*).

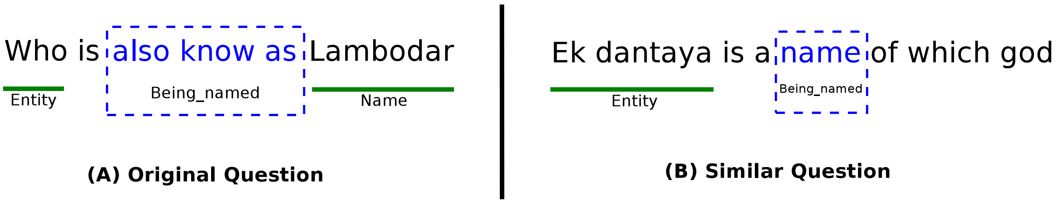


FIGURE 5.11: Structural cum semantic alignment between two question features

5.2.2 Question Topic Based Alignment

In this section, Natural Language Question Alignment is done by available Topic Word (TW) of the question and added to feature based model to increase the question alignment performance. For extracting the TW, semantic parsing of the question is required. Semantic parsing has become a challenging problem for open domain question answering. In semantic parsing, questions are mapped with their meaning representations. These representations are matched with possible answers in KBs (sharma et al. 2015(b)). In KBs (e.g. Freebase), knowledge is stored in the form of Topics. For a satisfactory answer extraction from Freebase, it is required to correctly identify the Topic Node (or Topic Word) of the question and retrieve every type and property associated with this Topic Node. Figure 5.12 is showing the combined feature-based and topic-based alignment model.

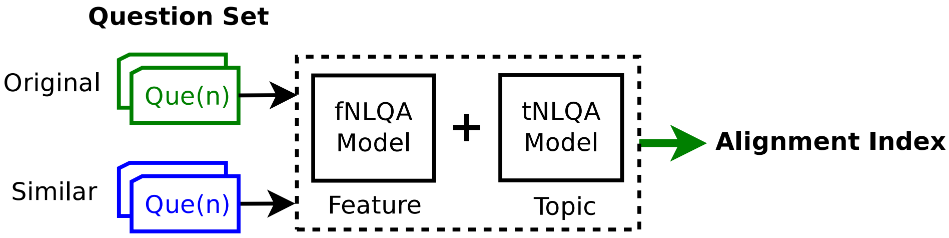


FIGURE 5.12: Combined feature-based and topic-based alignment model

In next sections, a Topic Word Identification (TDI) algorithm is proposed for correctly identifying question Topic and a Domain Word Identification (DWI) algorithm is proposed for correctly identifying the domain of the Topic Node. After domain identification, the Topic Node is further expanded for its all types and properties. Out of all types identified, one of the type and associated property is likely to be an answer to the question. TWI and DWI algorithms use techniques i.e. proposed rule-based and machine learning approach with the help of question dependency parser. Next, the Topic Word Identification and Domain Word Identification Algorithms are discussed iin details.

5.2.2.1 Topic Word Identification Algorithm

For accurately identifying the Topic Word a proper question analysis is required. An algorithm is proposed for correct Topic Word Identification, this Topic Word further plays an important role in making the Topic Oriented Parse Structure.

Algorithm 5 Topic Word Identification Algorithm (with freebase KB)

```
INPUT: Question set (Q)
OUTPUT: $T_i \in$ Topic words present in Q, i = 1, 2, 3 ... n)

Step 1: $NounExtraction$
$NN_w \leftarrow$ singular noun present in Q
$NNS_x \leftarrow$ plural noun present in Q
$NNP_y \leftarrow$ proper noun, singular present in Q
$NNPS_z \leftarrow$ proper noun, plural present in Q
Step 2: $NEExtraction$
$NER_i \leftarrow$ named entities (i = 0, 1, 2, 3 ... )
$NER_i \leftarrow NERt$ where, (i $\in$ tagged categories)
Step 3: $RuleBasedTopicWordn$
$T_r \leftarrow$ rulebase
$T_r \in$ focus word using proposed rulebase
Step 4: $TopicWordIdentification$
if i = 0 (Step 2 discarded) then
    $T_n \in NN_w, NNS_x, NNP_y, NNPS_z$
else
    i = 1 or 2, 3 ... (Step 2 is important)
    if if w or x or y or z = 0 then
        $Ti \leftarrow NER_i$
    else
        compare results of step 1, step 2 and step 3
    end if
end if
$T_im \leftarrow T_rNER_i, NN_w, NNS_xNNP_yNNPS_z$
where $T_{im} \leftarrow$ optimal topic word by comparing set of all possible focus words
    using machine learning methods
Return domain word $T_{im}$
```

The outcome of Topic Word Identification algorithm shown in Algorithm 5 is T_n which consist a set of relevant Topic Words for the question. In the first step, every noun (i.e. singular noun, proper noun, etc.) present in the question is collected in a noun container. In the second step a named entity tagger trained by Stanford NER is run on question and named entity is obtained in a function called NER_i. In the third step the proposed rule base is applied on the question and based on algorithm two Topic Words are identified T, Topic Word from a rule-based system and Topic Word using machine learning technique used on WebQuestions data set. TWI algorithm mainly based on our rule base which is formed with the help of several dependency parse of web-questions dataset. A glimpse of proposed rule-base is shown in Table 5.2.

Table 5.2: Proposed rules to extract Topic Word T_w

Rule No.	**Proposed Rule**	**Topic Word**
1	if Root is = ”IS” and root (ROOT, X)	$prep-n, nsub$
2	if nsub is = ”name”	$prep-of$
3	root (ROOT, X) prep-in (X, Z) and nn(Z, W)	WZ
4	nsub (VERB, X) amod (X, Y)	YX
5	root (ROOT, VERB (X)) and tmod (VERB(X), Y)	Y

Now, consider an example, from the Web-questions dataset, *‘what character did natalie portman play in star wars?’*. This question is of WebQuestions dataset and this question contains the possibility of identifying the two Topic Nodes (i.e. natalie portman or star wars). Now the weight function used in each step of algorithm uniquely assign a weight to the corresponding term in the question. In the initial processing phase question is represented in its parse structure this parse structure is useful to analyze various part of speech tags. The interest is in tagging Noun Phrases (NPs) for step 1 of the algorithm. Topic word weight is calculated by Equation 5.3 represents the average score and total weight.

W_T= Total combined weight (each step for each word)
QW_j = Weight of t^{th} word in question
$QWNP_j$ Weight of j^{th} word in question when it is a NP
NPT_j number of terms in j^{th} NP
NP = Noun Phrase in question and N, NN_w or NNS_x or NNP y or $NNPS_z$

$$QW_j = \frac{1}{(\sum NP) + N} \tag{5.1}$$

$$QWNP_j = \frac{1/(\sum NP) + N}{NPT_j} \tag{5.2}$$

$$TW_{weight} = \sum QW_j + \sum QWNP_j \tag{5.3}$$

Using the rule base to extract the TW (Algorithm 5, step 3). This total weight TW_{weight} counted when the most prominent Topic Word to be identified. A weight is used to calculate the each step weight, and the summation decides the correct Topic Word.

5.2.2.2 Domain Word Identification Algorithm

There are 77 domains (named entities) these are individually identified at the semantic parsing phase. DWI algorithm is explained here, as the output of this algorithm is to identify set of domains (D_n) which consist a set of relevant Domains for the question. In the first step, the output of the Algorithm 6 is stored in the D_x.

Algorithm 6 Domain Word Identification Algorithm

INPUT: Question set (Q)
OUTPUT: D_n : Domain words present in Q, i = 1, 2, 3 ... n)

```
Step 1: CallTWI
$D_x = X^{th}$ Domain
$D_x \leftarrow T_n$ (from algorithm 1 TWI)
Step 2: DomainCheckLoop
$W_i \leftarrow i^{th}$ word in question
$D_o$ = No domain
for $W_i \neq 0$ do
    if $W_i = D_j$ (where j = 1, 2, 3 ... 77) then
    else
        $W_i = D_o$
    end if
end for
Step 3: CompareTWI
$D_n = D_x, D_j$
Return domain word $D_n$
```

In the second step, a for loop is run for assigning whole 77 Domain words to a D_I. This domain word is collectively stored in with the previous step output D_x and then the output is identified as D_n. If there is no domain identified then x=o and D_x become D_o in this case the Topic Word category of the question will be the Domain Category. This may happen in the situation when Topic Word is not available in KB. After a proper named entity extraction (sharma et al. 2014), the proposed rule base is applied using the dependency parse of the question.

5.3 Learning Question Alignment

For the KBC question-option-answer dataset, the work is divided into following tasks: i) question analysis with question alignment, ii) feature extraction and 3) intermediate

QFF generation for alignment. In this section Alignment Index and learning the question structure with QFF is presented. In question analysis, the dependency features of two similar questions are collected. These dependency features are selected from the dependency parsers and an intermediate QFF and AI is generated. Further, it helps in designing an alignment learning algorithm. It learns the structure of a particular question using AI.

The solutions to each of the mentioned sub-problems can be applied in sequence to give a mapping of a question to another similar question. Alignment operators, define the space of possible answers for a given question, and a scoring function, which returns a real-valued confidence for a derivation also used to represent QFF score.

5.3.1 Alignment Operators & Scoring Function

A model for two similar questions is designed for mapping a question q_1 to another similar question q_2 by applying Learning Operators (L_o). Each operator O ϵ L_o takes a state object s ϵ features as input and returns a set o(s) $\in$ States of successor states as output. State objects encode intermediate values that are used during the question mapping procedure. In this implementation, the operator set combines both basic features (lexical, syntactic, semantic features) and new structural features. Incorrect mapping can be derived from most similar questions. To compute the confidence of a derivation, the algorithm uses a scoring function. The scoring function computes a real value for a given derivation, where large, positive scores are assigned to high-confidence derivations; the score of a derivation $d = (o, s, k)$ can be written as:

$$FeatureScore(q|q^{'}, w) = \sum_{a}^{b} w.f(s_0, s_{i-1}, o_i, s_i) \tag{5.4}$$

Where, **O** $\in$ Original Question and **S** $\in$ Similar Question and f is an n-dimensional feature function that maps a derivation and w is an n-dimensional similarity weight.

$$d^{*} or D^{*} = argmax_{(q,q^{'})} * score(d|f, w) \tag{5.5}$$

Where, **D/d** $\in$ Documents, the focus on finding a single answer with the highest confidence under the scoring function, which amounts to solving the following equation for an input question.

5.4 Dataset and Result Analysis

It is necessary to understand the difference between Logical Form (LF) and Intermediate Feature Score (IFS) used for Alignment Index (AI). IFS for AI is generated from LF of the text. Let it understand by a question: "Which is the highest mountain in Africa?", the Geo dataset has a generated Logical Form of this question *(A,highest(A,(place(A),loc(A, B),const(B, stated(Africa))).* LFS, for the same question, is a feature value, not a representation: say, LFS for AI in this work is 0.762. The value of LFS is calculated from various feature values extracted and described in next section.

5.4.1 Dataset Used

To evaluate the performance of the proposed approach, one of the most popular publicly available WebQuestion dataset is used. The WebQuestions dataset has 5500 questions along with their answers and top-ranked domains or documents which are collected from Freebase knowledge base. To make the experiments scientifically good to generate logical forms, publicly available TREC Question Classification (TQC) dataset and KBC question dataset are used. In TQC, questions and their category are mapped that is helpful in determining the Topic Word that is used for question alignment. If the answer type of two questions is similar, one can state that question might have been asked differently, but it should provide the same answer. There are approx 500 questions are collected for KBC dataset. This benchmark dataset consists of at least two similar questions. A detailed overview of the dataset viz. a total number of question type is mentioned in Table 5.3.

Table 5.3: Total questions containing Q-words with their expected answer type

500	**Question Word**	**Expected Answer Type**
283	what	Information related to Topic Word
180	who/where	Asking about person/place
33	when/which	Asking about time/choice
2	how + adj/adv	Asking manner
2	how far/long/much/many	distance/length/quantity/quantity

5.4.2 Results and Discussions

For setting up for experiments, all possible question features are collected and the relevant features are selected with feature selection techniques. Stanford parser is attached to automate the question processing phase to get question part of speech tags, question dependencies and named entities for feature space. Structural features are combined with other features to form the feature form. The combination of all relevant features provides the final QFF score and the AI score for each question is calculated. QFF score is calculated from all the features, and AI score denotes more than QFF score as it is calculated from QFF score and Topic Word alignment.

5.4.2.1 Feature Based Alignment

It is important that how to decide the individual feature value. These values are provided to a formula to calculate the final feature score. The formula is described and tested over 100 KBC dataset questions having at least two similar questions. Extraction algorithms for all features have been discussed in earlier sections. Document for example feature score has a passage *Indian tennis star Sania Mirza was today appointed 'Brand Ambassador' of Telangana.*

Lexical Score- It is shown here that how relevant lexical features are extracted from a question. The Lexical Feature Score or Lexical Score is calculated using lexical features. For an example from KBC dataset, *'Which sportswoman was made the brand ambassador of the newly formed state of Telangana?'* $\in \mathbf{Q}$ and, Indian tennis star Sania Mirza was today appointed *'Brand Ambassador'* of Telangana? $\in \mathbf{D}$. Lexical features are extraction here and individual feature score is calculated from these is shown in Table 5.4. Equation 5.6 is showing the value of R^2 for Unigram features. Similarly Equation

Table 5.4: Various systems with their intermediate form accuracy

Lexical	1	Unigram U_n	$U_nScore = \frac{TF}{SQ_{tf}} = \frac{5}{12} = 0.417$
	2	Bigram B_i	$B_iScore = \frac{TF}{SQ_{tf}} = \frac{3}{12} = 0.25$
	3	Trigram T_i	$T_iScore = \frac{TF}{SQ_{tf}} = \frac{5}{12} = 0.417$
	4	Wh-word W_w	$W_wScore = \frac{W_wW}{SQ_{W_ww}} = \frac{1}{12} = 0.083$
	5	Word Shape W_s	$W_sScore = \frac{WS}{WS_{SQ}} = \frac{4}{12} = 0.33$
	6	Q^n Length Q_l	$Q_lScore = \frac{Q^nL}{SQ^nL} = \frac{12}{13} = 0.923$
		L_e	**Average** $= \frac{\sum L_eScore}{\sum No.ofL_e} = \frac{2.24}{6} = \mathbf{0.403}$
Syntactic	1	Tagged U_n T_u	$T_uScore = \frac{TU}{SQ_{tu}} = \frac{12}{13} = 0.33$
	2	POS Tag P_t	$P_tScore = \frac{PT}{SQ_{pt}} = \frac{5}{7} = 0.417$
	3	Headword H_w	$H_wScore = \frac{HW}{SQ_{hw}} = \frac{10}{12} * \frac{10}{13} = 0.638$
	4	H_w Tag H_t	$H_tScore = \frac{HT}{SQ_{ht}} = \frac{20}{12} * \frac{10}{13} = 0.127$
	5	Focus Word F_w	$F_wScore = \frac{FW}{SQ_{fw}} = \frac{2}{12} = 0.166$
		S_y	**Average** $= \frac{\sum S_yScore}{\sum No.ofS_y} = \frac{1.678}{5} = \mathbf{0.335}$
Semantic	1	H_w Hypernym H_h	$H_hScore = \frac{HW_h}{SQ_{hWh}} = \frac{2}{3} = 0.667$
	2	Named Entity N_e	$N_eScore = \frac{NE}{SQ_{ne}} = \frac{3}{5} = 0.60$
	3	H_w NE H_n	$H_nScore = \frac{H_{ne}}{SQ_{hne}} = \frac{2}{5} = 0.40$
		S_e	**Average** $= \frac{\sum S_eScore}{\sum No.ofS_e} = \frac{1.667}{3} = \mathbf{0.556}$
	$FF_bScore = \sum_{i=1}^{n}[log(L_e \times S_y \times S_e)] = [log(0.403 \times 0.335 \times 0.556)]$ $= [log(L_e) + log(S_y) + log(S_e)] = 0.264 + 0.252 + 0.274 = \mathbf{0.79}$		

5.7, 5.8, 5.9, 5.10, and 5.11 are showing the respective feature line and value of R^2 of Bigram, Trigram, Wh-word, Word Shape and Question length feature respectively.

1) Unigrams (U_n)- Unigrams of the questions are tagged as, (Which, 1) (sportswoman, 2) (was, 3) (made, 4) (the, 5) (brand, 6) (ambassador, 7) (of, 8) (the, 9) (newly, 10) (formed, 11) (state, 12) (of, 13) (Telangana, 14). Refer table 5.4 to see the feature score calculation of Unigram. U_n feature regression line is shown in Equation 5.6.

$$U_n = Y_1 = 0.001x_1 + 0.529 - (R_1^2 = 0.18) \tag{5.6}$$

2) Bigrams (B_i)- Bigrams of the questions are tagged as, (Which-sportswoman, 1) (sportswoman-was, 2) (was-made, 3) (made-the, 4) (the-brand, 5) (brand-ambassador, 6) (ambassador-of, 7) (of-the, 8) (the-newly, 9) (newly-formed, 10) (formed-state, 11) (state-of, 12) (of-Telangana, 13). Refer table 5.4 to see the feature score calculation of Bigram. B_i feature regression line is shown in Equation 5.7.

$$B_i = Y_2 = 0.041x_2 + 0.639 - (R_2^2 = 0.13) \tag{5.7}$$

3) Trigrams (T_r)- Trigrams of the questions are tagged as, (Which-sportswoman-was, 1) (sportswoman-was-made, 2) (was-made-the, 3) (made-the-brand, 4) (the-brand-ambassador, 5) (brand-ambassador-of, 6) (ambassador-of-the, 7) (of-the-newly, 8) (the-newly-formed, 9) (newly-formed-state, 10) (formed-state-of, 11) (state-of-Telangana, 12). Refer table 5.4 to see the feature score calculation of Trigram. T_r feature regression line is shown in Equation 5.8.

$$T_r = Y_3 = 0.061x_3 + 0.739 - (R_3^2 = 0.23) \tag{5.8}$$

3) Wh-word (W_w), Word Shape (W_s) and Q-Length (Q_l)- W_w word of the example is *which*, W_s feature provide (UPPERCASE, 2), Q_l = 13. Refer table 5.4 to see the feature score calculation of Wh-word, Word Shape and Question Length. The feature regression line of W_w, W_s and Q_l is shown in Equation 5.9, 5.10, and 5.9.

$$W_w = Y_4 = 0.001x_4 + 0.529 - (R_4^2 = 0.38) \tag{5.9}$$

$$W_s = Y_5 = 0.063x_5 + 0.542 - (R_5^2 = 0.32) \tag{5.10}$$

$$Q_l = Y_6 = 0.052x_6 + 0.461 - (R_6^2 = 0.42) \tag{5.11}$$

5.4.3 Multiple Regression Analysis on Features

To calculate the final score of basic and proposed features, the formula is obtained from Multiple Regression (MR) techniques on feature scores. In MR the value of R^2, also known as the coefficient of determination is generally used statistic to estimate model fit.

R^2 is 1 minus the ratio of residual variability. While the variability of the residual values nearby the regression line corresponding to the overall variability is small, regression equation is well fitted. For example, if there is no link between the X and Y variables, then the ratio of the residual variability (Y variable) to the initial variance is equal to 1.0. Then R^2 would be 0. If X and Y are perfectly linked then the ratio of variance would be 0.0, making $R^2 = 1$. In most cases, the ratio and R^2 will be between these 0.0 and 1.0. If R^2 is 0.2 then we understand that the variability of the Y values nearby the regression line is 1-0.2 times the original variance; in other words, we have explained 20% of the original variability and left with 80% residual variability. The equation 5.12 is used for all the feature and tested on MR of available features. The regression line to of the individual feature to calculate QFF and DFF is shown in Figure 5.13.

$$Y = QFFScore = a + b_1 * L_e + b_2 * S_y + b_3 * S_e \tag{5.12}$$

5.4.3.1 Intermediate QFF Score

The logical form is used to query a knowledge base. Intermediate QFF generated in this work is not bothered about querying any KB, but it is used to represent the question to its QFF weight and then to map it with another QFF weights. These QFF weights are the QFF scores calculated from the formula shown in equation 5.13 (generated from RA of features). In the equation 5.14 L_e represents the lexical features, S_y represents the syntactic features, S_e represents the semantic features and S_t represents the structural features. In the equation 5.12 putting the value of $a = 0$ and as the value is used for the alignments the effect of coefficient can be ignored once and treated as $b_1 = b_2 = b3 = 1$. QFF score is shown in equation 5.13.

$$FFScore(QFF/DFF) = L_e + S_y + S_e \times S_t \tag{5.13}$$

$$FFScore = (logL_e + logS_y + logS_e) \times S_t \tag{5.14}$$

QFF and also a (Document Feature Score) DFF score can be compared to question and document as there are about each other. One can also use multiple regression

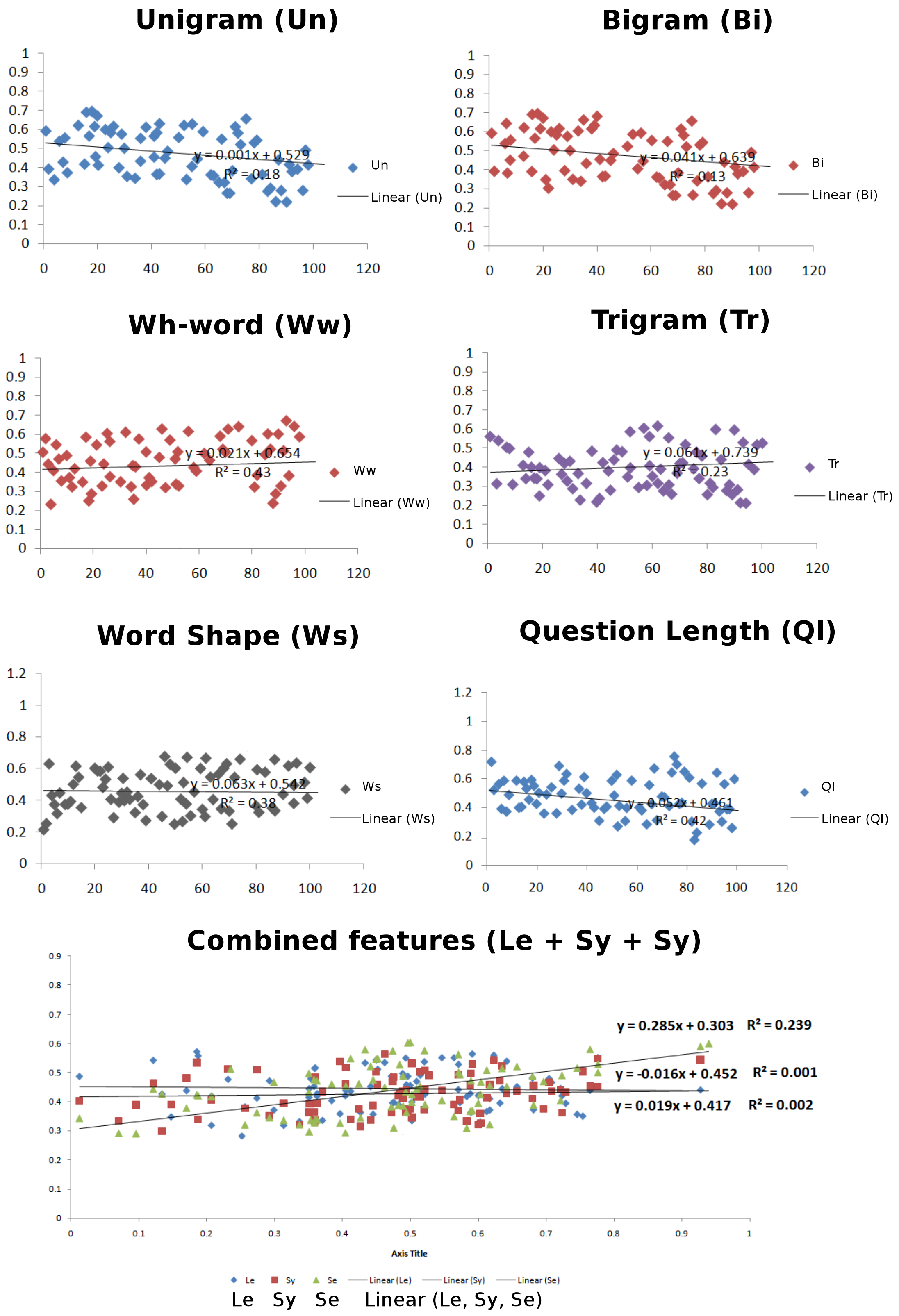


FIGURE 5.13: Regression line of faetures to calculate the QFF, DFF formuula

coefficients to compare QFF and DFF. In this work, the complete dataset has questions paired with options and answers and documents having the answer-evidence are ranked.

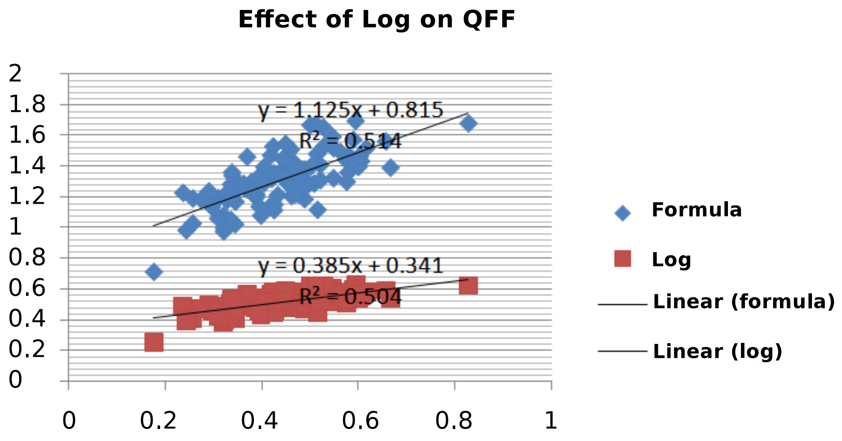


FIGURE 5.14: Diagram showing the stage where question alignment is required

The equation is merely showing that all feature are contributing equally to calculate QFF and QFF score. Equation 5.15 calculates the FFScore of each question in dataset.

$$FFScore = \sum_{i=1}^{n}[log(L_e \times S_y \times S_e)] \times S_t \tag{5.15}$$

5.4.3.2 Final Feature Extraction for Alignment

A value of individual feature and their average value is calculated in Table 5.5 this individual feature value calculates the final feature score. The formula calculates the final score in Equation 5.15.

The final feature extraction Table 5.6 is showing an example question and their two similar questions. In Table 5.5 original question is obtaining a score **0.58** and two similar questions ae getting a score **0.47** and **0.61** respectively. The score value is ranging from **[0, 1]** and the testing results for 500 KBC questions show that two questions which are similar can have a difference in their feature score ranging from **[0, 0.2]**. A feature score difference more than this value are not similar and will not be considered for question alignment. It is also noticed that the accuracy of alignment is increased while T_w is added to feature-based alignment. It is shown in Table 5.6.

Table 5.5: Original question aligned with two similar question

	Lexical(L_e)		*Syntactic*(S_y)		*Semantic*(S_e)		S_t	
Question	U_n	0.42	T_u	0.58	H_h	0.40	0.25	**= 0.58**
	B_i	0.45	P_t	0.52	N_e	0.23	–	
	T_i	0.48	H_e	0.41	H_n	0.52	–	
	W_w	0.57	H_t	0.38	–	–	–	
	W_s	0.13	F_w	0.43	–	–	–	
	Q_l	0.09	–	–	–	–	–	
AVG		**0.52**		**0.44**		**0.48**	**0.25**	
Question	U_n	0.40	T_u	0.48	H_h	0.37	0.33	**= 0.47**
	B_i	0.36	P_t	0.45	N_e	0.22	–	
	T_i	0.34	H_e	0.55	H_n	0.46	–	
	W_w	0.65	H_t	0.44	–	–	–	
	W_s	0.11	F_w	0.34	–	–	–	
	Q_l	0.07	–	–	–	–	–	
AVG		**0.44**		**0.56**		**0.39**	**0.33**	
Question	U_n	0.38	T_u	0.55	H_h	0.42	0.25	**= 0.61**
	B_i	0.41	P_t	0.43	N_e	0.24	0.66	
	T_i	0.45	H_e	0.37	H_n	0.56	–	
	W_w	0.49	H_t	0.31	–	–	–	
	W_s	0.14	F_w	0.42	–	–	–	
	Q_l	0.11	–	–	–	–	–	
AVG		**0.46**		**0.52**		**0.57**	**0.45**	

Table 5.6: Combination of feature and topic alignment

Topic Word (T_w) Match								
	Individual T_w							
	L_e	$\%E$	S_y	$\%E$	S_e	$\%E$	S_t	$\%E$
SimQue$_1$	178	64.4	167	66.6	270	46	255	49
SimQue$_2$	211	57.8	189	62.2	243	51.4	247	50.6
	Combined T_w							
	T_w **(Basic Features)**				T_w **(Proposed Features)**			
	$L_e + S_y + S_e$			$\%E$	$L_e + S_y + S_e$			$\%E$
SimQue$_1$	296			40.8	411			17.8
SimQue$_2$	255			49	372			25.6
Topic Word and Feature Based Match								
	Combined T_w **and Feature Based Score** (QFF_{Score})							
	$T_w + QFF_{Score}$							$\%E$
SimQue$_1$	397							20.6
SimQue$_2$	380							24

5.4.3.3 Question Topic Based Alignment

Question topic based alignment is added with feature-based alignment to improve the aligner accuracy. The Topic Word (T_w) based alignment is shown in Table 5.6. Table 5.6 is showing the number of topic words related to which feature category. Error percentage $(\%E)$ is showing that which category does not contain the extracted Topic Word. For the experiments first of all the dataset of WebQuestions is analyzed and from

Table 5.7: Topic word algorithms (TWI and DWI)

Topic Word (T_w) Match								
Question Set	**TWI** (%)		**DWI** (%)		F_1 **Measure** (%)		**Change** (%)	
	P	R	P	R	TWI	DWI	TWI	DWI
1 - 1500	72.5	62.6	70.0	64.3	67.2	67.1	-	-
1501 - 3000	78.4	67.3	74.5	68.5	72.4	71.4	+5.24	+4.35
3001 - 4500	81.2	71.5	78.6	76.5	76.1	77.5	+3.98	+6.16
4501 - 5810	79.8	68.5	82.7	79.3	73.7	80.9	-2.33	+3.43
AVG	77.9	67.4	76.4	72.1	72.3	74.2	-	-

this dataset, Topic Words are separated in a file because the proper identification of this Topic word is the goal. Once the list of Topics are divided, the question is parsed and made the dependency tree is made from Stanford dependency parser. The algorithm for correctly finding the Topic word is shown in TWI, which uses the Stanford named entity tagger for marking likely Topic words during the algorithm. In comparison with benchmark WebQuestions accuracy (around 40%) attained using jacana-freebase [20]. TOSP approach attained between 70%-80% while focusing on Topic Words only. For evaluation, in this case, a score with a partial weight is calculated. For every question, first, compute its precision (P) and its recall (R) by taking the dataset as gold standard Topics as the relevant Topics and the predicted Topics at the retrieved set. Taking an average of P and R over all Topics. This T_w extraction table is shown in Table 5.6 and TWI and DWI algorithm results are shown in Table 5.7.

5.4.4 Statistical Measures and Validation: T-Test

A T-test's statistical significance indicates whether or not the difference between two groups' averages most likely reflects a real difference in the population from which the groups were sampled (Adams et al. 1993).

To validate the proposed features and to find the significance between classification algorithms, a statistical test is performed which discovers whether the proposed features on NLQA and EG are significant or not. This test is used to determine experimentally the probability of incorrectly identifying the Type I error. Type I error is the incorrect rejection of a true null hypothesis, or False Positive. It identifies an effect in machine learning that is not present. Thus to evaluate the performance of proposed features on

Table 5.8: T-test measures on proposed and state-of-the-art datasets for question alignment using feature-base, topic-based and combined

T-test of datasets on basic and proposed features				
	Individual Features			
	L_e	S_y	S_e	S_t
KBC	5.47	5.41	4.57	4.42
TREC	4.56	4.78	6.54	4.48
WebQ	5.48	6.47	5.74	5.64
	Combined Features			
	(Basic Features)		**(Proposed Features)**	
	$L_e + S_y + S_e$		S_e **with (DP + DR + NER)**	
KBC	4.65		4.78	
TREC	4.69		4.99	
WebQ	4.88		5.98	

proposed algorithms (NLQA and EG),

$$t_{value} = \frac{X_T - X_C}{\sqrt{\frac{var_T}{n_T} + \frac{var_C}{n_C}}} \tag{5.16}$$

Where, X_T is tested sample variablity and X_C is available class sample variability calculates the value of t_{value} from Equation 5.16. As the T-sample distribution is a chi-square distribution, therefore, based on the observed t_{value} value experimentally, it is found that if the observed t_{value} value < 4.145 then the null hypothesis is accepted otherwise the alternate hypothesis is accepted for t_{value} value > 4.145 (possible experimented value of the threshold).

In this thesis, to test and validate the proposed feature extraction rule with state-of-art datasets WebQ and TREC and T-test is applied. Let the null hypothesis (H_0) is defined as:

1. **($\mathbf{H}_0$)**: Proposed features with the state-of-the-art and proposed dataset have same performance and,
2. **($\mathbf{H}_1$)**: Proposed features are better than the state-of-the-art features on available and proposed dataset.

In the experimentation result shown in Table 5.8 with T-test algorithm, it is found that the average observed value of the T-test in all cases 4.42, 4.65, 5.47, 6.54 > 4.145. So

The null hypothesis is rejected and accept the alternate hypothesis. Finally, the validation rule suggest that proposed features on EG and NLQA are better than state-of-art features.

5.5 Summary

In this chapter, the ability of aligners to recover gold-standard alignments ise evaluated. But since feature-based alignment is just one component of the NLA problem, Topic Words are also examined the impact of different aligners on the ability to recognize valid inferences. The proposed work is focusing on aligning two similarly asked questions. Initially, a question is analyzed, and lexical, syntactic, semantic and new structural features are extracted. These features are used to give an intermediate QFF to get an AI score. AI represents the question which can be compared with an another similar question. After the mapping two question.

Proposed work is compared with MANLI aligner also compared with Passage term-matching, Skip bigram, and Textual alignment. Question Feature Form model uses lexical, syntactic, semantic and structural features. The combination of these features gives better results as compared to MANLI for open-domain questions.

Chapter 6

Conclusions and Future Work

The area of question answering is an interesting and popular research direction due to a high number of applications. User asks a query to the search engine and expects an exact answer than a list of documents containing that answer. The improvement of methods for the open-domain-question answering systems is one of the notable parts of this field. Recently, users want to save their time to open and search the document for the answer. Users want to get the exact answer for the factoid, list and as well as the complex descriptive queries. QA has a long history; therefore, a lot of study and analysis is already available in the literature. Still, a complete ODQA system is not so promising that it can provide an answer to every question related to every domain. Therefore, the existing algorithms for QA systems need to be improved. These state-of-the-art QA systems can be enhanced using the question alignment at the question processing stage and proper evidence gathering. In this Chapter, the conclusions from this research work are outlined along with the potential future work directions.

6.1 Conclusions

This thesis used a feature-based model to improve QA algorithms with the addition of reference algorithms and topic word identifications. For feature-based approach, many basic features have been extracted and some features are proposed. Proposed features have structural features from the text dependency. These features are added with the design principles and named entities to improve feature performance.

These basic and proposed features are further useful in proposed components of the QA system. There are two proposed elements in this work, i) Natural Language Question Alignment, and ii) Evidence Gathering. In NLQA, a complex question is aligned with a simple, similar question. This alignment improves the performance of the overall QA systems. Because, after such an alignment a QA systems able to answer complex questions. Next, in EG, a question possible evidence from the source document to support the answer. This evidence gathering algorithm further improves the accuracy of a QA system. As, after a proper evidence gathering process a QA system will be more confident about the answer and misleading options.

This research work gets the answers of the question analysis, and evidence gathering failures raised in section 1.1 and also meets the objectives. The answers to these research failure is discussed below,

1. From the experiments, is observed that the proposed features are very important to extract long distance relations. These relations are very useful to get the link between those words that could not be linked otherwise. It is also observed that the combination of basic and proposed features improves the accuracy of both NLQA and EG.

2. By reducing the unnecessary features, the performance of the both approaches is improved. It can be concluded from the experiments that mRMR feature selection method enhances the performance by dropping the unnecessary features. The combination of the features on semantic basic improves the performance of complete answer extraction process in QA systems. This combination of existing features with proposed features solves the problem of aligning the complex question with a similar, simple question. This alignment removes the risk of question analysis failure for complicated natural language questions.

There are some other finding in this thesis given below,

1. The structural features proposed in this work are added with design principles and named entity. This addition improves the accuracy of overall answer extraction on KBC dataset. It is shown in Table 3.5 in chapter 3.

2. By analyzing the influence of basic features and with the combination proposed structural features. It can be concluded that besides the rare case of the question combination, accuracy is alway improved. This observation can be seen in Table 3.4 in chapter 3.

3. Proposed features to link the long distance semantic relations among the words raises the chances of gathering most prominent evidence. This improves the performance of the overall QA system. It is presented in chapter 4. In the addition of this; it is also apparent in chapter 4 that adding reference-based evidence with feature-based evidence further improves the performance.

4. The impact of the topic word on the alignment and the improvement in overall alignment approach is analyzed in chapter 5. In addition to this, the effect of adding the topic word to the feature-based alignment is also measured. These results can be seen in Table 5.10 in chapter 5.

5. The new features (called, structural features) from the dependency parse with an addition of named entities have been proposed. The algorithms are designed to extract the basic features (e.g. L_e, S_y, S_e).

6. A feature-based (with a combination of basic and proposed features) and reference-based (with indirect co-reference in the text) algorithms have been proposed for gathering better evidence from the unstructured text.

7. An algorithm to calculate the distance among indirect question referents have been proposed for evidence gathering approach.

8. A feature-based and topic-based (with Topic Word of the question) algorithms have been proposed for question alignment of complex questions.

9. Topic words are extracted from the Topic Word Identification algorithm and a domain word related to that topic word is retrieved from the Domain Word Identification algorithm, useful in knowledge-based QA systems.

10. A named entity based recognition system is designed to enhance Gazetteer performance for Indian domain questions (e.g. KBC questions). These named entities are added to the structural features for improving the performance.

11. A question alignment learning algorithm is designed that learn the possible question alignments.

6.1.1 Contributions

The contributions of the thesis are summarized as follows.

1. The structural features have been proposed, which are constructed using dependency rules of the question and document. The accuracy is improved by adding named entities to the proposed features.

2. Two Evidence Gathering (EG) approaches have been proposed; the first approach is based on the combination of basic and proposed features and the second approach relies on the indirect reference available in the question and the document.

3. The feature extraction algorithms for the basic and proposed features and also calculated the relevance of each feature by their individual answer extraction accuracy on QA systems.

4. Two Natural Language Question Alignment (NLQA) approaches; the first is a feature-based approach and the second is a topic-based alignment approach. For the topic-based NLQA approach, two algorithms have been proposed two algorithms: i) Topic Word Identification (TWI) and, ii) Domain Word Identification (DWI) algorithm.

5. A question alignment learning algorithm have been proposed for similarly asked questions using alignment index. A valuable KBC question dataset for alignment is also generated which include at least two similarly asked questions along with four options for a particular question.

6.2 Future Works

This area of research has attracted many data analysts and industrialists. The reason behind is the possibilities of improving the searching performance of current search engines. Further advancements in the area of question analysis are possible. Some possible approaches to move further in this direction as a future work are as follows.

1. If question processing has not done accurately then it is hard to reach to the exact answer even if the answer is available in the document. Therefore, new algorithms can be designed and added to the question alignment for deep question processing, that is the backbone of an ODQA system.

2. Apart from the question alignment, sentence alignment can be applied while retrieving relevant documents for expanding the source for missing evidence.

3. Finally, While dealing with many questions at question processing stage, it may become difficult to process them together so BIG data methods can be applied to the reduction of algorithmic processing time is another important issue for real-time QA for more complex or multiple questions. The processing time for these should not exceed the order of seconds. The role and use of the BIG data analytics in the QA domain may give the fast real time analysis on single and multiple questions asked.

Publications

1. **Journals:**

   (a) Sharma L. K., Mittal N., "Prominent Feature Extraction for Answer Evidences in Question Answering", Journal of Intelligent and Fuzzy Systems, IOS Press, vol. 32, no. 4, pp. 2923-2932, 2017. (Published **SCIE**) **pdf**

   (b) Sharma L. K., Mittal N., "An Algorithm to Calculate Semantic Distance among Indirect Question Referents", International Bulletin of Mathematical Research, Volume 2, Issue 1, Pages: 6-10, ISSN: 2394-7802, March 2015. (Published) **pdf**

   (c) Sharma L. K., Mittal N., "Answer Extraction in Question Answering System using Structure Features and Dependency Principals", International Journal of Computers and Electrical Engineering, (SCIE). (Under review)

   (d) Sharma L. K., Mittal N., "Feature Based Question Alignment and Learning using Structural Features and Word Similarity", In ACM transactions on Information Systems (SCI). (Under review)

2. **International Conferences:**

   (a) Sharma L. K., Mittal N., "Topic Oriented Semantic Parsing", 9^{th} IEEE International Conference on Semantic Computing (IEEE-ICSC 2015), Anaheim, California, USA, 2015. (Published) **pdf**

   (b) Sharma L. K., Mittal N., "Dependency Extraction for Knowledge-based Domain Classification", 12^{th} Conference on NLP (ICON 2015), Kerala, In ACL Proceedings 2015. (Published) **pdf**

(c) Sharma L. K., Mittal N., “Named Entity Based Answer Extraction form Hindi Text Corpus Using n-grams”, 11th Conference on NLP (ICON 2014), Goa, In ICON Proceedings 2014. (Published) **pdf**

(d) Sharma L. K., Mittal N., “Structural Feature and Named Entity Extraction Using Enhanced-CRF and Gazetteer”, In Proceedings International Conference on Computational Technologies (ICCT 2014), Jaipur, 2014. (Published)

Appendix A

Sample Dataset: KBC, TREC and WebQuestions

A.1 KBC

This dataset of question has 500 questions from the Indian game show (KBC). The questions are collected from a particular episode and similar questions are added in the question set manually. There are few example questions and the format of KBC dataset is shown,

Format of KBC data: Question (original), Question (similar), Option (1 to 4), Answer

1. **Original:** In medicine, which of these is usually denoted by 120/80 for an adult?
2. **Similar:** In medical science, 120/80 used for an adult?
3. **Option (1-4):** a. Normal Pulse b. Normal Hearing c. Normal vision d. Normal Blood Pressure?
4. **Answer:** Normal Blood Pressure

1. **Original:** In the film 2 states, a Punjabi boy falls in love with a ______ girl?
2. **Similar:** In the film 2 states,Ali Bhat play the role of ______girl?

3. **Option (1-4):** a. Bengali b. Marathi c. Tamil d. Malayali
4. **Answer:** Tamil

A.2 TREC

TREC has had a question answering track since 1999; in each track for fact-based, short-answer questions that can be drawn from any domain. Answer patterns are provided here for the TREC QA collections. A submission for the (main) QA task in each TREC format of a response is like,

1. **qid:** is the question number
2. **Q_0:** is the literal Q0
3. **rank:** (1-5) is the rank of this response for this question
4. **score:** is a system-dependent indication of the quality of the response tag is the identifier for the system
5. **tag:** is the category of the question
6. **answer-string:** is the text snippet returned as the answer. Answer string (only)

TREC Extraction: (Example)- 1 1 NYT20000727 0012 -1 July 18

A.3 WebQuestions

Both datasets are provided in JSON format. WebQuestions contains 3,778 training examples and 2,032 test examples.

1. **url:** http://www.freebase.com/view/en/justin_bieber
2. **targetValue:** (list (description "Jazmyn Bieber ") (description "Jaxon Bieber "))
3. **utterance:** what is the name of justin bieber brother?" Justin Biebr *in* person

Bibliography

Agarwal, B. and Mittal, N., “Prominent feature extraction for review analysis: an empirical study”, Journal of Experimental & Theoretical Artificial Intelligence, pp. 485-498, 2016.

Agarwal, B., and Mittal N., “Semantic feature clustering for sentiment analysis of English reviews”, IETE Journal of Research, vol. 6, pp. 414-422, 2014.

Aguado A. S., Mark S. N. and Montiel M. E., “Parameterizing arbitrary shapes via Fourier descriptors for evidence-gathering extraction”, In Computer Vision and Image Understanding, pp. 202-221, 1998.

Altschul S. F., Warren G., Miller W., Eugene W. M. and Lipman D. J., “Basic local alignment search tool”, In Journal of molecular biology, pp. 403-410, 1990.

Andrieu C., Nando D. F., Doucet A., and Jordan M. I., “An introduction to MCMC for machine learning”, In Machine learning, vol. 1-2, pp. 5-43, 2003.

Fader, A., Luke S. Zettlemoyer, and Etzioni O., “Paraphrase-Driven Learning for Open Question Answering.” In ACL, pp. 1608-1618, 2013.

Baluja, S. and Davies, S., “Using Optimal Dependency-Trees for Combinatorial Optimization: Learning the Structure of the Search Space (No. CMU-CS-97-107)”, Computer Science Department - Carnegie Mellon University, 1997.

Bär, Daniel, Torsten Zesch, and Iryna Gurevych., “DKPro Similarity: An Open Source Framework for Text Similarity”, In ACL (Conference System Demonstrations), pp. 121-126, 2013.

Bayer, S., Burger, J., Ferro, L., Henderson, J. and Yeh, A., “MITREs Submissions to the EU Pascal RTE Challenge”, In Proceedings of the Pattern Analysis, Statistical Modelling, and Computational Learning (PASCAL) Challenges Workshop on Recognising Textual Entailment, 2005.

Bishop, Lawrence C., and George Ziegler. ”Open ended question analysis system and method.” U.S. Patent No. 4,958,284. 18 Sep. 1990.

Brenner S. W. and Joseph J. Schwerha, “Transnational evidence gathering and local prosecution of international cybercrime”, In J. Marshall J. Computer & Info., pp. 347, 2001.

Brill E., Susan D., and Michele B., “An analysis of the AskMSR question-answering system”, In Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol. 10, pp. 257-264, 2002.

Brockett C., “Aligning the RTE corpus”, Technical report, Microsoft Research, 2007.

Bunescu R., Huang Y., “Towards a general model of answer typing: Question focus identication”, In Proceedings of The 11th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), pp. 231-242, 2010.

Burke R. D., Kristian J. H., Vladimir K., Steven L. L., Noriko T. and Schoenberg S., Question answering from frequently asked question files: Experiences with the faq finder system.”, In AI magazine, vol. 2, pp. 57, 1997.

Chambers N., D. Cer, T. Grenager, D. Hall, C. Kiddon, B. MacCartney, M.C. de Marneffe, D. Ramage, E. Yeh, and C.D. Manning, Learning alignments and leveraging natural logic, In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pp. 165-170, 2007.

Chang, M.W., Srikumar, V., Goldwasser, D. and Roth, D., “Structured output learning with indirect supervision”, In Proceedings of the 27th International Conference on Machine Learning (ICML), pp. 199-206, 2010.

Chatterji M., Lawrence W. G., and Shiriki K., “LEAD A Framework for Evidence Gathering and Use for the Prevention of Obesity and Other Complex Public Health Problems”, In Health Education Behavior, pp. 85-99, 2014.

Cherry C. and Dekang L., “A probability model to improve word alignment”, In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 88-95, 2003.

Chowdhury G. G., “Natural language processing”, In Annual review of information science and technology, vol. 1, pp. 51-89, 2003.

Chu-Carroll, Jennifer J. P., Welty C., Czuba K. and Ferrucci D., "A multi-strategy and multi-source approach to question answering", In IBM Research Centre, Newyork, 2006.

Corley C., and Mihalcea R., "Measuring the semantic similarity of texts", In Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment, pp. 13-18, 2005.

Cortes C. and Vapnik V., "Support-vector networks", In Machine learning, vol 3, pp. 273-297, 1995.

DeNero J. and Klein D., "The complexity of phrase alignment problems", In Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers, pp. 25-28, 2008.

Edgar R. C., "MUSCLE: multiple sequence alignment with high accuracy and high throughput", In Nucleic acids research, pp. 1792-1797, 2004.

Ferrucci D. A., "Introduction to This is Watson", In IBM J. Res. Dev., vol. 56, pp. 1:1-1:15, 2012.

Ferrucci D., Brown E., Chu-Carroll J., Fan J., Gondek D., Kalyanpur A. A., Lally A., Murdock J. W., Nyberg E., Prager J. and Schlaefer N., "Building Watson: An overview of the DeepQA project", AI magazine, pp. 59-79, 2010.

Fowler, B. A. and Woods, J. S., "Ultrastructural and biochemical changes in renal mitochondria during chronic oral methyl mercury exposure: the relationship to renal function", Experimental and Molecular Pathology, pp. 403-412, 1977.

Frakes, William B., and Ricardo Baeza-Yates., "Information retrieval: data structures and algorithms", In Semantic-Social-Networks, pp. 125-137, 1992.

Goldberg, David E., and John H. Holland., "Genetic algorithms and machine learning", In Machine learning, vol. 2, pp. 95-99, 1998.

Gomaa, Wael H., and Aly A. Fahmy., "A survey of text similarity approaches," In International Journal of Computer Applications, pp. 13-21, 2013.

Green Jr, B.F., Wolf, A.K., Chomsky, C. and Laughery, K., Baseball: an automatic question-answerer. In western joint IRE-AIEE-ACM computer conference (ACM), pp. 219-224, 1961.

Hatzivassiloglou, Vasileios, Judith L. Klavans, and Eleazar Eskin., "Detecting text similarity over short passages: Exploring linguistic feature combinations via machine learning", In Proceedings of the 1999 joint sigdat conference on empirical methods in natural language processing and very large corpora, pp. 203-212, 1999.

Hall, Tom A., "BioEdit: a user-friendly biological sequence alignment editor and analysis program for Windows 95/98/NT", In Nucleic acids symposium series, Information Retrieval Ltd, vol. 41, pp. 95-98, pp. 1979-2000, 1999.

Hendrix, G.G., Sacerdoti, E.D., Sagalowicz, D. and Slocum, J., "Developing a natural language interface to complex data", ACM Transactions on Database Systems, pp. 105-147, 1978.

Heng Li, Bob Handsaker, Alec Wysoker, Tim Fennell, Jue Ruan, Nils Homer, Gabor Marth, Goncalo Abecasis, and Richard Durbin, "The sequence alignment/map format and SAMtools", In Bioinformatics, pp. 2078-2079, 2009.

Hermjakob U, Hovy EH, Lin CY, "Knowledge-based question answering", In Proceedings of the Sixth World Multiconference on Systems, Cybernetics, and Informatics, pp. 122-131, 2000.

Hermjakob, Ulf, Abdessamad Echihabi, and Daniel Marcu., "Natural Language Based Reformulation Resource and Wide Exploitation for Question Answering", In TREC, vol. 90, pp. 91, 2002.

Hovy, E. H., Gerber, L., Hermjakob, U., Junk, M., Lin, C. Y., "Question Answering in Webclopedia", In TREC, vol. 52, pp. 53-56, 2000.

Huang, Cheng-Hui, Jian Yin, and Fang Hou., "A text similarity measurement combining word semantic information with TF-IDF method", In Jisuanji Xuebao (Chinese Journal of Computers), pp. 856-864, 2011.

Hermjakob, Ulf, "Parsing and question classification for question answering", In Proceedings of the workshop on Open-domain question answering, Association for Computational Linguistics, vol. 12, pp. 1-6, 2001.

Imamura, Kenji., "Hierarchical Phrase Alignment Harmonized with Parsing", In NLPRS, pp. 377-384, 2001.

Islam, Aminul, and Diana Inkpen., “Semantic text similarity using corpus-based word similarity and string similarity”, In ACM Transactions on Knowledge Discovery from Data (TKDD) 2, pp. 10-19, 2008.

Jonathan B., and Liang P., “Semantic Parsing via Paraphrasing”, In ACL, pp. 1415-1425, 2014.

Jonathan B., Chou A., Roy F. and Liang P., “Semantic Parsing on Freebase from Question-Answer Pairs”, In Proceedings of EMNLP, pp. 136-157, 2013.

Jurczyk, Pawel, and Eugene Agichtein., “Discovering authorities in question answer communities by using link analysis”, In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, pp. 919-922, 2007.

Kanaiaupuni, Shawn Malia., “Reframing the migration question: An analysis of men, women, and gender in Mexico”, In Social forces, pp. 1311-1347, 2000.

Katz B. , Federico Mora, Gary Borchardt and Sue Felshin, “STARTMobile: An Intelligent Phone Assistant (STARTMobile: Using Language to Connect People to Mobile Devices)”, Infolab Research Group, MIT CSAIL, 2006.

Khoo S. T. and Adams R. J., “Quest: The Interactive Test Analysis System”, pp. 134-145, 1993.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R. and Dyer, C., “Moses: Open source toolkit for statistical machine translation”, In Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions, Association for Computational Linguistics, pp. 177-180, 2007.

Lally, Adam, John M. Prager, Michael C. McCord, Branimir K. Boguraev, Siddharth Patwardhan, James Fan, Paul Fodor, and Jennifer Chu-Carroll., “Question analysis: How Watson reads a clue”, In IBM Journal of Research and Development, 2012.

Loni B., “A survey of state-of-the-art methods on question classification”,In Literature Survey, Published on TU Delft Repository, 2011.

Li, Xin, and Dan Roth, “Learning question classifiers: the role of semantic information”, Natural Language Engineering, vol. 03, pp. 229-249, 2006.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., & Durbin, R., “The sequence alignment map format and SAMtools”, Bioinformatics, pp. 2078-2079, 2009.

MacCartney, B., Galley, M. and Manning, C.D., October. A phrase-based alignment model for natural language inference. In Proceedings of the conference on empirical methods in natural language processing, Association for Computational Linguistics, pp. 802-811, 2008.

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze., “Introduction to information retrieval”, vol. 1, pp. 156-163, 2008.

Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K. and Schasberger, B., March. The Penn Treebank: annotating predicate argument structure. In Proceedings of the workshop on Human Language Technology, Association for Computational Linguistics, pp. 114-119, 1994.

Markless, Sharon, and David Streatfield., “Gathering and applying evidence of the impact of UK university libraries on student learning and research: a facilitated action research approach”, In International Journal of Information Management, vol. 1, pp. 3-15, 2006.

Mazzetti, S.A., Kraemer, W.J., Volek, J.S., Duncan, N.D., Ratamess, N.A., GÓmez, A.L., Newton, R.U., Hakkinen, K.E.I.J.O. and Fleck, S.J., The influence of direct supervision of resistance training on strength performance. Medicine and Science in Sports and Exercise, pp. 1175-1184, 2000.

McCallum A., Bellare K., and Fernando P., “A Conditional Random Field for Discriminatively-trained Finite-state String Edit Distance”, In Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence (UAI), pp. 178-186, 2005.

McCord, M.C., Slot grammar. In Natural language and logic. Springer Berlin Heidelberg, pp. 118-145, 1990.

Mengqiu Wang and Christopher D. Manning., “Probabilistic tree-edit models with structured latent variables for textual entailment and question answering”, In

Proceedings of the 23rd International Conference on Computational Linguistics, COLING, USA, pp. 1164-1172, 2010.

Metzler, Donald, and W. Bruce Croft., “Linear feature-based models for information retrieval”,In Information Retrieval, vol. 3, pp. 257-274, 2007.

Michael Heilman and Noah A. Smith. 2010. Tree edit models for recognizing textual entailments, para-phrases, and answers to questions. In Proceedings of NAAC, pp. 1011-1019, 2010.

Michael R. and Anette Frank, Aligning predicates across monolingual comparable texts using graph-based clustering. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 171-182, 2012.

Mihalcea, R., Corley, C., & Strapparava, C., “Corpus-based and knowledge-based measures of text semantic similarity”, In AAAI, pp. 775-780, 2006.

Miller G. A., “WordNet: A Lexical Database for English,” In Communications of the ACM, vol. 38, pp. 39-41, 1995.

Mohler, M., and Rada irschmaMihalcea., “Text-to-text semantic similarity for automatic short answer grading”, In Proceedings of the 12th Conference of the European C hapter of the Association for Computational Linguistics, pp. 567-575, 2009.

Moldovan, Dan, Marius Pasca, Sanda Harabagiu, and Mihai Surdeanu., “Performance issues and error analysis in an open-domain question answering system”, In ACM Transactions on Information Systems (TOIS), pp. 133-154, 2003.

Moldovan D., Clark.Christine., Harabagiu Sanda., Maiorano Steve., “Cogex: A logic prover for question answering”, In Proceedings of NAACL, pp. 87-93, 2003.

Moore, Robert C., “A discriminative framework for bilingual word alignment”, In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 81-88, 2005.

Murdock, J. William, James Fan, Adam Lally, Hideki Shima, and B. K. Boguraev., “Textual evidence gathering and analysis”, In IBM Journal of Research and Development, vol. 3.4, pp. 1-8, 2012.

Ng H.T., Goh, W.B. and Low, K.L., “Feature selection, perceptron learning, and a usability case study for text categorization”, In ACM SIGIR Forum, ACM, vol. 31, pp. 67-73, 1997.

Pedoe W. T., “True Knowledge: Open-Domain Question Answering Using Structured Knowledge and Inference,” In Association for the Advancement of Artificial Intelligence, vol. 31, pp. 122-130, 2014.

Peng F., Weischedel R., Licuanan A., Xu J., “Combining deep linguistics ace pattern learning: A hybrid approach to chinese definitional question answering”, In Proceedings of EMNLP, pp. 307-314, 2005.

Pinto D., Branstein M., Coleman R., Croft W.B., King M., Li W. and Wei X., July. “QuASM: a system for question answering using semi-structured data”, In Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, ACM, pp. 46-55, 2002.

Ponte, Jay M., and W. Bruce Croft., “A language modeling approach to information retrieval”, In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 275-281, 1998.

Pradhan, S.S., Ward, W.H., Hacioglu, K., Martin, J.H. and Jurafsky, D., May. Shallow Semantic Parsing using Support Vector Machines. In HLT-NAACL, pp. 233-240, 2004.

Prager.M.John., “Open-domain question-answering. Foundations and Trends”, In Information Retrieval, pp. 91-231, 2006.

Qin, Tao, Tie-Yan Liu, Jun Xu, and Hang Li., “LETOR: A benchmark collection for research on learning to rank for information retrieval”, In Information Retrieval, pp. 346-374, 2010.

Quarteroni.Silvia. and Manandhar.Suresh.,”Designing an interactive open-domain question answering system”, In Natural Language Engineering, pp. 73-95, 2009.

Rasmussen, Carl Edward., “Gaussian processes for machine learning”, pp. 178-195, 2006.

Ravichandran, Deepak, and Eduard Hovy., "Learning surface text patterns for a question answering system", In Proceedings of the 40th annual meeting on association for computational linguistics, pp. 41-47 , 2002.

Reder, Lynne M., "Strategy selection in question answering", In Cognitive psychology, vol. 1, pp. 90-138, 1987.

Richard D. and Barbara E. A., "The recall of physical activity: using a cognitive model of the question-answering process", In Medicine & Science in Sports & Exercise, pp. 23-33, 1996.

Rocchio, Joseph John., "Relevance feedback in information retrieval", pp. 313-323, 1971.

Salton, Gerard, and Michael J. McGill., "Introduction to modern information retrieval", pp. 188-193, 1986.

Samuelsson, Yvonne, and Martin Volk., "Phrase alignment in parallel treebanks", In Proc. of the Fifth Workshop on Treebanks and Linguistic Theories, pp. 91-102, 2006.

Scharpf, Fritz W., "Judicial Review and the Political Question: A Functional Analysis",In The Yale Law Journal, pp. 517-597, 1966.

Severyn Aliaksei.,Moschitti.Alessandro.,"Automatic feature engineering for answer selection and extraction", In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 458-467, 2013.

Sharma L. K., Mittal N., "An Algorithm to Calculate Semantic Distance among Indirect Question Referents", International Bulletin of Mathematical Research, vol. 2, pp. 6-10, ISSN: 2394-7802, 2015.

Singhal A. and Kaszkiel M, "TREC-9", In TREC at ATT, 2000.

Steedman, M. and Baldridge, J., "Combinatory categorial grammar", Non-Transformational Syntax: Formal and Explicit Models of Grammar, Wiley-Blackwell, 2011.

Stefik, M.J., Bobrow, D.G., Lanning, S.M., Tatar, D.G. and Foster, G.S., Xerox Corporation, Small-scale workspace representations indicating activities by other users. U.S. Patent 4,974,173, 1990.

Strohman T, Metzler D, Turtle H, Croft WB, “Indri: A language model-based search engine for complex queries”, In Proceedings of the International Conference on Intelligent Analysis, vol. 2, pp. 2-6, 2005.

Stuntz, William J., “Lawyers, deception, and evidence gathering”, In Virginia Law Review, pp. 1903-1956, 1993.

Sun.Mingyu. and Chai.Y.Joyce., “Discourse processing for context question answering based on linguistic knowledge”,In Knowledge-Based Systems, pp. 511-526, 2007.

Swan, Shanna H., Eric P. Elkin, and Laura Fenster., “The question of declining sperm density revisited: an analysis of 101 studies published”, In Environmental health perspective, pp 961-972, 2000.

Taskar, B., Simon Lacoste-Julien, and Dan Klein., “A discriminative matching approach to word alignment”, In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 73-80, 2005.

Tellez-Valero.Alberto.,Gomez.Montes-y.Manuel, Pineda.Villasenor.Luis,Penas.Anselmo., “Towards multi-stream question answering using answer validation”, In Informatica (Slovenia), pp. 190-211, 2010.

Toutanova, Kristina, H. Tolga Ilhan, and Christopher D. Manning., “Extensions to HMM-based statistical word alignment models”, In Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol. 10, pp. 87-94, 2002.

Vasin P., Dan Roth, and Wen-tau Yih. “Mapping dependencies trees: An application to question answering”, In Proceedings of AIMath, pp. 1-10, 2004.

Voorhees, E.M., “The TREC-8 Question Answering Track Report”, In TREC, vol. 99, pp. 77-82, 1999.

Voorhees, Ellen M., “The TREC-8 Question Answering Track Report”, In TREC, vol. 99, pp. 77-82, 1999.

Waltz, D.L., “An English language question answering system for a large relational database”, Communications of the ACM, pp. 526-539, 1978.

Wang, Y. and Hu, J., “A machine learning based approach for table detection on the web”, In Proceedings of the 11th international conference on World Wide Web, ACM, pp. 242-250, 2002.

Wang, Z., Yan, S., Wang, H. and Huang, X., “An overview of Microsoft deep QA system on Stanford WebQuestions benchmark”, Technical report, Microsoft Research, 2014.

Wei Xing, Croft Bruce. W, Mccallum Andrew, “Table extraction for answer retrieval”,In Information Retrieval, pp. 589-611, vol 9, 2006.

Wener, R.. “Effectiveness of the Direct Supervision System of Correctional Design and Management A Review of the Literature. Criminal Justice and Behavior, pp. 392-410, 2006.

Wilensky, R.L., Yudelman, P., Cohen, A.I., Fletcher, R.D., Atkinson, J., Virmani, R. and Roberts, W.C., “Serial electrocardiographic changes in idiopathic dilated cardiomyopathy confirmed at necropsy”, The American journal of cardiology, pp. 276-283, 1988.

Yang L., Qun Liu, and Shouxun Lin., “Log-linear models for word alignment”, In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 459-466, 2005.

Yang, H., Chua, T.S., Wang, S. and Koh, C.K., “Structured use of external knowledge for event-based open domain question answering”, In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, pp. 33-40, 2003.

Yao X., Durme V. B., “Information Extraction over Structured Data: Question Answering with Freebase”, In Proceedings of ACL, Baltimore, MD, USA, pp. 203-223, 2014.

Yao X., Durme V. B., Clark P., “Automatic Coupling of Answer Extraction and Information Retrieval”, In Proceedings of ACL short, Sofia, Bulgaria, pp. 109-116, 2013.

Yao X., Durme V. B., Clark P., and Callison-Burch C., “Answer Extraction as Sequence Tagging with Tree Edit Distance”, In Proceedings of NAACL, pp. 167-188, 2013.

Yao, X. and Van Durme, B., “Information Extraction over Structured Data: Question Answering with Freebase”, In ACL, pp. 956-966, 2014.

Yao, X., Van Durme, B., Callison-Burch, C. and Clark, P., August. A Lightweight and High Performance Monolingual Word Aligner, In ACL, pp. 702-707, 2013.

Yih Wen-tau, Toutanova K., Platt J. C., and Christopher M., “Learning discriminative projections for text similarity measures”, In Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 247-256, 2011.