# Improve and Implement an Open Source Question Answering System

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Salil Shenoy

December 2017

The Designated Project Committee Approves the Project Titled

Improve and Implement an Open Source Question Answering System

by

Salil Shenoy

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

December 2017

| | |
|---|---|
| Dr. Chris Pollett | Department of Computer Science |
| Dr. Mark Stamp | Department of Computer Science |
| Dr. Robert Chun | Department of Computer Science |

# ABSTRACT

Improve and Implement an Open Source Question Answering System

by Salil Shenoy

A question answer system takes queries from the user in natural language and returns a short concise answer which best fits the response to the question. This report discusses the integration and implementation of question answer systems for English and Hindi as part of the open source search engine Yioop. We have implemented a question answer system for English and Hindi, keeping in mind users who use these languages as their primary language. The user should be able to query a set of documents and should get the answers in the same language. English and Hindi are very different when it comes to language structure, characters etc. We have implemented the Question Answer System so that it supports localization and improved Part of Speech tagging performance by storing the lexicon in the database instead of a file based lexicon. We have implemented a brill tagger variant for Part of Speech tagging of Hindi phrases and grammar rules for triplet extraction. We also improve Yioop's lexical data handling support by allowing the user to add named entities. Our improvements to Yioop were then evaluated by comparing the retrieved answers against a dataset of answers known to be true. The test data for the question answering system included creating 2 indexes, 1 each for English and Hindi. These were created by configuring Yioop to crawl 200,000 wikipedia pages for each crawl. The crawls were configured to be domain specific so that English index consists of pages restricted to English text and Hindi index is restricted to pages with Hindi text. We then used a set of 50 questions on the English and Hindi systems. We recored, Hindi system to have an accuracy of about 55% for simple factoid questions and English question answer system to have an accuracy of 63%.

# ACKNOWLEDGMENTS

I would like to thank everyone who helped me towards completing my project.

I would like to thank my advisor, Dr. Christopher Pollett, for his patience and constant guidance towards the project, without which the success in this project would not have been achievable.

I would like to extend my thanks to my committee members, Dr. Mark Stamp and Dr. Robert Chun, for their suggestions given towards the project.

# TABLE OF CONTENTS

CHAPTER

# LIST OF FIGURES

# CHAPTER 1

## Introduction

The web has information written in almost 7000 languages and it is important to have systems which can retrieve information effectively in different languages. The most researched Indian language is Hindi which means a relatively large number of Hindi documents are available. The purpose of implementing a question answer system which can accept and retrieve answers in Hindi is to increase the access of Hindi speakers to more advanced information retrieval software. The users prefer and feel a sense of comfort if they can use a software, in a language they have expertise in. This is where Natural Language Processing [1] plays an important role as it helps to handle information in native languages. I have implemented a Hindi question answer system for Yioop which allows user to ask a question in Hindi and returns the answer also in Hindi. I developed a part of speech tagger and a triplet extractor to process Hindi text extracted by Yioop.

There are various Question Answering systems implemented over the years which extract data from the web and return answers. The systems implemented may use a knowledge store, machine learning or a combination of the two to process the data and extract question answers. We describe some of the related work and existing systems for Question Answering systems.

Question answer systems were developed with a view of extending research in natural language processing. One of the first question answer systems developed was STUDENT [2]. This system was capable of solving high school algebra problems. Another example was LUNAR [3] which was a system developed to answer questions related to moon rock data. LUNAR answer questions with an accuracy of 78%. Hindi

Language Interface to Database (HLIDB) [4] is a knowledge based question answer system which takes input for user in Hindi and converts it to SQL and retrieves answers from the database.

In order to understand how effective a question answer system is, one needs to establish a quantifiable means of testing question answers. One early such approach was taken by the developers of START, Question Answer System [6]. START extracts data from the web to answer questions ranging from cities to people etc. The system uses a database to fetch the answer to a user asked question. START system was tested using some sample questions related to various domain, and most of them were answered. Results were precise and few of them were supported with images. Garima Nanda et al. implemented a Hindi Question Answer system, WebBasedQA [7] using a combination of machine learning and a knowledge base to answer user queries. The system parses input, tokenizes and extract features using previously calculated results. The system uses Naive Bayes as a classifier combined with known data store to return answers in Hindi. The system was tested with two sets of questions, one set included questions asked by a user aware of the domain and another included questions asked by a user who was unaware about the domain. The results were 92% and 88% respectively. Information retrieval system for laws (IRSL)[8] is a closed domain Question Answer system developed for helping users get answers related to the Indian penal code. The system uses OpenNLP to preprocess the input and a Q-Learning algorithm to learn from user inputs and a Wordnet developed at Princeton University to extract synonyms. The system uses a set of words which identify the theme and a set of indexed keywords which are words in the laws. The system returns results based on the number of main and indexed keywords in the user query. The system improves evolves based on feedback from the user. The system was evaluated using precision score based on number of indexed keywords found in the user query. The accuracy of the system was found to be 79.19%.

The Question Answer Systems are mainly classified as Open Domain and Closed Domain Systems. Open domain systems are responsible for handling large amounts of data and wide range of questions. They are designed to answer questions about everything. Google, Wikipedia, etc. are examples of such systems. They depend on their knowledge store to tackle the questions and return the best answers. Closed domain systems are designed to handle questions in a specific domain. For example, IRSL system is designed specifically to handle question-answers related to the Indian Judiciary. Such systems have a fixed set of documents which they process when a user asks a query to return the best answer.

The current work on Hindi information retrieval systems is limited to closed domain systems. These systems cannot be used by people from outside that domain. I have added an open domain Hindi Question Answer system to Yioop which answers simple questions asked in Hindi by retrieving answers also in Hindi, from the internet.

Natural Language Processing is the core of any Question Answer system. The number of modules involved in building an efficient Question Answer system adds to the complexity of such a system. As part of the project, we have improved the performance of an existing Question Answering Module [10] in Yioop. The work includes implementing a similar Question Answer system for Hindi which is an Indian language. In this report, we describe the the different steps we followed to build the system. The report is organized as follows: Chapter 2 gives a background on Question Answer Systems and describes the different approaches used in developing such systems, Chapter 3, describes the individual modules like part of speech tagger, triplet extractor which were implemented to build the Hindi Question Answer System, and also a feature which allows users to add named entities to the Yioop database, Chapter 4, describes tests and experiments conducted on the Hindi Question Answer System, and a comparison of English and Hindi Question Answer systems in Yioop, Chapter 5 we give a short summary of the work done.

# CHAPTER 2

## Background on Question Answer Systems

Even with the simplifications developed for searching information online it can be tough and time consuming to navigate through the vast amount of data. One of the solutions to this problem is developing an automated system which is capable enough to accept input in natural language and generate an output which is equally natural. This is where a Question Answer System plays an important role. The aim of a Question Answer System is basically to allow a user to ask a question in everyday language and receive an answer in user comprehensible format.

As with developing software systems, building an efficient and robust Question Answering System has its own fair share of challenges. There are multiple ways in which we can build this system from machine language translation [11], neural networks [12] and different machine learning algorithms [13].

The first challenge faced while developing such a system is how do we collect the data, how do we store it. The next challenge which comes to mind are the users of this system. A system developed for the internet needs to effective enough to process the data in a way so as to not restrict the users, in other words, the system needs to handle the different ways which users can express themselves. Then comes the most challenging aspect of this system, the language itself. We have about 7000 languages spoken worldwide, getting data of sufficient quality so that we develop such a system is a difficult task as except a few, most of the languages have minimal resources on the world wide web.

Below we explain in brief the architecture of a Question Answer System and discuss the different approaches which are used to extract information while implementing a

Question Answer System. In general, any such system will always have the following modules,

**Information Retreival** The Information Retrieval module is responsible for extracting data from different sources, a collection of documents, text, transcripts or a relational database

**Data Processing** The system needs to retrieve the data and extract information from it, this involves multiple phases from extracting text data like summaries, then performing part of speech tagging, using some form of term chunking to recognize named entities like persons, organizations, or locations, and eventually generating some form of question answer pairs.

**Answer Processing** Some systems give the best answer they find on processing the data from what information is gathered others using some form of scoring and ranking to give out the best answer to the user.

## 2.1 Question Answer System Paradigms

We have had multiple Question Answer Systems developed till today [6] [14]. A question answer system can be implemented by following any of the three paradigms [15] which are IR-based Question Answer Systems, Knowledge based Question Answer Systems and Hybrid Question Answer Systems.

### 2.1.1 IR-based Question Answering System

A simple Question Answer system should be capable of providing an answer which is short, concise and as close as possible to the correct answer. A Question Answer system which answers facts, generally returns short strings. These strings are mostly named entities viz. a person, an organization or a location. Such a system is known as a Factoid Question Answering System. A factoid question answering system searches for answers on the Web, or a document collection or short transcripts from which it can retrieve the possible answers [16]. These are then formatted and presented to the user. These systems generally involve three steps, Question Processing, Retrieving and Ranking Passages and Extracting the Answers. Figure 1 describes the architecture for an IR-Based question answer system.

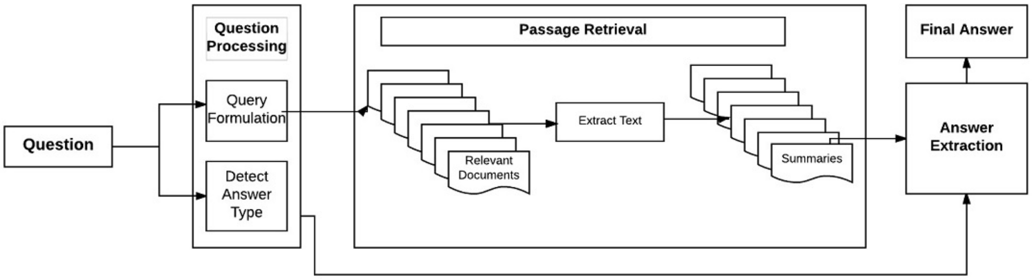


Figure 1: IR Based Question Answer System.

**Question Processing** This phase decides what type of question is being asked and subsequently which type of answer to extract. For question processing to work we need to perform the query formulation. Query formulation converts the posed question into a form which can be used in Information Retrieval. Once the form of the question is identified the exact answer type can be retrieved.

**Retrieving and Ranking Passage** Once we get a query format from Step 1, we can use this to search the answer in the documents. In this step, we first rank the documents in which we find probable answers. For the documents which do not contain a match, we use the user written rules or machine learning algorithms to rank them.

**Extracting Answers** The system extracts answers from passages using one of the two methods, either by using answer type or by using N-grams tiling.

### 2.1.2 Knowledge based Question Answer System

This paradigm relies on the mechanism to query a database. A semantic query is formed for the question which is asked. The query formed is used to retrieve the result from the database. The system ideally functions like a semantic parser as it maps a text string to a logical format. The database can be a relational database or the system may store triplets. The triplet has a predicate which defines the relation between the other 2 (two) parts. For example, DBPedia [17], Freebase [18] are triplet stores derived from Wikipedia Infoboxes. One of the questions which can be posed to Knowledge based Question Answer system is to ask about one of the missing factors in the triplet. Figure 2 describes the architecture of a Knowledge BAsed Question Answer system and we then different approaches followed when developing such a system.

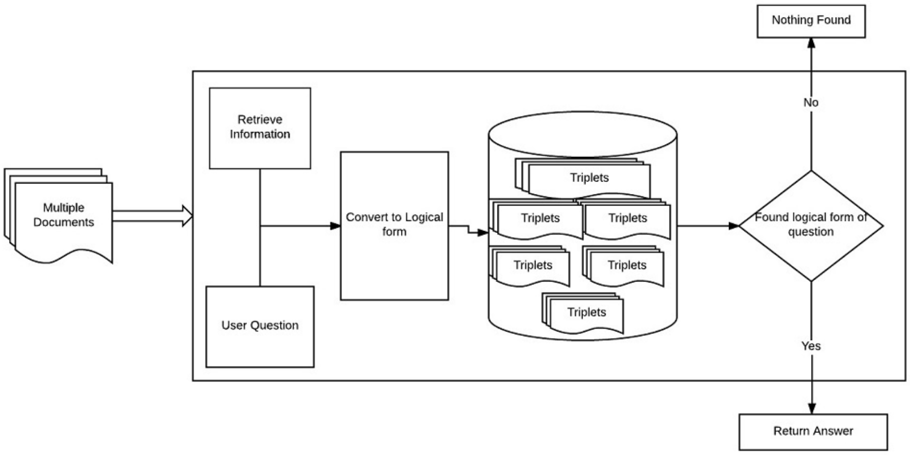


Figure 2: Knowledge Based Question Answer System.

**Rule Based Approach** This approach involves implementing hand written rules which will be used to extract the missing element from the triplet.

**Supervised Approach** In this approach, we have a training data consisting of mappings of questions to their logical forms. The model is trained using this data, so that going ahead it can identify which mapping to use in the future.

**Semi Supervised Approach** With the ruled based approach one needs to know the language for which the system is being implemented which cannot be the case always. For the supervised approach the most challenging aspect is having quality and correct training data, which always never happens. To overcome the drawbacks of these approaches we have a Semi supervised approach. One system known as REVERB extracts information from triplet stores and other sources like Wikipedia to create new relations while paralelly undergoing training to map between questions and the logical forms.

### 2.1.3 Hybrid Approach to Question Answering System

The previous approaches were limited to using text or knowledge for the Question Answer System. In the hybrid approach we combine the steps in these to implement the system. One example of such a system is IBM Watson [19]. Figure 3 describes the different phases involved in the implementing a hybrid question answer system.

**Question Processing** In this phase, the system parses the question, tags it for named entities and extracts possible relations and as in the IR based approach it detects the answer type and question type.

**Candidate Answer Generation** Once we have the query phase from Step 1, we search external documents, texts and transcripts as well as a structured database to

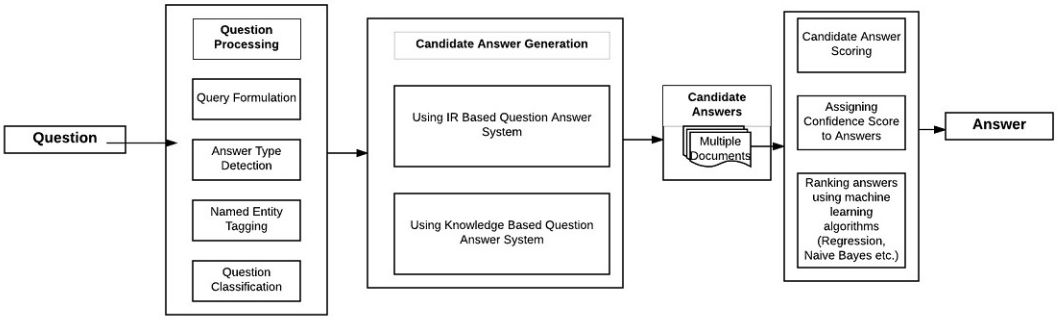

Question
Question Processing
Query Formulation
Answer Type Detection
Named Entity Tagging
Question Classification
Candidate Answer Generation
Using IR Based Question Answer System
Using Knowledge Based Question Answer System
Candidate Answers
Multiple Documents
Candidate Answer Scoring
Assigning Confidence Score to Answers
Ranking answers using machine learning algorithms (Regression, Naive Bayes etc.)
Answer

Figure 3: Hybrid Question Answer System.

extract as many as possible candidate answers. The manner in which we search the query phrase will differ, for the searching the external sources, the methods depend on the text we are searching. For the database, we can use queries similar to the ones we use with triplet stores like FreeBase, DBPedia, etc.

**Answer merging and scoring** This involves merging the extracted answers which are similar. For example, the United States of America and U.S.A would be merged, this needs a dictionary with similar entities which can help detect this type of conflicts. At the end, we have a set of answers each with a feature vector. These are then subjected to a classifier and assigned a confidence value. This step runs iteratively helping output the best answer to the user.

Yioop is a Knowledge based question answer. The user asked questions are answered using the triplets stored as part of the Yioop index. Yioop extracts summaries from webpages which are then subjected to part of speech tagging and triplet extraction. These triplets are then stored in the database. When a user asks a question, the question undergoes part of speech tagging and a triplet is generated. These triplet is looked up in the Yioop database to retrieve the answer. Figure 4 describes the architecture of Question Answer system in Yioop. In case the triplet is not found, Yioop returns a list of links and references related to the question. I have implemented a Knowledge based Question Answer for Hindi, in which the user asks a question in Hindi and Yioop returns a answer also in Hindi.

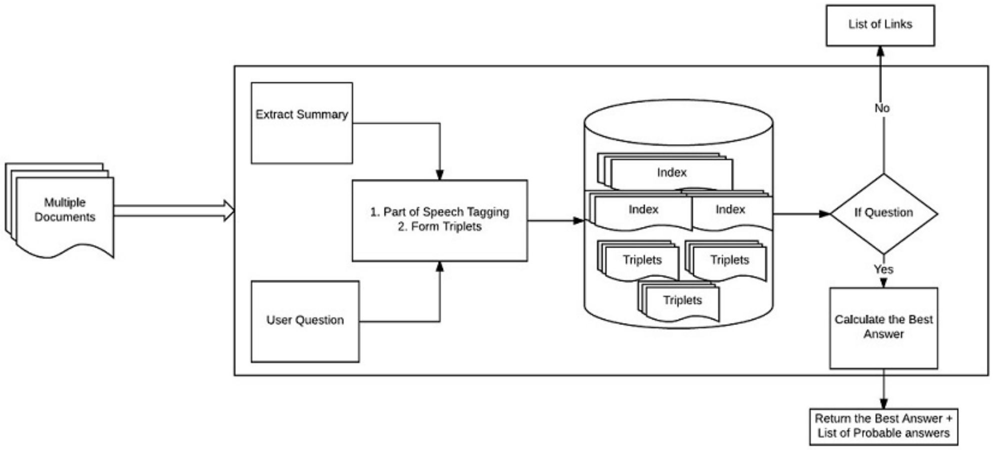


Figure 4: Question Answer System in Yioop.

# CHAPTER 3

## Implementation of Hindi Question Answer System

This chapter discusses about the different stages involved in the functioning of the Question Answer System in Yioop. The inputs to the system are the summaries of the web pages Yioop has crawled. For any given page, the summary is processed by removing punctuation, special characters, etc. After this initial processing, Yioop extracts phrases from the summary which are then subjected to the following modules.

1. Part of Speech Tagger.

2. Triplet Extraction i.e Subject, Object, Verb extraction.

3. Named Entity Addition to Database

### 3.1 Part of Speech Tagger

A part of speech tagger plays an important role in natural language translation. Part of speech tagging is also known as word category distribution or grammatical tagging. Tags are basically parts of speech like noun, adjective, verb, etc. A robust part of speech tagger can not only retrieve information more efficiently but can also help in understanding what the text actually means. Even with all the advances in machine learning automatic tagging of words is daunting as many times even hand tagging sentences is difficult purely because of the ambiguous nature of languages. Having said that there are systems designed to perform this task. One such Part of Speech tagger is the Brill tagger [20] which is nothing but a Rule Based Approach to tagging a sentence. Below are the approaches for part of speech tagging,

**Rule Based Approach** This is one of the earliest approaches followed to implement tagging. In this approach, we have a set of well-defined rules which are applied to a sentence. The approach uses labelled data which is a lexicon or a dictionary of words and its probable parts of speech. The rules are then applied to define the most accurate tag for that word. Ambiguity between multiple tags is sorted by using the tags of words which precede the particular word.

**Machine Learning Based Approach** Today we have multiple machine learning algorithms which are used for Part of Speech tagging [21]. Algorithms like Hidden Markov Model, Neural Networks, etc. are trained on a small sample of text and over time these models prove to be almost as effective in retrieving information as a human would do. The most known part of speech tagging algorithms apart from Brill Tagging are the Viterbi Algorithm.

We have an implementation of a Brill tagger for English and a similar variant for Hindi [22], [23]. The initial implementation of the Question Answer System had a file based lexicon which meant tagging a word required a file scan, which eventually slowed the Question Answer module in Yioop as a whole. As a solution, the lexicon is now stored as part of the database which will be used by Yioop. The lexicon table created is indexed on the word and locale meaning the retrieval time is reduced to $O(1)$ greatly improving the speed of the Question Answer module.

For the Hindi variant of the Brill tagger, we first tag the words as per data available in the database. Then for the remaining words of the sentence we use the rules to assign the most probable tag. On the next page, we describe the algorithm and the rules used to tag words in a Hindi Sentence.

**Use the Lexicon in the database** First tag the words in the phrase which are present in the Lexicon. The words which are not present in the lexicon are tagged as Unknown. We then apply the below rules to tag the words.

**Rule to identify Noun** Rule1: If the previous word tagged is an Adjective / Pronoun / Postposition then the current word is likely to be a noun
Rule 2: If the current word is a verb then the previous word is a noun
Rule 3: If the current tag is a noun then next / previous is a noun

**Rule to identify demonstrative nouns** Rule 1: If the current and previous words are tagged as pronouns then the previous word is a demonstrative
Rule 2: If the current word is a noun and the previous word is a pronoun then the current word is demonstrative

**Rule to identify pronoun** Rule: If the previous word is unknown and the current word is a noun then the previous word is a pronoun

**Rule to identify Noun** Rule: If we get two words which are untagged the most probably they form a name and will be tagged as noun

**Rule to identify Adjective** Rule: If the word ends with <tar>, <tam>, <thik> then we tag it as a Adjective

**Rule to identify verbs** Rule: If the current word is tagged as Auxilary verb and the previous word is tagged as Unknown then the previous word is a verb

**No rule matched** Rule: After applying all the rules if the word is still tagged as Unknown then tag it as a Noun.

**Algorithm 1** Part of Speech Tagger

```
procedure TAGPARTOFSPEECH($sentence$)
    terms ← sentence split on space
    result ← []
    i ← 0
    for term in terms do
        term['tag'] ← unknown
        partofspeech ← select partofspeech from database where word ← term
        if partofspeech exists then
            term['tag'] ← partofSpeech
        result[i++] = term
    result = tagUnknowWords(result)
    return result
procedure TAGUNKNOWNWORDS($partiallytaggedsentence$)
    for term tagged as Unknown do
        if $previous['tag'] \leftarrow Adjective$ OR $previous['tag'] \leftarrow Pronoun$ OR $previous['tag'] \leftarrow Participle$ then
            current['tag] ← noun
            result ← current
        if $current['tag'] \leftarrow verb$ then
            previous['tag] ← noun
            result ← previous
```

**Algorithm 2** Part of Speech Tagger

```
        if $previous['tag'] \leftarrow unknown$ AND $current['tag'] \leftarrow noun$ then
            previous['tag] ← pronoun
            result ← previous
        if $current['tag'] \leftarrow aux.verb$ OR $previous['tag'] \leftarrow unknown$ then
            previous['tag] ← verb
            result ← previous
        if current['token'] ends with 'eek' OR current['token'] ends with 'tar' OR current['token'] ends with 'tam' then
            current['tag'] ← adjective
            result ← current
        if $current['tag'] \leftarrow unknown$ then
            current['tag] ← noun
            result ← current
    return result
```

### 3.2 Parse Tree Generation

After all the words in the sentence are tagged with the most probable part of speech, we generate a tree like structure composing of the Noun, Verb and PostPosition or Preposition Phrase. For a Hindi sentence [24], the structure is generally Noun Phrase followed by the Verb Phrase, which is same as most of the English sentences. In English, the verb or the predicate helps us separate the Noun and Verb Phrase, in case of Hindi we have something similar, these are known as *case* words. These help us distinguish between the subject and object in a given sentence. This is the reason we have implemented the Hindi parse tree to be comprised of Noun Phrase, Postposition Phrase and Verb Phrase.

**NP:** {⟨ **JJ** | **NN*** ⟩}+

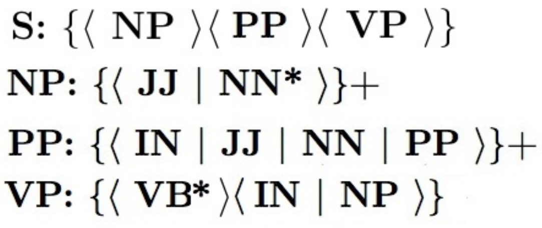
For any given sentence, the initial sequence of adjectives and nouns becomes part of the noun phrase. The next step is to extract the post positional or the prepositional phrase following the noun phrase this will form the object of the sentence,

**PP:** {⟨ **IN** | **JJ** | **NN** | **PP** ⟩}+

This rule extracts the information from the sentence till we encounter the verb phrase. Hindi being a subject-object-verb language, most of the sentences usually end with verbs. So the verb phrase extraction rule is

**VP:** {⟨ **VB***⟩⟨ **IN** | **NP** ⟩}

In Hindi, the case words which separate the noun and post position phrase are the ones which help define the relationship between the subject and object. Hence, identifying the case words more accurately will help increase how accurate the generated parse tree will be. Below we describe the algorithms for parsing the sentence for different parts like the Noun Phrase, Postpositional Phrase and Verb Phrase,



**S:** $\{\langle$ **NP** $\rangle\langle$ **PP** $\rangle\langle$ **VP** $\rangle\}$

**NP:** $\{\langle$ **JJ** | **NN*** $\rangle\}+$

**PP:** $\{\langle$ **IN** | **JJ** | **NN** | **PP** $\rangle\}+$

**VP:** $\{\langle$ **VB*** $\rangle\langle$ **IN** | **NP** $\rangle\}$

Figure 5: Grammar Rules.

**Algorithm 3** Extract Noun Phrase

```
procedure EXTRACTNOUNPHRASE($taggedPhrase, parseTree$)
    curNode ← parseTree[node]
    adjectiveTree ← extractAdjective from the tagged phrase from currentNode
    currentNode ← just after the node last processed by extractAdjective
    nounTree ← extractNouns from the tagged phrase from currentNode
    tree ← adjectiveTree ∪ nounTree
    return tree
```

**Algorithm 4** Extract Post Positional Phrase

```
procedure EXTRACTPOSTPOSITIONPHRASE($taggedPhrase, parseTree$)
    currentNode ← parseTree[node]
    if $current['tag']$ is post position then
        postPositionTree ← extractpostposition from the tagged phrase from currentNode
        currentNode ← just after the node last processed by extractpostposition
        adjectiveTree ← extractAdjective from the tagged phrase from currentNode
        currentNode ← just after the node last processed by extractAdjective
        nounTree ← extractNouns from the tagged phrase from currentNode
        extractPostpositionPhrase with updated currentNode extract information until verb is encountered
    return tree
```

**Algorithm 5** Extract Verb Phrase

```
procedure EXTRACTVERBPHRASE($taggedPhrase, parseTree$)
    currentNode ← parseTree[node]
    verbTree ← extractVerbs from the tagged phrase from currentNode
    postPositionTree ← extractpostposition from the tagged phrase from currentNode
    nounphrasetree ← call extractNounPhrase
    return tree
```

### 3.3 Triplet Extraction

Phrases in languages like English and Hindi are mainly composed of three parts viz. the Subject, Object and Verb. The triplet extraction module is responsible for retrieving these parts from a phrase, which are then rearranged to form multiple question answer pairs.

After a given sentence has all the words tagged with its most probable part of speech we generate a tree like structure known as the parse tree which is made of the Noun Phrase and Verb Phrase. In case of Hindi, we have three parts to the parse tree the Noun Phrase, PostPosition Phrase and the Verb Phrase. We do this because English being a Subject-Verb-Object language, the triplet extractor is able to extract the subject, verb, object triplet from the Noun and Verb Phrase. Incase of Hindi, which is a Subject-Object-Verb language, we use the PostPosition Phrase to help us distinguish between the Subject and the Object in a given sentence. We describe the triplet extraction algorithm which takes as input the parse tree and returns a triplet.

**Algorithm 6** Triplet Extraction

```
procedure TRIPLETEXTRACTION($parseTree$)
    triplet ← extractSubject(parseTree) ∪ extractObject(parseTree) ∪ extractVerb(parseTree)
    return triplet
```

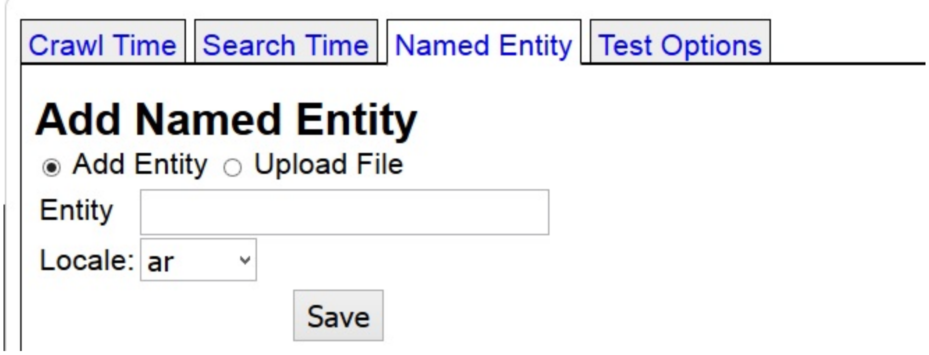
**Subject Extraction** We extract the subject from the parse tree by recursively parsing the noun phrase for the first level noun, this constitutes the subject for a CONCISE triplet, we then continue to parse the entire noun phrase from the parse tree to form the subject for the RAW version of the triplet. The RAW version of the triplet consists of words tagged as JJ, NN, NNP, NNP, NNPS. The CONCISE form of the triplet consists only of the first level noun this results in lose of some information from the text. On the other hand, the RAW triplet consists of the adjectives in the noun phrase which helps preserve the information.

**Object Extraction** The next step is extracting the object from the parse tree. We achieve this by parsing the postposition phrase in the parse tree. The object constitutes to the remaining nouns in the sentence. For a hindi sentence, the object is responsible for providing the answers to most of the wh questions like who, where, when etc. For example, a question in English like "Who is Barack Obama" becomes "Barack Obama kaun hai", here the word 'kaun' represents the 'Who' in the question, the answer to which is stored by the object. Another example, "When is the new year" becomes "Naya saal kab hai" here "kab" represents the "when" in the question and is answered to by the corresponding object stored as part of the triplet.

**Predicate Extraction** The predicate is extracted from the Verb phrase of the parse tree. These are terms tagged with VB, VBZ, VBG, VBP. As compared to a English sentence, the verb phrase has minimal information related to the text. It helps define the tense and relation between the information extracted from subject and object.

### 3.4 Named Entity Addition

The performance of the Question Answer system depends on the ability to extract information from the text. As we know most of the answers are facts (named entities) for such a system, the accuracy of the system is directly proportional to its ability to recognize them from the text. Most of the existing systems use a hand written dictionary of named entities to enhance the results. Following a similar approach, we add an option under the Page Options section in Yioop for the user to add entities to the Lexicon table. The user can do this by adding a single entity at a time as shown below in Figure 6



Add Named Entity

● Add Entity ○ Upload File

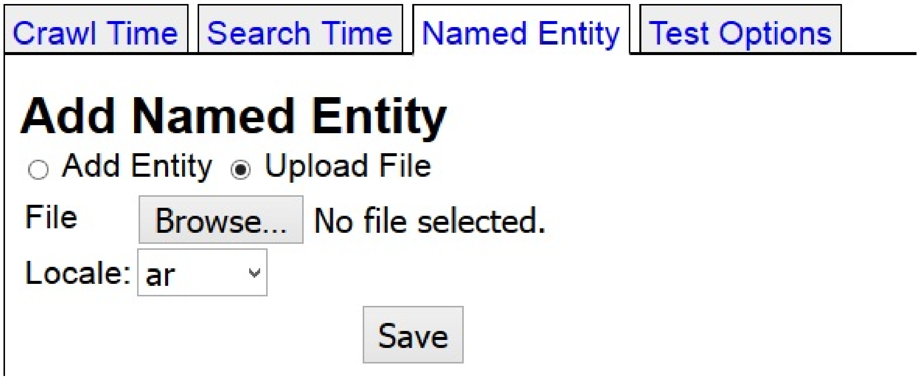
Entity

Locale: ar

Figure 6: Add Named Entity.

Or they can upload a text file to Yioop which contains line separated entities and select the locale for the which the file contains the entities as shown in Figure 7

For any given language user can view all the entities already added to the database, they can edit or delete entities of their choice. The UI for this functionality is as shown in Figure 8



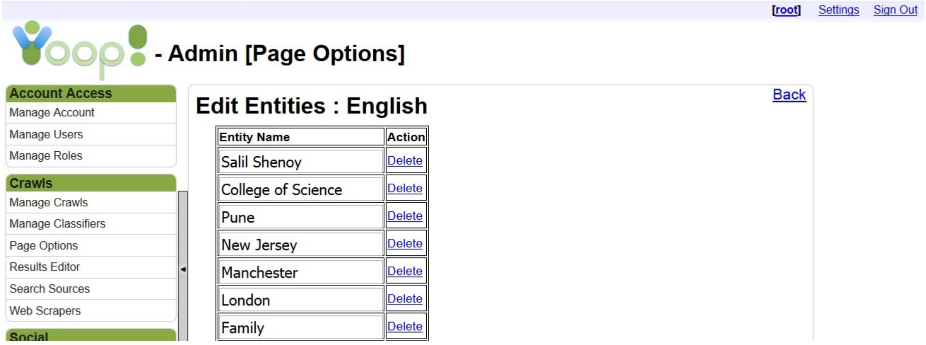
Add Named Entity

○ Add Entity ● Upload File

File: No file selected.

Locale: ar

Figure 7: File Upload to add Named Entities.



Edit Entities : English

| Entity Name | Action |
|---|---|
| Salil Shenoy | Delete |
| College of Science | Delete |
| Pune | Delete |
| New Jersey | Delete |
| Manchester | Delete |
| London | Delete |
| Family | Delete |

Figure 8: View all Entities.

# CHAPTER 4

## Tests and Results

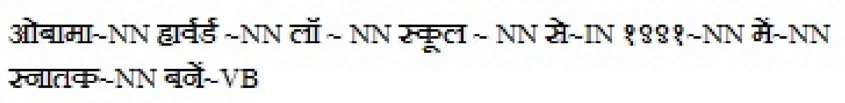
A general approach to evaluating question answer tasks is using the mean reciprocal rank (MRR). The score for an individual question is the reciprocal of the rank at which the first correct answer is returned or 0 if no correct response is returned. The score for the run is then the mean over the set of questions in the test. The number of questions for which no correct response is returned is also reported. We use. a similar approach for our evaluation of the question answer systems in Yioop. The Question Answer System implemented is part of the Yioop search engine and is platform independent. The system works by tagging phrases using a Brill variant Part of Speech tagger for Hindi sentences. In the next step, triplets are formed and stored in the database using the grammar rules for Hindi. The test data for the system is an index created by configuring Yioop to crawl Hindi webpages from Wikipedia and Indian websites with Hindi content. We describe the experiments conducted on the Question Answer Module as a standalone utility and next we describe the results when integrated in Yioop.

### 4.1 Question Answer Module: Standalone Testcase

In this section, we describe test cases for the system as standalone module. It is assumed that the input to the system is a processed to remove special characters, punctuations, etc. Also, the given sentence is semantically and syntactically correct.

Figure 9 shows the sentence after it is tagged for parts of speech

A word for word translation of the above sentence to English is 'Obama Harvard law school from 1999 graduate complete'. Figure 10 shows the Parse Tree generated for this sentence



ओबामा~NN हार्वर्ड ~NN लॉ ~ NN स्कूल ~ NN से~IN १९९१~NN में~NN स्नातक~NN बनें~VB

Figure 9: Hindi Sentence 1.

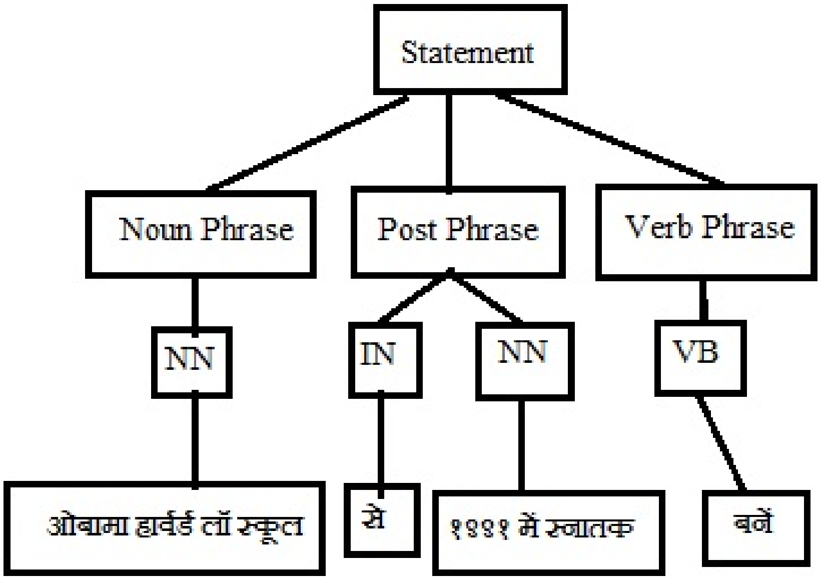
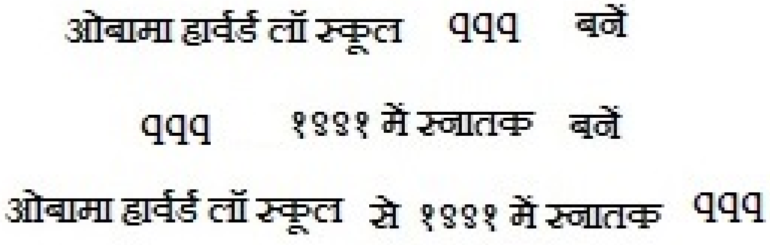


Figure 10: Parse Tree for sentence in Figure 7.

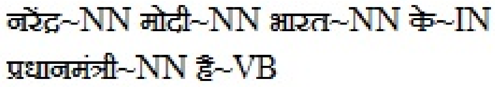
The triplets extracted for the above parse tree are as shown in Figure 11.



| | | |
|---|---|---|
| ओबामा हार्वर्ड लॉ स्कूल | ९९९ | बनें |
| ९९९ | १९९१ में स्नातक | बनें |
| ओबामा हार्वर्ड लॉ स्कूल | से १९९१ में स्नातक | ९९९ |

Figure 11: Triplets Extracted for sentence Figure 8.

Figure 12 shows the sentence after it is tagged for parts of speech



नरेंद्र~NN मोदी~NN भारत~NN के~IN
प्रधानमंत्री~NN है~VB

Figure 12: Hindi Sentence 2.

A word for word translation of the above sentence to English is 'Narendra Modi India (s) Prime Minister is'. Figure 13 shows the Parse Tree generated for this sentence

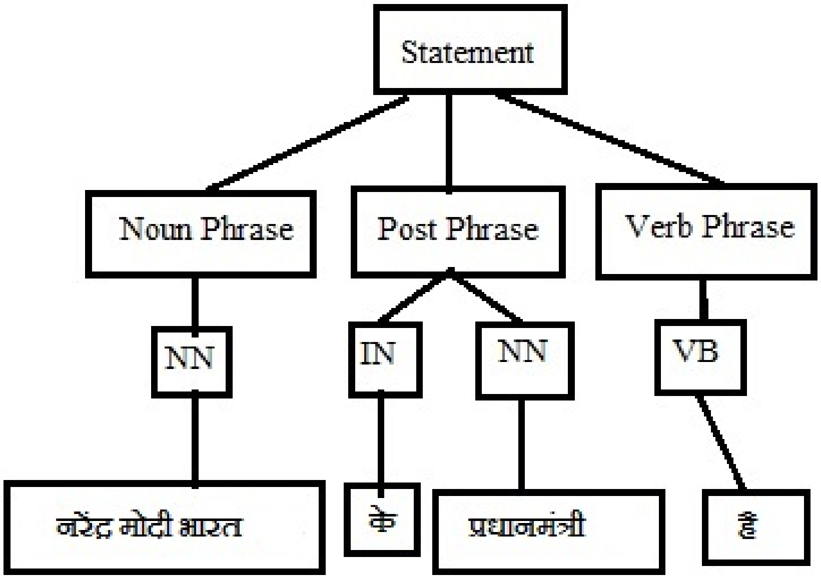
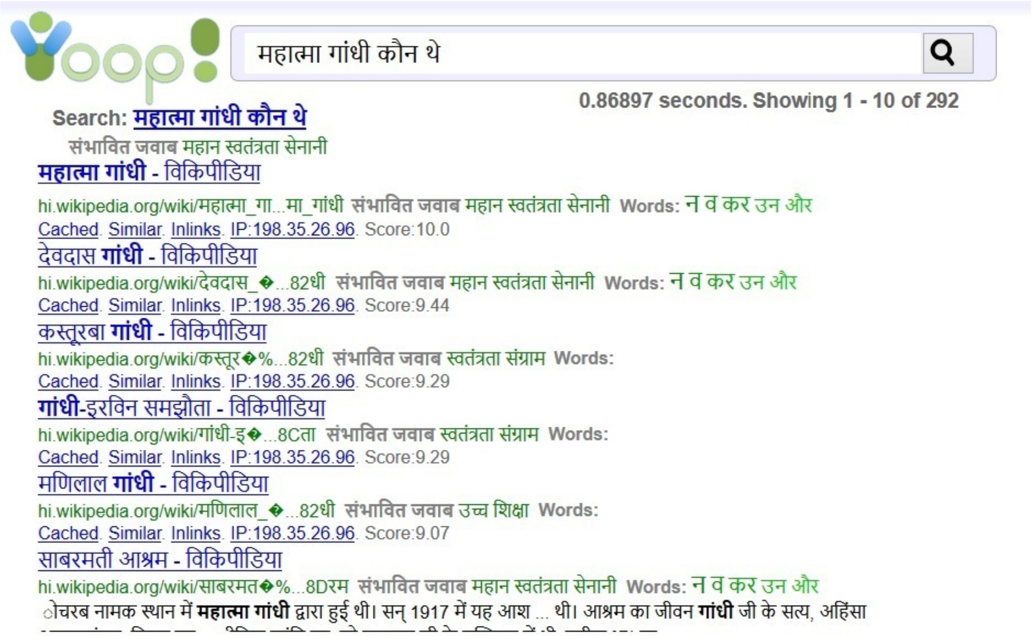


Figure 13: Parse Tree for sentence in Figure 10.

## 4.2 Question Answer Module Integrated in Yioop

Below are results for the Question Answer System when a crawl was setup for all wikipedia pages in Hindi, Indian websites. We set up a crawl by configuring Yioop in under crawl options. For the crawl, we restrict the crawler to websites from domains *hi.wikipedia.org*, *co.in* and *in*. We stopped the crawl after we hit 200,000 urls. The crawler extracted information from 7925 webpages to create the index. Figure 14, Figure 15, show the results after the Question Answer system is integrated in Yioop.



0.86897 seconds. Showing 1 - 10 of 292

Search: महात्मा गांधी कौन थे

संभावित जवाब महान स्वतंत्रता सेनानी

महात्मा गांधी - विकिपीडिया

hi.wikipedia.org/wiki/महात्मा_गा...मा_गांधी संभावित जवाब महान स्वतंत्रता सेनानी Words: न व कर उन और

Cached. Similar. Inlinks. IP:198.35.26.96. Score:10.0

देवदास गांधी - विकिपीडिया

hi.wikipedia.org/wiki/देवदास_�...82धी संभावित जवाब महान स्वतंत्रता सेनानी Words: न व कर उन और

Cached. Similar. Inlinks. IP:198.35.26.96. Score:9.44

कस्तूरबा गांधी - विकिपीडिया

hi.wikipedia.org/wiki/कस्तूर�%...82धी संभावित जवाब स्वतंत्रता संग्राम Words:

Cached. Similar. Inlinks. IP:198.35.26.96. Score:9.29

गांधी-इरविन समझौता - विकिपीडिया

hi.wikipedia.org/wiki/गांधी-इ�...8Cता संभावित जवाब स्वतंत्रता संग्राम Words:

Cached. Similar. Inlinks. IP:198.35.26.96. Score:9.29

मणिलाल गांधी - विकिपीडिया

hi.wikipedia.org/wiki/मणिलाल_�...82धी संभावित जवाब उच्च शिक्षा Words:

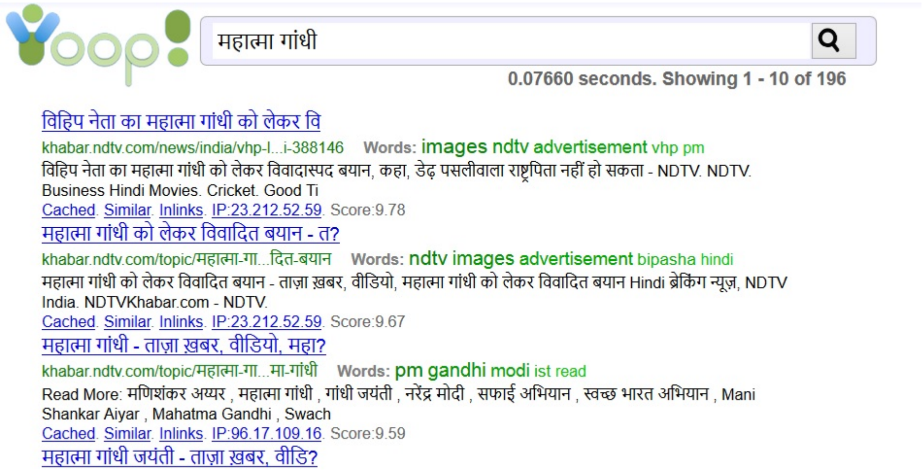
Cached. Similar. Inlinks. IP:198.35.26.96. Score:9.07

साबरमती आश्रम - विकिपीडिया

hi.wikipedia.org/wiki/साबरमत�%...8Dरम संभावित जवाब महान स्वतंत्रता सेनानी Words: न व कर उन और

ोचरब नामक स्थान में **महात्मा गांधी** द्वारा हुई थी। सन् 1917 में यह आश ... थी। आश्रम का जीवन **गांधी** जी के सत्य, अहिंसा

Figure 14: Question Answer Integration in Yioop.



महात्मा गांधी

0.07660 seconds. Showing 1 - 10 of 196

विहिप नेता का महात्मा गांधी को लेकर वि
khabar.ndtv.com/news/india/vhp-l...i-388146 Words: images ndtv advertisement vhp pm
विहिप नेता का महात्मा गांधी को लेकर विवादास्पद बयान, कहा, डेढ़ पसलीवाला राष्ट्रपिता नहीं हो सकता - NDTV. NDTV.
Business Hindi Movies. Cricket. Good Ti
Cached. Similar. Inlinks. IP:23.212.52.59. Score:9.78

महात्मा गांधी को लेकर विवादित बयान - त?
khabar.ndtv.com/topic/महात्मा-गा...दित-बयान Words: ndtv images advertisement bipasha hindi
महात्मा गांधी को लेकर विवादित बयान - ताज़ा ख़बर, वीडियो, महात्मा गांधी को लेकर विवादित बयान Hindi ब्रेकिंग न्यूज़, NDTV
India. NDTVKhabar.com - NDTV.
Cached. Similar. Inlinks. IP:23.212.52.59. Score:9.67

महात्मा गांधी - ताज़ा ख़बर, वीडियो, महा?
khabar.ndtv.com/topic/महात्मा-गा...मा-गांधी Words: pm gandhi modi ist read
Read More: मणिशंकर अय्यर , महात्मा गांधी , गांधी जयंती , नरेंद्र मोदी , सफाई अभियान , स्वच्छ भारत अभियान , Mani
Shankar Aiyar , Mahatma Gandhi , Swach
Cached. Similar. Inlinks. IP:96.17.109.16. Score:9.59

महात्मा गांधी जयंती - ताज़ा ख़बर, वीडि?

Figure 15: No Question Answer System in Yioop.

The integration of the Question Answer system slows down Yioop as extra processing is performed while generating and storing the triplets. But the performance improves for query time as whenever the user enters a question it is looked up directly from a map. Figure 16 shows the time impact when we asked a simple question in Hindi.

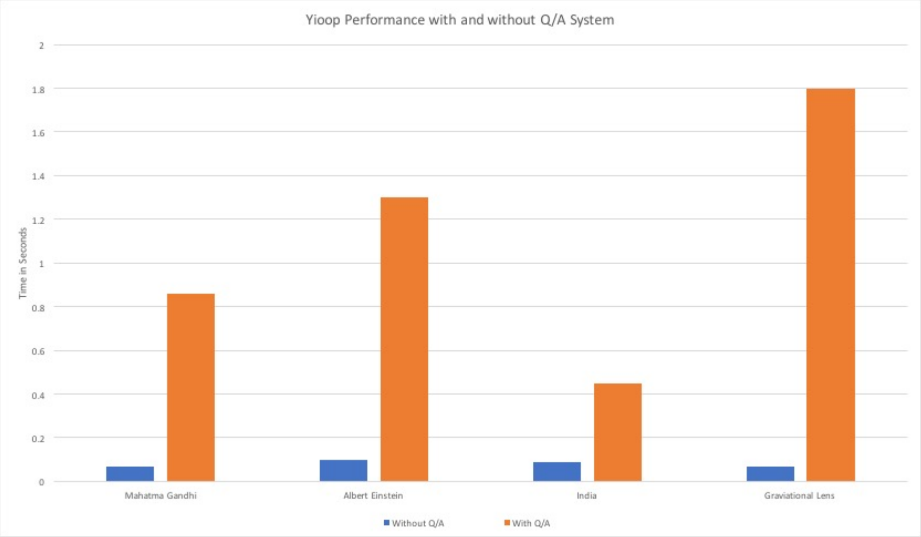


Figure 16: Yioop performance before and after integration of Q/A System.

The initial implementation of the Question Answer system read performed part of speech tagging by reading a file based lexicon from a file. It performed a sequential search on the lexicon read in memory, Figure 17 shows the improvement in part of speech tagging, as words are tagged from the database indexed on term and locale. For the test, I used 1000 - 1500 word paragraphs for each of the subjects as input to the two variants of the part of speech tagger module.

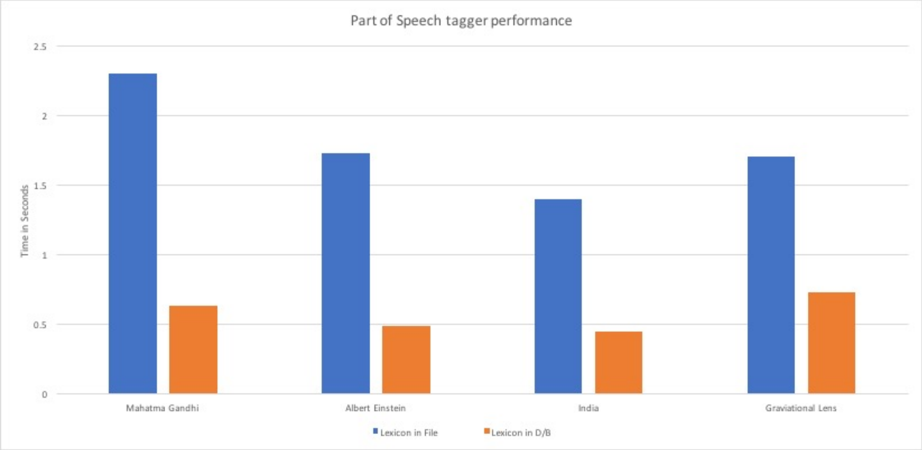


Figure 17: Part of Speech tagging time comparison.

We compare the English and Hindi Question answer systems for relevant answers. I used 4 topics on which I asked the same set of questions in English and Hindi. Figure 18 shows the number of correct answers retrieved on Page 1 of the search result. We can see that English system is better at providing more accurate answers compared to Hindi.

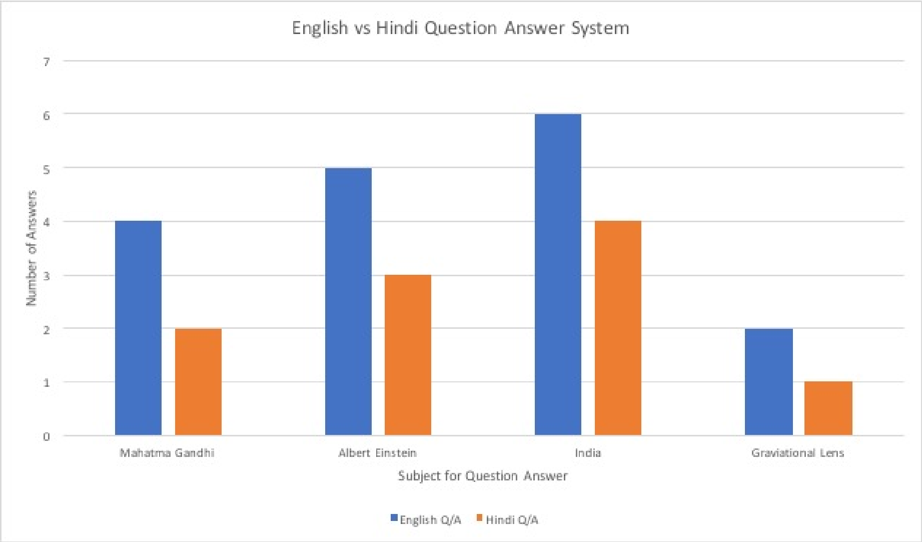
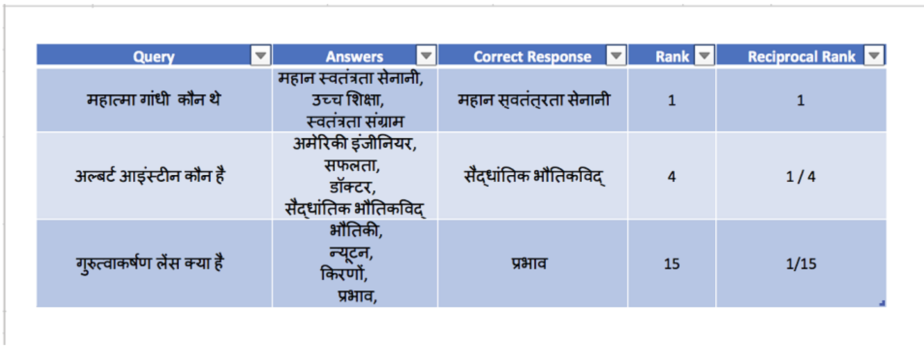


Figure 18: English v/s Hindi Question Answer System.

We calculate the mean reciprocal rank for the same set of questions asked to the English and Hindi Question Answer systems. Figure 19 shows the reciprocal ranks for Hindi



| Query | Answers | Correct Response | Rank | Reciprocal Rank |
|---|---|---|---|---|
| महात्मा गांधी कौन थे | महान स्वतंत्रता सेनानी,<br>उच्च शिक्षा,<br>स्वतंत्रता संग्राम | महान स्वतंत्रता सेनानी | 1 | 1 |
| अल्बर्ट आइंस्टीन कौन है | अमेरिकी इंजीनियर,<br>सफलता,<br>डॉक्टर,<br>सैद्धांतिक भौतिकविद् | सैद्धांतिक भौतिकविद् | 4 | 1 / 4 |
| गुरुत्वाकर्षण लेंस क्या है | भौतिकी,<br>न्यूटन,<br>किरणों,<br>प्रभाव, | प्रभाव | 15 | 1/15 |

Figure 19: Reciprocal Rank for Hindi Question Answers.

We compare the English and Hindi Question Answer System on the Average Precision scores. Average Precision is basically the number of correct answers interpreted as correct from the total number of results returned. Figure 20 shows the score comparison between the 2 systems.

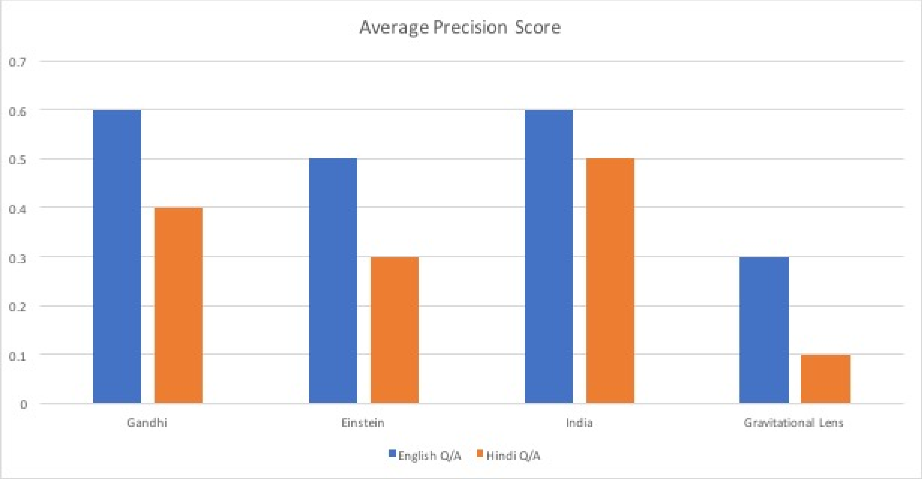


Figure 20: Average Precision Score.

The Mean Average Precision (MAP), as the name says is the mean of all the average precision scores is the measure of accuracy of a information retrieval system. For our test with the Question Answer sytem integration in Yioop, I observed the MAP to be 0.43 for Hindi Question Answer system and 0.61 for the English Question Answer System.

The systems are tested for accuracy comparing the answers retrieved against a known set of answers. I used a set of 25 questions with a corresponding set of answers which are known to be true. I then asked the same questions to the English and Hindi question answering systems in Yioop. Figure 21 and Figure 22 show the efficiency of the systems for simple questions.

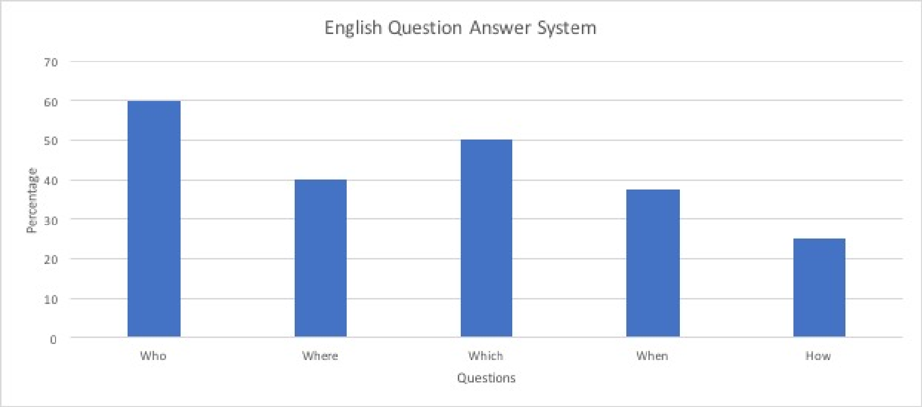


Figure 21: Accuracy of English Q/A in Yioop.

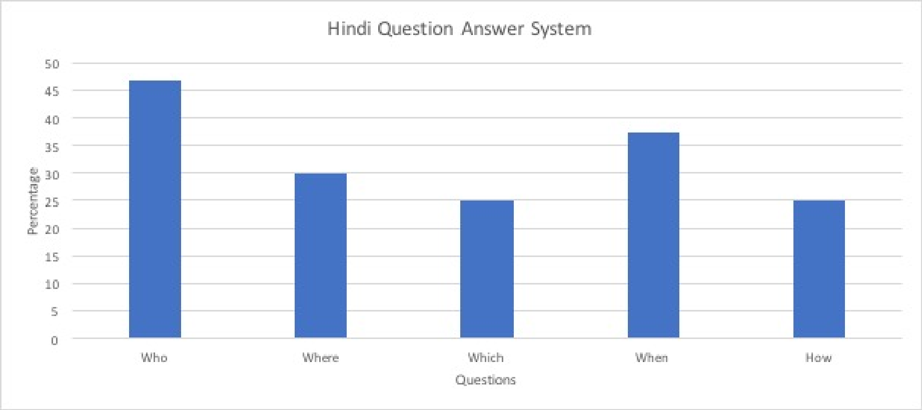


Figure 22: Accuracy of Hindi Q/A in Yioop.

# CHAPTER 5

## Conclusion

As part of the project we improved the existing english question answer system in Yioop and integrated a hindi question answer system in Yioop. The system uses a rule based approach for tagging and generating triplets for Hindi documents and storing them as part of the index. The system is able to handle simple questions asked by the user.

This project is one of the steps to improving and implementing a open source question answer module for two languages english and Hindi respectively. These modules are able to answer simple *Wh* questions in both languages. We have performed some preliminary tests for these languages by creating separate indexes for english and Hindi. The systems were then evaluated by comparing the retrieved answers to a dataset of answers known to be true.

Our systems for English and Hindi are open domain systems. We created a set of questions and answers which are know to be true. We then evaluated our system by asking it questions from the question set and comparing retrieved answers with the known answers in our dataset. Our system accuracy is observed to be 63% for English and 55% for Hindi. Using a rule based approach although faster has it limitation as implementing it requires knowledge about the language. Our system accuracy is caused by the fact that because sometimes the system may convert the user question to a triplet which is not part of the index or simple because the system was not able to extract information to answer related to the subject.

We have followed a rule based approach in implementing the question answer system. Our systems can be improved going ahead by adding improving the text parsing. We can improve the parsing of sentences extracted from the web page summary by identifying the named entities by modifying the part of speech tagging to use the entities added to the lexicon by the user.

# LIST OF REFERENCES

[1] G. G. Chowdhury, ''Natural language processing,'' *Annual review of information science and technology*, vol. 37, no. 1, pp. 51--89, 2003.

[2] D. G. Bobrow, ''A question-answering system for high school algebra word problems,'' in *Proceedings of the October 27-29, 1964, fall joint computer conference, part I.* ACM, 1964, pp. 591--614.

[3] E. M. Voorhees and D. M. Tice, ''Building a question answering test collection,'' in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2000, pp. 200--207.

[4] M. Dua, S. Kumar, and Z. S. Virk, ''Hindi language graphical user interface to database management system,'' in *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, vol. 2. IEEE, 2013, pp. 555--559.

[5] D. H. Warren, ''Efficient processing of interactive relational data base queries expressed in logic,'' in *Proceedings of the seventh international conference on Very Large Data Bases-Volume 7.* VLDB Endowment, 1981, pp. 272--281.

[6] B. Katz, ''Annotating the world wide web using natural language,'' in *Computer-Assisted Information Searching on Internet.* LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE, 1997, pp. 136--155.

[7] G. Nanda, M. Dua, and K. Singla, ''A hindi question answering system using machine learning approach,'' in *Computational Techniques in Information and Communication Technologies (ICCTICT), 2016 International Conference on.* IEEE, 2016, pp. 311--314.

[8] D. Sangeetha, R. Kavyashri, S. Swetha, and S. Vignesh, ''Information retrieval system for laws,'' in *Advanced Computing (ICoAC), 2016 Eighth International Conference on.* IEEE, 2017, pp. 212--217.

[9] M. Alupului, A. Ames, B. Collopy, J. Pesot, R. Pierce, and D. Steinmetz, ''Question-answering system,'' Oct. 18 2016, uS Patent 9,471,668. [Online]. Available: https://www.google.com/patents/US9471668

[10] N. Patel, ''Question answering system for yioop,'' 2015.

[11] J. Bao, N. Duan, M. Zhou, and T. Zhao, ''Knowledge-based question answering as machine translation,'' *Cell*, vol. 2, no. 6, 2014.

[12] M. Iyyer, J. L. Boyd-Graber, L. M. B. Claudino, R. Socher, and H. Daumé III, ``A neural network for factoid question answering over paragraphs.'' in *EMNLP*, 2014, pp. 633--644.

[13] D. Zhang and W. S. Lee, ``Question classification using support vector machines,'' in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval.* ACM, 2003, pp. 26--32.

[14] Z. Zheng, ``Answerbus question answering system,'' in *Proceedings of the second international conference on Human Language Technology Research.* Morgan Kaufmann Publishers Inc., 2002, pp. 399--404.

[15] R. Wongso, D. Suhartono, *et al.*, ``A literature review of question answering system using named entity recognition,'' in *Information Technology, Computer, and Electrical Engineering (ICITACEE), 2016 3rd International Conference on.* IEEE, 2016, pp. 274--277.

[16] D. Chopra, N. Joshi, and I. Mathur, ``Named entity recognition in hindi using conditional random fields,'' in *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies.* ACM, 2016, p. 106.

[17] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, ``Dbpedia: A nucleus for a web of open data,'' *The semantic web*, pp. 722--735, 2007.

[18] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, ``Freebase: a collaboratively created graph database for structuring human knowledge,'' in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data.* AcM, 2008, pp. 1247--1250.

[19] R. High, ``The era of cognitive systems: An inside look at ibm watson and how it works,'' *IBM Corporation, Redbooks*, 2012.

[20] E. Brill, ``A simple rule-based part of speech tagger,'' in *Proceedings of the workshop on Speech and Natural Language.* Association for Computational Linguistics, 1992, pp. 112--116.

[21] E. Brill and M. Pop, ``Unsupervised learning of disambiguation rules for part-of-speech tagging,'' in *Natural language processing using very large corpora.* Springer, 1999, pp. 27--42.

[22] N. Garg, V. Goyal, and S. Preet, ``Rule based hindi part of speech tagger.'' in *COLING (Demos)*, 2012, pp. 163--174.

[23] A. Dalal, K. Nagaraj, U. Sawant, and S. Shelke, ``Hindi part-of-speech tagging and chunking: A maximum entropy approach,'' *Proceeding of the NLPAI Machine Learning Competition*, 2006.

[24] R. A. Bhat, I. A. Bhat, and D. M. Sharma, "Improving transition-based dependency parsing of hindi and urdu by modeling syntactically relevant phenomena," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 16, no. 3, p. 17, 2017.