A Natural Language Question Answering System in Malayalam Using Domain Dependent Document Collection as Repository

Pragisha K.
M. Tech Computational Linguistics
Department of Computer Science and Engineering
Govt. Enng. College Sreekrishnapuram
Palakkad, India-678633

Dr. P. C. Reghuraj
Professor
Department of Computer Science and Engineering
Govt. Enng. College Sreekrishnapuram
Palakkad, India-678633

Abstract—
This paper reports the design and implementation of a Question Answering System in Malayalam. The system finds answers of Malayalam factual questions by analyzing a repository of Malayalam documents. It is designed as a closed domain system that handles four classes of factual questions in Malayalam. It uses the techniques Information Retrieval, Natural Language Processing and Information Extraction in Malayalam to perform the extraction of appropriate responses.

There is no reported work in Malayalam QA systems. This system shows 81% of accuracy in extracting precise answers and 94% of accuracy in extracting correct short answers.

Keywords: Factual questions, Question Answering System, closed domain system, Information Retrieval, Natural Language Processing and Information Extraction.

I. INTRODUCTION

Question Answering System is basically a search program that strives to give the best possible answers to the questions asked by the users by analyzing the knowledge-base it has. The knowledge-base could be local or stored in a remote machine.

Question Answering offers a more intuitive approach to information processing. Given a collection of documents and a natural language question posed by the user, a question answering system attempts to find the precise answer or at least the precise portion of the text in which the information appears. This is unlike the Information Retrieval system in which the entire document that seems to contain information relevant to the user is returned. A question answering system also requires the techniques present in Natural Language Processing. It needs a precise analysis of questions and portions of texts.

Malayalam is a Dravidian language and is morphologically very rich. It is the mother tongue of more than three Crores of people in Kerala, living inside and outside of this state. Lakhs of people living in urban Kerala know Malayalam and are very poor in using English. These people want to know a lot of things about the government, rules and acts of the nation e.t.c. The proposed system can help those people to find precise or short answers of their questions. Ultimate aim of the design is to become a part of the work that create intelligent machines that can receive questions in Malayalam and respond with answers in Malayalam. In the proposed system questions, responses and documents of answer searching are in Malayalam.

The organization of this paper is as follows. The next section gives the literature survey, which is followed by methodology and implementation of the proposed system. Section 5 explains the result and result analysis of the system. Section 6 lists the limitations and future enhancement of the system. Section 7 gives concluding remarks.

II. RELATED WORK

Discussion of open domain QA system MULDER is found in [2]. QA in TREC collection is discussed in [4]. The architecture of a closed domain QA system is the topic of discussion in [5]. Basic ideas of QA system, IR and IE are found in[6]. Detailed description of issues involved in QA system when web as a knowledge base is found in [7]. This paper also explains the architecture of a QA system that handles factual questions in English.

Paper[9] gives basic ideas on QA System, Information Retrieval and Information Extraction system. It explains basic
framework and important components of above systems. It also discusses the explanations of text processing and different technologies involved in the text processing. Architecture and explanation of the QA system NALURI, that uses natural language understanding and advanced reasoning to find the answer, is found in [10].

Currently, Malayalam document retrieval is possible using search engines, but a Malayalam QA system for finding precise answer for a user question is still not available. In Malayalam document retrieval, as same as in English, a set of top ranked documents are displayed as response. User is flooded with information and he want to analyze the retrieved documents to find his expected answer. The proposed system is designed to find precise and short answers of Malayalam factual questions. The system can extract the most appropriate answer of a user query by analyzing an available document collection. In a Master’s thesis, it is very difficult to implement all aspects of a QA System. So the dimensions are set to be limited. They are limited as in the following sections.

A. Domain of the System

The system is designed as a closed domain system. Multiple documents in a single domain are used to find the answer. Domain contains articles related to India such as documents in Indian history, geographical features of India etc. Domain is limited to handle the complexity of Information Extraction module present in the answer extraction procedure.

B. Classes of Questions

The system is designed to handle limited classes of factual questions. The system handles four classes of Malayalam questions and they are

'അക്കാശ', 'അന്യാ', 'അവസാനപുരുഷ' and 'അവസാനാധിനി'.

These are the questions having the meaning what/which, who, when and How many/much respectively. All morphological variations of these question words are handled by the system.

Within this 'അക്കാശ' is highly ambiguous in providing linguistic or knowledge information. The question word 'അক്കാശ' requires analysis of one more word, along with the question word, to identify the class of answer. Questions and expected answers of the system are shown in figure 1.

The question asked by the user is processed using text processing methods. Then the question word and query words are identified. Using this query words text retrieval and answer snippet extraction is performed. These extracted answer snippets are processed and by using information extraction techniques, precise and short answer are identified.

III. METHODOLOGY

This section gives the details of architecture and main modules of the system.

A. System Architecture

Most of the QA systems have a similar framework. The system accepts question through the user interface. Then the user’s question is preprocessed to identify the keywords and question word. The question word says about what is asked by the user, whether it is a person name, location name, explanation or anything else. Other keywords are used for formulating the query for document retrieval.

Based on the stored indexes the documents are retrieved for answer selection. These documents are preprocessed for answer snippet extraction. Answer snippet means the paragraph or sentence that has a chance to contain the answer. This extracted answer snippets are passed to the answer identification module. This answer identification performs scoring, ranking and information extraction to find the answer.

B. Important Modules of the System

Main modules of the system and flow of control in between them are shown in fig.2.

- Question Analysis.
- Text Retrieval and answer snippet extraction.
- Answer identification.

The communication between the modules shows that the question analysis module receives question from the user and do the question analysis procedure. Then the control passes to the text retrieval and answer snippet extraction module. This module communicates with the answer identification module and answer is identified. The detailed block diagram is shown in fig. 3.
System uses the techniques present in information retrieval (IR), NLP and information extraction (IE). The main lexical resources used for the implementation are tokenizer, Malayalam wordnet, stemmer, POS tagger and NER system.

IV. IMPLEMENTATION

A Unicode based approach is used for the implementation. Malayalam words are processed using their Unicodes [12]. This section explains the design of all the modules of the proposed natural language Question Answering system in Malayalam. User can interact with the system through the interface for the QA system. User can ask questions using Malayalam virtual keyboard present in the interface. Answer is displayed at the area allocated in the interface for the display of answer.

Four classes of factual questions are handled by the system. Any morphological variations of these question words also can be handled. The answer is precise answer or short answer extracted by the system from the domain dependent collection of documents. Precise answer is single word or multi word answer and short answer is the complete sentence containing that precise answer. If the system cannot identify the precise answer then it gives a failure message and displays short answer only.

A. Question Analysis

Question analysis module accepts single sentence level questions from the user. The aim of this module is to identify
- the question word present in the question
- the query words required for the text retrieval and answer snippet extraction
- expected set of answer templates

Question word is any morphological variations of the four class of questions. These forms are stemmed and the resulted root form is identified as the question word. All other words except the question word are considered for the query formulation. Using Malayalam Wordnet, this set is expanded to include all synonyms, hypernyms and hyponyms of each word. The algorithm of this module is as follows.

- Accept the question from the user.
- Remove all punctuations.
- Tokenize the question.
- Stemming.
- Identify the question word present in the user’s question.
- Identify and store the surrounding words in the case of question ‘what/which’ (since ‘what/which’ needs one more word to identify the user’s requirement).
- Take all other words except the question word, use Malayalam Wordnet and add synonyms of each word to expand the query.

In this algorithm, steps 2 to 4 perform the question preprocessing. An instance of input and output to question analysis module is shown in figure 4.

B. Text retrieval and Answer Snippet Extraction

Text Retrieval and Answer snippet extraction is an important module of the system, since it performs the selection of answer candidates for answer identification. Based on the query words the answer candidates are retrieved from the document collection. This document collection is indexed and documents which have total keyword match with the question are selected for answer snippet extraction.

For selecting the answer candidates, text preprocessing of the selected texts are performed. This preprocessing has the same steps as the question preprocessing. This procedure includes:
- Detect the sentence boundary of the text.
- Remove all punctuations
- Tokenize the text
- Stemming

This list of root words are used for further processing. The stemmer STHREE, implemented for using with this system,
Fig. 4. An instance of input and output of the question analysis module returns valid root words. So when using Malayalam Wordnet to identify the synonyms, words in the stem list can be passed without applying any other transformations.

C. Answer Snippet Extraction

Answer snippet is the portion of text which has a chance to contain the answer. In this system, sentence level extraction of answer candidates are performed. This system does not consider co-reference resolution and discourse analysis.

This module receives formulated query from the question analysis module. It uses the preprocessed text for further operations. Then it checks the count of match of the query with each sentence. The sentences which have a fuzzy match with the query are selected as the answer candidates.

This answer candidates are represented using a triplet containing the sentence, index and count of match. The sentence is in the stemmed form. The index is used to extract the actual sentence after the operations. The index value is assigned at the time of text splitting. Count of match represents the value of match with the question. This answer candidates are passed to the next module for filtering, i.e., for the selection of answer candidates that contains the most specific word of the query.

D. Most Specific Word Identification for Filtering the Candidates of Answer Selection

This module is used for the filtering of the set of answer candidates extracted by the previous module. It selects answer snippets which show high relevance with the question. This is determined by identifying the most specific word of the query.

The phrase most specific word indicates that the word which shows the exact need of the user, or the word or set of words which can discriminate the question. The implementation of this module is based on the tf/idf [Term Frequency/Inverse Document Frequency] value of the question with respect to the extracted answer candidates. The word which has the lowest value for idf is selected as the most specific word.

E. Answer Identification

This module receives filtered answer candidates from the previous module. The output of this module is the precise answer or short answer for the question. The two sub-modules of this module are,

- Module for scoring and ranking the answer candidates
- Module for Answer extraction using Named Entity Recognition

F. Scoring and Ranking of the answer candidates

The set of answer candidates after filtering contains the answer snippets with most specific word. This module performs the scoring and ranking, and selects the winner candidate. This answer candidate which has the highest score is selected as the winner candidate and this snippet is further processed for answer extraction.

In the scoring, question is assumed with windows having the size from one to number of words in the question. For different size of the window, match of same window size of each candidate is counted. Count of match with each window size is stored. Answer candidate which has the highest value for the possible highest window size is placed in the first position. This answer candidate is selected as the winner. This winner candidate is passed to the next module for further processing.

G. Answer Extraction using Named Entity Recognition

This module receives winner candidate from the previous sub-module. A Named Entity Recognition system is implemented for using in the proposed QA system. This implemented NER system used a hybrid approach containing dictionary based extraction and regular expression method.

The expected named entity of the question is identified by analyzing the question word. For eg., if the question word is ’who’, then the expected named entity should be NAME_PERSON. In the case of question word ’what/which’ two nearest surrounding words of the question word are analyzed to identify the expected answer entity.

If the system can find expected entity in the winner candidate unambiguously then that entity is extracted as the precise answer. This candidate may have more than one entity of the same class. In this case, the system displays a failure message to inform the failure in extracting the precise answer and shows the winner candidate as the short answer. An instance of this type is shown in fig.7.

If the winner is absent with the expected entity, the system contacts the previous sub-module and gets second ranked candidate for further processing.
If the question word does not specify any named entity then the winner candidate is communicated as the short answer. In this case, precise answer is not extracted.

H. Interface of the System

Interface is designed such that, it can accept questions from the user, send the received question to the system and display the extracted answer.

A Malayalam virtual keyboard is present in the interface. User can enter their questions through this. Interface is implemented using Python GUI package.

V. RESULT AND RESULT ANALYSIS

This section shows the result and result analysis of the system.

A. Result of the System

Input of the system is the question given by the user. Output is a precise answer and/or a short answer. Fig. 5 shows an instance of output of the system. This shows the cases in which the system successfully extracted the precise answer. When the system is not able to extract the precise answer, it shows the short answer only. This case is shown in Fig 6.

For Performance evaluation the measures precision and recall are used. Measuring precision has no significant role in a QA system. So the recall is considered. Table 1 shows the result analysis of the system with respect to different question classes.

<table>
<thead>
<tr>
<th>Name of Question Class</th>
<th>No. of questions</th>
<th>No. of short answers extracted correctly</th>
<th>No. of precise answers extracted correctly</th>
</tr>
</thead>
<tbody>
<tr>
<td>'mē' (who)</td>
<td>25</td>
<td>24</td>
<td>21</td>
</tr>
<tr>
<td>'ug' (when)</td>
<td>25</td>
<td>24</td>
<td>21</td>
</tr>
<tr>
<td>'gh/m' (How much/many)</td>
<td>25</td>
<td>24</td>
<td>20</td>
</tr>
<tr>
<td>'ug/m' (what/which)</td>
<td>25</td>
<td>22</td>
<td>19</td>
</tr>
</tbody>
</table>

TABLE I
RESULT ANALYSIS OF THE MALAYALAM QA SYSTEM

The system shows 81% of accuracy in extracting precise answers correctly and 94% of accuracy in extracting correct short answers.

VI. LIMITATIONS AND FUTURE ENHANCEMENTS

A. Limitations of the System

Following are the limitations of the system.

- Domain specific system: The system is implemented as a closed domain system. It uses a repository of domain dependent documents to find the answer. The main module that limits the openness of domain selection is NER module for information extraction. It use a combination of two approaches, dictionary based approach and regular expressions. Creating dictionary entry for all the named entities in an open domain is very laborious.
- Limited number of question classes: The system now handles four classes of factual questions. Extracting answers for the questions that need semantic level knowledge is not handled by the system.
- Co-reference resolution is not considered in the system. In this case, the system gives the extracted short answer as the output and does not find the precise answer.

B. Future Enhancements

- Implementation of open domain QA system.
- Implementation of Malayalam QA system that can answer the questions using semantic level knowledge (eg., By using ontology).
- The system can be modified to handle co-reference resolution.
- System can be enhanced to answer all classes of questions.
VII. CONCLUSION AND FUTURE SCOPE

In this paper we described about the design and implementation of a QA system in Malayalam. It receives Malayalam natural language questions from the user and extract most appropriate response by analysing a collection of Malayalam documents. The system handles four classes of factual questions and it extracts precise answer and short answer for each question. The main answer extraction module is NER in Malayalam.

The proposed system also covered the implementation of some linguistic resources. One of the resources implemented as part of the system is Malayalam Stemmer STHREE. This module can find valid root word of Malayalam inflected words. By finding the valid root words, it becomes useful for all systems which use the lexical resource Malayalam Thesaurus. Another sub-module implemented as part of this system is the NER system for Malayalam. It was implemented using the combination of two methods, dictionary based approach and Unicode regular expressions.

The proposed QA system can handle all different questions in the four classes of questions. It gives expected precise answer for 78% of questions and correct short answer for 94% of questions. This system can be scaled up to include co-reference resolution, openness in domain and handling of all calasses of questions.

REFERENCES

[8] Steven Bird, Ewan Klein and Edward Loper, Natural Language Processing with Python, O’reilly(June 2009).