Geographic Question-Answering System for Hindi Language

Rahul Agrawal*
5323-6632-67
rahulagr@usc.edu

Simrat Singh Chhabra
5054-4614-73
simratsc@usc.edu

Harsh Fatepuria
6826-0244-19
fatepuri@usc.edu

Chinmaya Gautam
2210-4126-83
cgautam@usc.edu

***Abstract*— The purpose of this paper is to describe the experiments of building a Geographic Question-Answering System for Hindi, the official language of India. The following sections describe in detail the methodology used for collection of the corpus/ data, development procedures and evaluation metrics. Furthermore, the results are discussed and the scope for future research described.**

I. INTRODUCTION

Question-Answering systems are a natural way of obtaining information. We focus on geographic questions in Hindi, the fourth most spoken language in the world. Given an input question in Hindi, the system processes it using various natural language techniques, including Parsing and Named Entity Recognition and returns answer to these questions.

We believe that the large scope and reach of this application makes it quite interesting and the sparseness of the corpus of questions makes it a challenging field to explore.

Though there has been some work on question-answering systems in Hindi, to the best of our knowledge, ours is the first paper addressing the sub-domain of geography in detail. Sekine et al. [5] built a Hindi-English cross lingual system which took in English questions and search Hindi newspapers for their answers. Kumar et al [3] developed a QA system for extracting relevant information from Hindi documents. Their work preceded the construction of the Hindi shallow parser so it used other heuristics to get past that issue. The most recent work on Hindi QA systems is Prashnottar [4] but their accuracy is quite limited.

Our work aims to build a new type of system by drawing upon some of the techniques used in these previous works. However, since we are focused on a more specialized domain, we can apply specific methods to improve our system's performance.

II. METHOD

A. Materials

Our data consists of the following:

1) A corpus of questions: A set of around 100 questions were developed by us in order to start working with the system. For these questions to accurately represent how humans converse, we gathered another set of 100 questions through crowd-sourcing (using Google forms).
2) A corpus of Named Entities and their Synonyms: A set of Geographic Named Entities and their translations in Hindi were collected (using web-crawling). Hindi Wordnet [2] was used to find all the synonyms for each of these words. Then, the mappings of words (NERs) for various domains (such as rivers, lakes, cities, and states) and their Hindi translated synonyms were stored in JavaScript Object Notation (JSON) format. These set of mappings would be used for dictionary lookup at later stages.
3) Geographical Information System: The data for the Geographical Information System was collected from DIVA-GIS. Only data for India was used in this project. The data provided by the source (DIVA-GIS) are in two types of formats: (a) *.shp files containing the shapes of various geographical features such as water bodies and rivers, administrative divisions such as states and cities etc. and, (b) *.grd files which are DIVA-GIS grid files containing images of terrains, altitude etc. For this project the data on administrative divisions, rivers and lakes were used. However, these can be easily extended by adding more tables in the database.
   The GIS was modified in the following ways:
   a) GIS didn't have data about capitals of various states. Data for capitals of each state was populated in the database using a SQL script.
   b) Standardized the table names in various databases so that more type of questions could be generalized using lesser number of SQL queries.

Annotations: The questions collected were manually annotated with parts of speech to find patterns in the corpus. Only Nouns (including Proper Nouns) were considered for Named Entity Extraction.

B. Procedure

First, the application workflow is described in brief, and then the experimental procedure is explained along with examples.

A (Hindi) question is taken as input from the user. The encoding format is UTF-8. The process was then split into two steps:

1) Tokenize the query, and find the property which applies to this query (based on a list of properties corresponding to the query types that are supported). eg: distance, area, length, intersection, list, count etc. are some of the properties that are used in this project.
2) Shallow parse the query using a Shallow Parser [1]. The Nouns are then used to get the Named Entities using dictioanary lookup. The dictionary/ mappings generated earlier (II.A.2) were used to find the relevant entities in the query.

*Team 6: The Anaphora Resolution

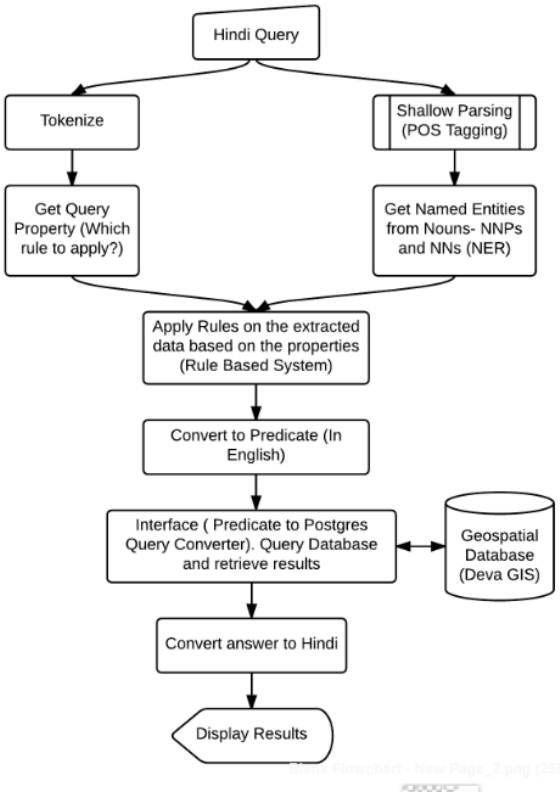

Fig. 1: Workflow of the algorithm

A set of rules are applied on the the extracted Named Entities based on the property it refers to. The predicate thus generated (in English) is fed into a converter program which converts it to a PostgreSQL Query.

A database interface was developed to interact with the database and fetch the results. The answers were then converted to Hindi using a Google Translate API. The above steps are diagramatically expalined in Flowchart 1.

1) Building Rules: Rules are basic building blocks of any rule based system. In this project, rules for the following types of queries were developed:

1) Distance between 2 cities/states
2) Neighbors of State
3) Entities (City/River) in State
4) River flows through which states
5) Area of States/Lakes
6) Capital of State
7) City is in which State
8) Length of River
9) Direction of X from Y (City/State)
10) Entities in Direction of State/City

Pseudo-code for a sample rule is shown below:

```
'''
Rule to get direction property
and parameters
'''
def getDirectionParameters(query):
  NER <- get named entities
  if 'se' or 'ke' Case Marker in query
    L1 <- NER before Case Marker
    L2 <- NER after  Case Marker
```

The following example query illustrates the algorithm steps. CM refers to Case markers in Hindi- 'se', 'ke' , 'ka', 'ki' etc.):

मुंबई से दिल्ली के बीच कितनी दूरी है?
Mumbai se Dilli ke beech kitni doori hai?
Mumbai CM Delhi CM between how-much distance VB
How far is Mumbai from Delhi?

2) Building Predicate: After application of rules on the query, a predicate of property-NER pair is generated in the following format:

```
<query type>
[<subject tag>:<subject value> x n]
```

Here 'n' refers to the number of named entities. The example query discussed above generates the following predicate:

```
distance
{L1:'Delhi' L2:'Mumbai'\}
```

3) Slot Filling Query Generator/ Database Interface: The Slot filling system uses a set of rules to find the most appropriate SQL query. A number of generic SQL queries are written in these programs. The slots are filled up with named entities and property elements. SQL query generated from the above example query is shown below:

```
SELECT ST_Distance(T3.G1,T3.G2) FROM (
SELECT T1.name AS N1, T1.geom AS G1,T2.
name AS N2, T2.geom AS G2 FROM adminis-
trative3 AS T1 CROSS JOIN administrati-
ve3 AS T2 where T1.name like 'Delhi'
and T2.name like 'Mumbai' ) AS T3;
```

This Query is then used to fetch the results from database (GIS) and the results are displayed. The results are translated into Hindi. Answer to the example query is shown below:

दूरी : 1127.85336745 KM
Distance: 1127.85336745 KM

A makefile (with a list of all the dependencies and libraries) was developed to integrate all the components. Thus, we present this project as a proper software system.

C. Evaluation

Question-Answering systems need to be tested on conversational style language to be robust and be able to handle multiple versions of a query. For example, the query, 'How far is Mumbai from Delhi?' can be asked in the following ways:

मुंबई दिल्ली से कितनी दूर है?
Mumbai Delhi CM what distance VB

मुंबई से दिल्ली की दूरी कितनी है?
Mumbai CM Delhi CM distance what VB

मुंबई और दिल्ली में दूरी क्या है?
Mumbai and Delhi CM distance what VB

To test with multiple sentence constructions while evaluating our system's performance, questions were collected from Hindi speakers unfamiliar with our system. The aim here was to evaluate the system in 2 different ways:

1) Questions by people with no prior knowledge of our system and,
2) Questions after telling them the properties of the queries we support.

A total of 10 people were interviewed, and each of them were requested to submit 15 questions each. The first 5 questions were obtained by telling them just the scope of our system (That it contains information about Cities, States, Lakes and Rivers). For the next 10 questions, for each question the volunteer was provided an overview of our rules in English and asked to write a Hindi question related to that rule. The volunteers also provided answers to these questions, which were used as answer keys to evaluate our system's performance.

The answer to the questions is a binary function, i.e. the answers can either be right or wrong. Accuracy is used as the metric to measure the system's performance:

$$\text{Accuracy} = \frac{\text{Answers Correct of Number}}{\text{Questions of Number Total}}$$

As domain specific question answering system have not been developed for Hindi so far, our system's performance was compared against Prashnottar [4], a generic QA system in Hindi (recorded accuracy of 68 Percent).

III. Results

The results of the evaluation are shown in Table I. The accuracy of each type of question is mentioned as well as the overall accuracy of the system (85.56%). The table also shows the accuracy on Open-ended Questions (questions where the evaluators had no information about the type of queries our system handles). As expected, that accuracy is a bit low (33.33%) since our rule-based system is equipped to handle specific types of questions.

The accuracy on questions in which the evaluators were informed of the category is much higher. The system has perfect accuracy for some of the simpler questions such as "Area of states/lakes". However, even for trickier types of questions (which have a larger variety of possible inputs) like "Entities in Direction of State/City" the system gives a reasonably high accuracy of 85.71%. The overall accuracy of domain-limited questions is 85.56% which is much higher than the one obtained by Prashnottar [4].

Type of Question	Correct	Incorrect	Accuracy
Distance between 2 cities/states	9	2	81.81%
Neighbors of State	9	1	90.00%
Entities (City/River) in State	10	0	100.00%
River flows through which states	5	2	71.42%
Area of States/Lakes	6	0	100.00%
Capital of State	10	2	83.33%
City is in which State	7	4	63.63%
Length of River	7	1	87.50%
Direction of X from Y (City/State)	8	0	100.00%
Entities in Direction of State/City	6	1	85.71%
Overall	**77**	**13**	**85.56%**
Open-ended Questions	7	14	33.33%

TABLE I: Results of evaluation

The limitations of rule-based systems is best exemplified by the type of question - "City is in which state." Based on the corpus, which we had created ourselves and collected through crowd-sourcing, we created a rule which looked for a locative case marker. It gave a lower accuracy (63.63%) than the other categories since a few of the evaluators posed the question in a completely novel way. On the other hand, this is also a strength of our system since we can very easily modify our rules to take care of such cases.

Another caveat of the system is that it works even for questions which are not directly mentioned in the type of questions. Due to the standardization of the database, the system has become general enough to deviate from the rigid types. For example, "City is in which state" is supposed to return the state a particular city belongs to. However, after standardization, it can even answer questions such as which state a particular lake (water body) belongs to.

IV. Discussion

Hindi is one of the most widely spoken languages in India. It is imperative that a system is developed which would be able to interact with monolingual Hindi speaking people. This project showcases the development of a domain specific question answering system which can be used as a guideline to develop similar systems in other languages or other domains. The domain of the system can be easily extended for a wider coverage as more data becomes available by adding new rules.

A. Limitations

The major limitations of the system stem from two factors:

1) Sparseness of available data to cover various types of question that can be asked to the system in Hindi. This also limits the effectiveness of a machine learning based approach.
2) Lack of a parser capable of creating a full parse tree to find the association of subject-verb.

B. Future Scope

The system can be made more flexible by collecting more types of questions and growing the synonym list of trigger words. The domain of the system can be increased by populating the database with more data such as population distribution, elevation, land cover etc. and eventually incorporating data of other countries and oceans.

V. Division of Labor

We divided the work as follows:

1) Chinmaya Gautam: Corpus collection, Shallow Parser interface, Development of database, Standardization of database, Evaluation, Documentation
2) Harsh Fatepuria: Corpus collection, Building rules/ coding, Hindi Wordnet, Crawling, Evaluation
3) Simrat Singh Chhabra: Corpus collection, Building rules/ coding, Crawling, Evaluation, Documentation
4) Rahul Agrawal: Corpus collection, Building rules/ coding, Reverse index mapping scripts, Documentation

Word Count: 2051

References

[1] Akshar, R. Bharati, D. Sangal, M, and Sharma. Ssf: Shakti standard format guide. In *LTRC-TR33*, 2007.

[2] S. Jha, D. Narayan, P. Pande, and P. Bhattacharyya. A wordnet for hindi. In *International Workshop on Lexical Resources in Natural Language Processing, Hyderabad, India*, 2001.

[3] P. Kumar, S. Kashyap, A. Mittal, and S. Gupta. A hindi question answering system for e-learning documents. In *Intelligent Sensing and Information Processing, 2005. ICISIP 2005. Third International Conference on*, pages 80–85. IEEE, 2005.

[4] S. Sahu, N. Vashnik, and D. Roy. Prashnottar: A hindi question answering system. *International Journal of Computer Science and Information Technology (IJCSIT)*, 4(2):149–158, 2012.

[5] S. Sekine and R. Grishman. Hindi-english cross-lingual question-answering system. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2(3):181–192, 2003.