A Review of the State of the Art in Hindi Question Answering Systems

Santosh Kumar Ray, Amir Ahmad and Khaled Shaalan

Abstract Question Answering Systems (QAS) are tools to retrieve precise answers for user questions from a large set of text documents. Researchers from information retrieval and natural language processing community have put tremendous efforts to improve the performance of QASs across several languages. However, Hindi, the fourth most spoken language has not seen a proportional development in the field of question answering to an extent that information seekers accept QASs as a good alternative of search engines. In this chapter, a pipelined architecture for the development of QASs has been explained in the context of English and Hindi languages. This chapter also reviews the developments taking place in Hindi QASs while explaining the challenges faced by researchers in developing Hindi QASs. To encourage and support the new researchers in conducting researches in Hindi QASs, a list of techniques, tools and linguistic resources required to implement the components of a QAS are described in this chapter in a simple and persuasive manner. Finally, the future directions for research in Hindi QASs have been proposed.

Keywords Question answering systems · Tools for hindi information retrieval Language resources · Architecture · Query expansion · Document retrieval Answer presentation

S.K. Ray (✉)
Department of IT, Al Khawarizmi International College, Al Ain, UAE
e-mail: santosh.ray@khawarizmi.com

A. Ahmad
College of Information Technology, UAE University, Al Ain, UAE

K. Shaalan
Faculty of Engineering & IT, The British University in Dubai, Dubai, UAE

© Springer International Publishing AG 2018

K. Shaalan et al. (eds.), *Intelligent Natural Language Processing: Trends and Applications*, Studies in Computational Intelligence 740,
https://doi.org/10.1007/978-3-319-67056-0_14

1 Introduction

With the advent of the Internet and the accumulation of huge amount of data in electronic form, there had to be a means for users to retrieve the information they seek from the enormous amount of information. Hence, search engines such as Google and Yahoo were introduced which provided a free platform that would help humans retrieve the required information. However, these search engines assume that if a term in a query was found in a document, then this document could be relevant to that query. The search engines present to the user a list of documents believed to be relevant to his query. They leave it to the user to skim through all the documents to find what he is looking for. In contrast, a Question Answering System (QAS) takes the user question as input, and returns concise and precise answers to the user. So, researchers in QASs investigate approaches and techniques that make it more convenient for users looking for specific and proper information rather than a text file. Thus, a QAS saves the user time, money and the frustration he may have while going through different documents returned by an information retrieval system.

Hindi is the 4th most spoken language in the world. Hindi, a language based on the famous Paninian grammar, is known for its syntactic richness. Nevertheless, research and development in the field of Hindi QASs, compared to efforts in Latin languages, is still in the early stage mainly due to several challenges and language characteristics such as absence of capitalization, free word order etc. To fill this gap and boost up the research work in the field of Hindi question answering, new generation of researchers need to be aware of the state-of-art and know-how of the Hindi QASs. Though some shallow surveys on developments in Hindi QASs have been published, none of them describe the chronological development of QASs, required tools and precise usage of these tools in terms of developments of individual components of QASs. A researcher needs to know how these tools and resources fare in development of QASs. The description of the systems where these tools and resources have been used will provide opportunities to study those systems and organize their research plans accordingly. To the best of our knowledge, the techniques, resources and tools available for designing and developing Hindi QASs and its components have not yet been surveyed extensively, which has motivated us to write this chapter.

The structure of the remaining of the chapter is as follows: Sect. 2 of the chapter provides an architecture of a typical QAS which will work as a roadmap for the development of a Hindi QAS. Section 3 of the chapter reviews the developments taking place in the Hindi QASs. Section 4 introduces the necessary elements of Hindi language and the challenges faced by Hindi QASs. Section 5 provides the details of the tools and resources available for developing Hindi QASs. Finally, Sect. 6 enumerates the possible directions for research in Hindi QASs.

2 A Typical Pipeline Architecture of a Question Answering System

A QAS essentially takes user question as input and presents answers of the question to the user. Accordingly, modern QASs share a number of features and technologies, and the overall designs of the different systems are in most cases quite similar to each other. Most of the QASs follow typical pipeline architecture that divides the question answering process into three distinct phases [1]: question processing, document processing, and answer processing. It should be noted here that none of these phases are mandatory for all QASs. Also, the implementation of these phases may vary to a great extent from one system to another system. In this section, we are providing a generic description of the three phases.

2.1 Question Processing

The question processing phase takes user question as input and applies several processes such as tokenization, stemming, part-of-speech tagging, and query expansion on the question. Thus, the question processing phase can be accomplished by various sub-processes, namely: question classification, derivation of the expected answer type, keyword extraction, and query expansion. Cross-language QASs include an additional process named question translation where the user question is translated into multiple languages [2]. In the rest of this subsection, we shall describe different tasks and subtasks possibly used by different QASs.

2.1.1 Question Classification

Question classification task analyzes the user question and classifies them into one of the predefined classes. The outcome of question classification provides vital information about what to look precisely into documents. Question classification plays a crucial rule in factoid QASs. Moldovoan et al. [3, 4] did a careful survey of the collection of questions in the TREC question collection and identified eight main question patterns; 6 standard Wh-questions, "How" questions, and other questions. Each of these patterns further consists of several sub-patterns. In the following, we provide descriptions of these patterns.[1] These same eight classes can be applied to questions in Hindi also. However, one specific feature of Hindi

[1]Notice that one question can be paraphrased and asked using more than one pattern, getting more than one surface form that share the same meaning. For example the following questions should get the same answer: "During which month tourists visit Kashmir the most?", "What month do tourists visit Kashmir the most?", "Which month do tourists visit Kashmir the most?", and "When do tourists visit Kashmir the most?".

language is worth to mention here. In Hindi, position of question keywords (Wh-questions) may completely alter the meaning of the question. For example, consider the two questions, "क्या आप खाना चाहते हैं? (Do you want to eat?)" and "आप क्या खाना चाहते हैं? (What do you want to eat?)". The second question has been constructed by interchanging the position of first two words of the first question. However, answer of the first question is in Yes/No, but the second question expects name of some food item as answer. Also, though it is grammatically possible to start most of the questions in Hindi with क—question words (counterparts of wh-words in English), in practice, use of क-words at the beginning of question is less frequent compared to that at the middle of sentence, especially if subject (noun) is present in the sentence.

As the focus of this chapter is on Hindi QASs, descriptions of the question patterns are provided first in English, and subsequently equivalent Hindi patterns are described using suitable examples. In all the example questions cited in this section, the Hindi questions are followed by literal English translation with the Hindi word order and then by grammatically correct version of the question.

(i) **Function Word Questions**: These questions contain none of the क—question (non wh-words or how) words. These questions are usually non-factoid questions or explanatory questions. All *Non-Wh-questions (except How questions)* fall under the category of functional word questions.

Example: भारतीय कृषि पर वैश्वीकरण के प्रभाव पर टिप्पणी लिखें। (*Indian agriculture on globalization of effect on comment write*, "Provide comments on the effects of globalization on Indian agriculture.").

(ii) **When Questions**: *"When Questions"* in Hindi contain the keyword "कब (when)" and are temporal in nature. The general pattern in English for "*When Questions*" is "When (do|does|did|AUX) NP [VP] [Complement]?", where AUX, NP, and VP represent auxiliary verb, noun phrase, and verb phrase, respectively. The operator 'l' indicates "Boolean OR" operation and 'Complement' can be any combination of words usually playing insignificant roles in the answer type determination. The constituents written inside '[ ]' are optional. The question pattern of "When questions" in Hindi is much different; the keyword कब (when) rarely appears as the first or the last word of the question. Usually, it appears in the middle of the question. Hence a commonly used pattern for when questions is "NP [Complement] when VP [AUX]?"

Example: भारत को अंग्रेजो से आज़ादी कब मिली? (*India Britishers from freedom when got*, "When did India get freedom from the Britishers?").

Like English, there can be a positional reordering of some constituents of the question in Hindi also. For example, the question "भारत को अंग्रेजो से आज़ादी कब मिली? (*India Britishers from freedom when got*?") can be rewritten as "भारत को कब अंग्रेजो से आज़ादी मिली? (*India Britishers when from freedom got?*)". This is true for all types of questions discussed in this subsection.

(iii) **Where Questions**: *"Where Questions"* in Hindi contain the keyword "कहाँ (where)" question word and relate to a location. These may represent natural entities such as mountains, geographical boundaries, man-made locations such as a temple, or some virtual location such as the Internet or a fictional place. The general pattern for "*Where Questions*" in English is "Where (do|does|did|AUX) NP [VP] [Complement]?". The question pattern of "Where questions" in Hindi is much different; the keyword कहाँ (when) rarely comes as the first or last word of the question. Usually it comes in the middle of the question. Hence a commonly used pattern for when questions in Hindi is "NP [Complement] where VP [AUX]?"

Example: श्री सिद्धिविनायक गणपति मंदिर कहाँ है? (*Shree Siddhivinayak Ganapati Temple where is?*, "Where is Shree Siddhivinayak Ganapati Temple?").

(iv) **Which Questions**: The general pattern for *"Which Questions"* in English is "Which NP [do|does|did|AUX] VP [Complement]?". The equivalent questions in Hindi contain the keyword "किस/ कौनसी/ कौनसा (Which)". The expected answer type of such questions varies and is generally decided by the entity type of the first NP following the keyword "किस/ कौनसी/ कौनसा (Which)". The question pattern of "Which questions" in Hindi may or may not contain keywords किस/ कौनसी/ कौनसा at the beginning of question. Hence a commonly used pattern for when questions in Hindi is "Which NP [Complement] [VP] [AUX]?"

Example: किस राज्य की राजधानी अगरतल्ला है? (W*hich state of capital Agartala is,* "Which state's capital is Agartala?") or अगरतल्ला किस राज्य की राजधानी है? (*Agartala* w*hich state of capital is,* "Which state's capital is Agartala?").

(v) **Who/Whose/Whom Questions**: Questions falling under Who/Whose/Whom category in English have the general pattern as "(Who|Whose|Whom) [do|does|did|AUX] [VP] [NP] [Complement]?. These questions generally ask about an individual, group of individuals or an organization. The Hindi questions in this categories contain the keywords कौन / किसका /किसकी / किसको /किसने (Who/Whose/Whome). The usually adopted form in Hindi for this type of questions is "NP [Complement] (Who|Whose|Whom) VP [AUX]?"

Example:जय जवान जय किसान का नारा किसने दिया? (*Hail the soldier hail the farmer slogan who gave?*, "Who gave the slogan '*Hail the soldier hail the farmer*'?").

(vi) **Why Questions**: *"Why Questions"* always ask for certain reasons or explanations of some facts or events. The general pattern for "*Why Questions*" in English is "Why [do|does|did|AUX] NP [VP] [NP] [Complement]?". The "Why questions" in Hindi contain the keyword "क्यों". The usually adopted form for this type of question in Hindi is "NP [Complement] Why VP [AUX]?"

Example: सोडियम को मिट्टी के तेल में क्यों रखा जाता है? (*Sodium kerosene oil in why stored is?*, "Why sodium is stored in kerosene oil?").

(vii) **How Question**: *"How Questions"* in English have two types of patterns: "How [do|does|did|AUX] NP VP Complement?" or "How (many|much…) NP Complement?". For the first pattern, Hindi provides a keyword "कैसे", and they usually take the form "NP [Complement] how VP [AUX]?". The expected answer type of this type of questions is a description of some process or event. The second pattern of how questions in Hindi contains the keywords कितना/कितने /कितनी, takes the form "NP How many [Complement]?", and looks for some number as the answer.

Example of the first pattern:

सामान्य लोगों के जीवन का पुनर्निर्माण इतिहासकार कैसे करते हैं

? (*Common people life reconstruction historians how do?*, "How do historians reconstruct the lives of common people?").

Example of the second pattern: भारत में कितने राज्य हैं? (India in h*ow many states?,* "How many states are there in India?").

(viii) **What Questions**: "*What Questions*" are most versatile questions which can ask for virtually anything. *"What Questions"* may have several types of patterns. The most general pattern for "*What Questions*" in English can be written as "What [NP] [do|does|did|AUX] [functional-words] [NP] [VP] Complement?". This type of questions in Hindi contain the keyword "क्या". A commonly used pattern for what type of questions in Hindi is "[NP] [Complement] what [VP] [AUX]?".

Example: कंप्यूटर को हिंदी में क्या कहते हैं? (*Computer Hindi in what say?*, "What do you call Computer in Hindi?").

These question patterns (usually represented by regular expressions or context free grammars) are helpful in predicting the expected answer type for a given question.

2.2 Answer Type Determination

After classifying the user query into one of the eight question classes, a QAS predicts the type of entity expected to be present in the candidate answer sentences. Most of the QASs consider following expected entity types in the answers: Person (व्यक्ति), Location (स्थान), Organization (संस्था), Percentage (प्रतिशत), Date (दिनांक), Time (समय), Duration (अवधि), Measure (माप), and monetary values (मुद्रा). Non-factoid QASs can expect reason (कारण), explanation (व्याख्या) as expected answer types, and return a paragraph to the user query. Table 1 summarizes the question types and corresponding expected answer types.

Table 1 Expected answer type for questions

Question type	Question class	Answer type
Factoid questions	When	Date (दिनांक), Time (समय), Duration (अवधि)
	Where	Location (स्थान)
	Which	Person (व्यक्ति), Location (स्थान), Organization (संस्था), Date (दिनांक)
	Who/Whose/Whom	Person (व्यक्ति), Organization (संस्था)
Non factoid Questions	Why	Reason (कारण)/Explanation (व्याख्या)
Hybrid questions	What	Person (व्यक्ति), Location (स्थान), Organization (संस्था), Date (दिनांक), Number (संख्या), Reason (कारण)/Explanatio (व्याख्या)
	How	Reason (कारण)/Explanatin (व्याख्या),
	How many	Percentage (प्रतिशत), Measure (माप), Monetary value (मुद्रा)
	Function Word	Person (व्यक्ति), Location (स्थान), Organization (संस्था), Date (दिनांक), Number (संख्या), Reason (कारण)/Explanatio (व्याख्या)

2.3 Keyword Extraction

The keyword extraction process starts with tokenizing the user query into keywords. A token is the minimal syntactic unit of a sentence; it can be a word or a group of words. These keywords, if needed, are tagged for part of speech. The keywords can then be stemmed to their roots for finding related words. Usually, a tokenizer is implemented as a preprocessing module of POS tagger or named entity recognizer tasks. A POS tagger takes these tokens as input, and assigns POS category to each token. The keywords are then stemmed to their roots for keyword expansion and passage retrieval. Removal of stopwords can be an optional subtask in keyword extraction process.

2.4 Query Expansion

Query expansion process takes the extracted keywords (both original and stemmed) and adds semantically equivalent words to the question with the help of other linguistic sources such as thesaurus, ontology, treebank etc. Query expansion process helps improving retrieval performance of a QAS by increasing the Recall of the QAS [5]. To understand query expansion, consider the question, भारत को अंग्रेजो से आज़ादी कब मिली ? (When did India get freedom from British?).

This question may fetch only documents which contain the words भारत (India) or आज़ादी (freedom). However, there are several documents which contain the words हिन्दुस्तान, भारतवर्ष, हिन्द (different names of India), स्वाधीनता, स्वतंत्रता, मुक्ति, स्वातंत्र्य (frequently used Hindi words for freedom). If we change the above question to (भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द) AND (को अंग्रेजो से) AND (आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति OR स्वातंत्र्य) AND कब मिली?, all the documents containing any combination of these words will be retrieved by search engines.

Considering the importance of query expansion process, modern information retrieval engines are using it to reduce the gap between syntax and semantics of the question and the documents. A detailed survey of the literature provides numerous proposals for the query expansion [5]. Query expansion techniques may be broadly classified as manual, automatic, or interactive [6]. In manual query expansion technique, semantically equivalent queries are obtained and compiled manually. Then the semantically equivalent words are added to the original query through logical operators such as AND, OR and NOT. The modified query is fed into search engines to retrieve the relevant documents. In automatic query expansion, the information retrieval system itself is responsible for expanding the initial or subsequent queries based on some methodology. In interactive query expansion, as opposed to manual or automatic query expansion, the retrieval system and the user both are responsible for determining and selecting terms for the query expansion. An interactive retrieval system is first designed to select, retrieve and rank the expansion terms. The user is then presented with the ranked list of terms, and he has to decide which terms are helpful in the expansion of the query.

2.5 Document Processing

Document processing typically involves identification of documents relevant to user question, and within the set of relevant documents, identification of the passages most likely to contain the answer to the user question. The accuracy in identification of relevant documents will obviously affect the performance of the answer extraction phase [7]. QASs retrieving documents from locally stored documents implement document retrieval modules. One of the widely adopted techniques for identifying the relevant document is to create an inverted index of the document collection. An inverted index provides list of documents in the document base containing a particular keyword in the user query. For example, if the user puts the query किस राज्य की राजधानी अगरतल्ला है? (Which state's capital is Agartala?); in the simplest form, the inverted index will be a list of the documents containing the words "राज्य" (state), "राजधानी" (capital) and "अगरतल्ला" (Agartala) and these documents will be considered as relevant documents. Some of the QASs use stemming of the keywords to increase the recall of the retrieval while

many other systems avoid stemming to avoid compromising the precision of the system [7]. Some other systems use hybrid approaches where they use original keyword as well as stemmed words, but assign less weightage to the stemmed words [8]. On the other hand, web-based QASs usually pass the question keywords and semantically equivalent keywords to one or more search engines such as Google and retrieve the documents with higher ranks [9]. A vast majority of the current information retrieval systems use document retrieval techniques ranging from simple Boolean techniques to sophisticated statistical or NLP based techniques. There is a large variation of document ranking or passage ranking models. Each of these models has its advantages and drawbacks. These models receive the user query and a collection of documents as input and convert them to a non-textual representation. One of the basic document ranking model is Boolean Model. In Boolean model, basic Boolean operators such as AND, OR and, NOT are used for the matching of the query to the document index. Consider the question, कंप्यूटर को हिंदी में क्या कहते हैं? (What do you call Computer in Hindi?), the documents containing both the words "कंप्यूटर" (Computer) and "हिंदी" (Hindi) will be considered more relevant than those containing only one of these words. In this model, the presence or absence of the user query terms in the document is considered, and evaluation of documents only indicates whether they are relevant to the query or not. This set of retrieved documents is presented to the user without giving any consideration to the degree of relevancy. Statistical documents ranking models exploit statistical information about the document such as term frequency, inverse document frequency, document length, etc. to compute the similarity degree of document and the query. Vector Space Model [10] is the most popular model in this category. Probabilistic models provide an intuitive justification for the relevance of matched documents by applying probability theory for ranking documents and uses variant methods for representing the document and the query. One of the well-known of probabilistic models is Inference Model [11], which applies concepts and techniques originating from AI (Artificial Intelligence) without any need to training data sets. Hyperlink based models exploit the hyperlink structures for ranking of documents. These models basically assume that a hyperlink between documents indicates that these documents are on the same topic and one document is recommending some other document. Some of the well-known hyperlink based models are HITS [12], PageRank algorithm [13], WLRank [14] and SALSA algorithm [15]. Finally, Conceptual models [16] work on the principle that there exists some conceptual hierarchy in the documents. These models map the words and phrases in the documents to concepts using the conceptual structures present in the document. Then they extract the concepts of the documents and the query and compare them to compute the degree of similarity.

2.5.1 Passage Retrieval

While identifying relevant documents (or in some other cases after identification of relevant documents), QASs also look for most relevant passages in the documents. Typically a paragraph or a section is selected based on the density or proximity of keywords (or semantically related words) present in them. In this approach, a passage is considered more relevant if it contains a higher number of keywords with minimal distance between the keywords. A review on the keyword density based passage retrieval algorithms and their evaluations can be found in [17]. Another method to retrieve relevant passages is to develop possible answer patterns for the question. To develop the pattern, the question keywords, expanded keywords and expected answer entity obtained from question classification are considered. The passages containing these patterns fully or significantly are considered more relevant. For example, consider the question, भारत को अंग्रेजो से आज़ादी कब मिली? (When did India get freedom from British?). The candidate passages should contain sentences like

दिनांक को भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द को अंग्रेजो से आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति मिली। (On [Date] India got freedom from British.)

भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द को अंग्रेजो से दिनांक को आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति मिली। (On [Date] India got freedom from British.)

[वर्ष] में भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द को अंग्रेजो से आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति मिली। (In [Year] India got freedom from British.)

भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्दको अंग्रेजो से आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति [वर्ष] में मिली। (India got freedom from British in [Year].)

[दिनांक] को भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द को अंग्रेजो से आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति OR स्वातंत्र्य मिली। (On [Date] India got freedom from British.)

भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द को अंग्रेजो स[दिनांक] को आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति मिली। (On [Date] India got freedom from British.)

[वर्ष] में भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द को अंग्रेजो से आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति मिली। (In [Year] India got freedom from British).

भारत OR हिन्दुस्तान OR भारतवर्ष OR हिन्द को अंग्रेजो से आज़ादी OR स्वाधीनता OR स्वतंत्रता OR मुक्ति [वर्ष] में मिली। (India got freedom from British in [Year]).

2.6 Answer Extraction

Answer processing is the final phase of a QAS. It consists of small subtasks such as candidate answers identification, answer ranking, and answer formulation. Candidate answers identification requires full parsing of the passage retrieved by passage retrieval phase and comparing it to the expected answer type derived in the question processing phase. This produces a set of candidate answers that are then ranked according to some algorithm or a set of heuristics [18]. These algorithms or heuristics assign weights to candidate answer sentences. Answer sentences with scores lower than a predetermined threshold score are rejected and remaining sentences are ranked according to their scores. The basic strategies employed in answer identification and ranking are to find named entities that match the expected answer type [19], matching syntactic relations from the questions with those from the corpus [20], or attempting to justify the answer using an abductive proof [21]. The answer formulation process restructures the retrieved answer sentences in the user question specific format.

2.6.1 Named Entity Recognition

The Named Entity Recognition (NER) is an important task in answer extraction process of a QAS. The main objective of NER process is to identify the proper names, or temporal and numeric expressions, and classify them under one of the predefined categories such as organization, person, location, date, etc. Thus, the precision of a QAS depends a lot on the correct recognition of named entities. Chu-Carroll et al. [22] investigated the impact of NER on document retrieval precision and observed an improvement of 15.7% in precision of document retrieval when NER was also used.

The approaches to recognize named entities can be broadly classified into two categories: Rule-based approaches and Machine Learning-based approaches. The rule-based approaches [23] rely on handcrafted grammatical rules to recognize named-entities. The rule-based approaches are accurate but more labor intensive. Machine Learning-based approaches, on the other hand, are less time consuming as once developed, trained and tested over a large data set, they adapt themselves according to new patterns or require a little modification. A hybrid approach for NER was recently introduced which combines the machine learning and rule-based approaches together [24, 25]. This has resulted in significant improvement by exploiting the rule-based decisions of named entities as features used by the machine learning classifier.

2.6.2 Answer Scoring and Ranking

A variety of heuristics are used to evaluate whether the candidate entity/sentence/passage is the real answer or not. These heuristics [26] include the frequency and position of occurrence of a given named entity within retrieved passages. Each candidate answer is assigned some score. The top ranked answers are extracted and presented to the user.

2.6.3 Answer Presentation

The last but not the least important issue in the question answering is the presentation of the answers. Different QASs use different approaches to present the answers. Some of the systems present an entity (name, locations, etc.) as an answer to the factoid question along with some additional information [20]. Some other systems present the answer in a sentence/passage form [27] while many other systems present the link to the relevant passage or document along with the candidate answer sentence [28]. Lin et al. [29] showed that users prefer passages over exact phrase answers in a real-world setting because paragraph-sized chunks provide context. Similarly, the number of candidate answer sentences also varies from system to system. Some of the systems present only one answer while other systems present multiple candidate answers.

3 Developments in Hindi Question Answering System

Larkey et al. [30] developed a cross language English-Hindi information retrieval system. They employed several techniques such as normalization, stop-word removal, transliteration, structured query translation, and language modeling using a probabilistic dictionary derived from a parallel corpus in developing this cross language information retrieval system. They tested the system with 15 queries and 41697 Hindi documents from BBC. The reported mean average precision is 0.4298. Some of the challenges posed by Hindi during cross language information retrieval were proprietary encodings of much of the web text, lack of availability of parallel news text, and variability in Unicode encoding.

In the same year another Hindi-English cross language QAS was developed by Sekine and Grishman [31]. This system accepted question in English and analyze the question for expected answer types. The keywords from the questions were translated to Hindi using bilingual dictionary. Then the system searches for answers containing keywords and expected answer type in pre-annotated Hindi newspaper articles. Once the system finds the relevant text in the newspaper containing expected answer type, it translates the answer to English and presents to the users. This system has a web interface designed using Perl-CGI. They collected BBC newspaper article for 6 months to make the corpus. After removing duplicates, the

final number of articles in the corpus was 5557. The system was tested with 56 questions. The MRR for the top 5 answers for this system was 0.25 which indicates that cross-language QASs are viable options for question answering.

Shukla et al. [32] developed a restricted domain multilingual QAS. They used Universal Networking Language (UNL) [33] to convert contents of a document in Hindi or English to intermediate language. They analyzed user query to determine its focus and expected answer type. Then an answer template for each question was generated which was again converted to UNL expression. Then the UNL expression for question was matched with UNL expression for documents. The matched answers were finally converted from UNL to natural language. The system provided answers with up to 60% accuracy. However, the authors did not report the details such as number of questions and documents used in testing.

Surve et al. [34] designed another language independent restricted domain QAS named AgroExplorer in 2004. The uniqueness of this system was that instead of doing search on plain text, it first extracts the meaning from the user query using UNL structures and then searches for the extracted meaning in the document base. The document base is created by collecting HTML pages from the web, then parsing and converting these documents in UNL representation. The documents are ranked by matching of UNL graph of user query to the UNL graph of sentences in the documents. Documents have more similarities between query graph and document sentence graphs are given higher ranks. However, this system was tested with a set of only 7 documents in the agricultural domain.

The emphasis on cross-language information retrieval in India can be attributed to the fact that there India is a land of linguistic diversity. Though Hindi is understood by a large section of Indians, it is not only major language of India. According to 2001 census, India has 122 major languages and 1599 other languages. However, not all of these languages are used in academic and administrative communications. In fact, there are 22 schedules languages in India which cover all the states of India. Consider this factor, Government of India initiated a consortium project titled "Development of Cross–Lingual Information Access System" where the users could enter query and retrieve answers in the language of their choice [35].

Kumar et al. [36] developed a QAS for E-Learning Hindi Documents. They classified the question into one of six categories: reasoning questions containing words क्यों (why), क्या (what), वर्णन (explain/describe), कैसे (how); numerical questions containing keyword कितना (how many/how much); time related questions containing keywords कब, जब (when); person and location related questions containing keywords किसने (who), किसको (to whom), कौन (who), कहाँ (where), किधर (which side); questions requiring answers from different passages and containing keywords कौन-कौन (who in plural sense), क्या-क्या (What in plural sense), विभिन्न (different): and miscellaneous questions which do not fit into any of the category. Then stopwords were removed from the question and important keywords were filtered out. The important keywords were stemmed to be used in finding semantically equivalent words for query expansion using a self-constructed small

lexical database. The reformulated queries were fed into retrieval engine which used locality based similarity heuristic to select the answer for the given queries. The system was tested with a set of 60 questions whose answers were retrieved from a corpus of Hindi documents related to agriculture and science. According to the authors, the system answered 86.67% of the questions.

Later, Sahu et al. [37] developed a factoid Hindi QAS; Prashnottar, that can answer question questions of type "when", "where", "how many" and "what time". The system uses handcrafted rules to identify question patterns. However, it is not clear how they are extracting answers from document database. The reported accuracy of the system is 68%.

Recently, Nanda et al. [38] propose a Hindi QAS that uses machine learning approach for entity type prediction from the user question. They tested their system over 75 questions. They have not provided any description of the document set. Hence, it is not clear how and from where the system is extracting the answer. The reported accuracy is 90%.

3.1 Developments in Tasks of Question Answering Systems

Cucerzan and Yarowsky [39] developed a language independent model for NER. This model was tested over 5 languages, Hindi being one of them. Among all these language, performance for Hindi was the worst. Later, Li and McCallum [40] developed an NER for Hindi using conditional random fields. The f-value for this model was 71.5. Kumar and Bhattacharyya [41] developed a Hindi NER using Maximum Entropy Model with f-value of 79.7. Saha et al. [42] used a hybrid approach for named entity extraction for Indian languages including Hindi. They used class specific language rules to improve baseline NER based on Maximum Entropy model. They also included some gazetteers and context patterns to improve the performance of the system. The system was trained over half million Hindi words. They reported a precision of 82.76, recall of 53.69, and f-measure of 65.13.

Ekbal and Saha [42] applied simulated annealing based classifier ensemble techniques to POS tagging in Hindi and Bengali. They used, first, the concept of Single Objective Optimization (SOO) for POS tagging, and later developed a method for Multi-objective optimization (MOO). They used Conditional Random Fields and Support vectors for underlying classification. The reported accuracy of POS tagging in Hindi using SOO was 87.67% and 89.88 using MOO. Avinesh and Karthik [43] reported an accuracy of 78.66% for Hindi POS tagging. Ray et al. [44] proposed an algorithm for POS tagger that reduces the number of possible tags for a given sentence by imposing some constraints on the sequence of lexical categories that are possible in a Hindi sentence. Singh et al. [45] used a decision tree based learning algorithm for POS tagging in Hindi. They used a corpora of 15,562 words for training and testing purposes. The reported accuracy of POS tagging is 93.45%.

Akshar et al. [46] developed a parser based on Paninian Grammar formalism to analyse Hindi sentences. This parser based on karaka theory used Integer Programming to analyse simple Hindi sentences.

4 Introduction to Hindi Language and Its Challenges for QASs

Hindi, one of the two official languages of India, is the fourth most-spoken language in the world after Mandarin, Spanish and English. Hindi is written using Devanagari script. The most basic unit of writing Hindi is *Akshara* which can be combination of consonants and vowels. Words are made of aksharas. Words can also be constructed from other words using grammatical constructs called Sandhi and Samaas. Though Hindi is a syntactically rich language, it has certain inherent characteristics that make the computer based processing of the documents in this language, from the information retrieval point of view, a very difficult task. In this section, we are presenting some of these challenges.

- ***No Capitalization***: The factoid QASs require to correctly identify the name of locations, persons and other proper nouns. Identification of proper nouns is done by named entity recognizers which typically exploit the fact that proper nouns in many languages including English are usually started with capital letters. However, Hindi language does not use the capitalization feature to distinguish proper nouns to other word forms such as common nouns, verbs or adjectives. For example, the Hindi proper name "संतोष" [*Santosh*] can be used in a sentence as a first name, or as a common noun.
- ***Lack of uniformity in writing styles:*** In real context, many of translated and transliterated proper nouns tend to be inconsistent. This lack of standardization of the Hindi spelling leads to variants of the same word that are spelled differently but still refers to the same word with the same meaning, creating a many-to-one ambiguity. For example, the word an and (name of a person or happiness) can be spelled as आनंद or आनन्द .
- ***Expressions with multiple words***: It is very common to use same word (or words with similar meaning) consecutively two times in Hindi. For example, the word कौन (who) is used as कौन- कौन in plural sense, धीरे (slow) is used as धीरे-धीरे to emphasize low speed, बहुत (many) सारे (all) are combined together as बहुत सारे (so many). This type of usage of words can be crucial in tokenization process, or it can even negatively affect the performance of cross-language QASs where translation from one language to another language is needed.
- ***Vaalaa morpheme constructs***: The 'vaalaa (वाला)' Hindi morpheme is frequently used in Hindi as suffix to construct new words or to modify the verbs in a sentence. It can take different forms according to gender and number form of the base noun. For example, if we add "vaala" suffix to the word चाय (Tea),

a new word चायवाला (male tea seller) will be formed. However, if we add "vaali" suffix to the word घर (house), a new word घरवाली (wife) will be formed. This can make the automatic word sense disambiguation task more complex.

5 Tools and Resources for Hindi Question Answering

As discussed in Sect. 2, development of a fully functional QAS requires several text processing tasks such as segmentation of user questions and documents in the knowledge base, morphological analysis of question keywords (lemmatization or stemming), determining the part-of-speech (POS) of words, named entity recognition, parsing the question and answers. In order to save their time and energy, researchers can integrate specialized third party open source tools in the main program to perform these tasks. In recent years, a number of tools have been developed for text processing tasks. Many of these tools can be used to implement phases/subtasks of Hindi QASs, and are freely available to the research community. The availability of free tools to the research community will significantly lower down the cost of developing Hindi QAS as compared to tools under license agreements. In this section, we are describing some of the tools and linguistic resources which are freely available and useful in developing components of Hindi QASs.

(a) **Stopwords**: There are certain words in questions which are not useful in question answering once the correct entity type is predicted for the question. These words are called stopwords, and consist of database of most common words that are filtered out prior text processing. Researchers working in the field of Hindi information retrieval have developed their own list of stopwords as and when needed. However, one publicly available list of stopwords can be downloaded from the website.[2]

(b) **Morphological analyzer**: Morphological analysis is an important component of computational linguistic applications. It helps in finding various inflectional and derivational forms of words in a text. As Hindi is a morphologically rich language compared to English, computational linguistic applications such as QASs for Hindi require good morphological analyzers. In order to meet this requirement, a shallow parser was developed at Language Technology Research Centre, IIIT Hyderabad. This parser can be downloaded from the website of LTRC.[3] This parser provides morphological analysis for Hindi sentences and gives the root and other features such as gender, number, tense etc. It also does POS tagging for the sentences.

[2]List of Hindi stopwords, http://members.unine.ch/jacques.savoy/clef/hindiST.txt.

[3]A shallow parser, http://ltrc.iiit.ac.in/showfile.php?filename=downloads/shallow_parser.php.

(c) **Stemmer**: A stemmer conflates morphologically similar words into a single root word. Most of the information retrieval applications use stemmer as one of the most basic components. This helps in reducing the storage size for information retrieval applications as the applications have to store only root words instead of storing several variations of a single word. One of the most popular stemmer used for stemming words in Hindi language was proposed by Ramanathan and Rao [47]. A python implementation of this work is available for public at the website.[4]

(d) **POS Tagger**: POS tagging is an important task in question answering. Classifying the words into various syntactic category helps QASs to parse the questions as well as possible answer sentences. As discussed in Sect. 3.1. Several POS taggers have been developed for tagging Hindi words. However, these POS taggers are not available to public. One publicly available POS tagger for Hindi words can be downloaded from the website.[5] This Hindi POS tagger developed by Reddy and Sharoff [48]. This POS tagger is based on TnT model [49], a popular implementation of the second-order Markov model for POS tagging. The distinctive feature of this tagger is that it does morphological analysis as well as POS Tagging at the same time, and thus mutually benefitting both of the tasks. This Hindi POS tagger supports only Unix based systems.

(e) **Apache openNLP**[6]: Apache OpenNLP is a machine learning based tool for the processing of natural language texts. It can be used for various tasks in QASs such as sentence segmentation, part-of-speech tagging, named entity extraction, parsing, and co-reference resolution. There is no explicit support for any specific natural language from OpenNLP tool. It is a language independent tool which can be used to train models from any language. However, there are some pre-trained models for some tasks in specific languages, These pre-trained models[7] are language dependent and perform well on text in the language of their training only. Because of its language independent nature, OpenNLP has been used for NER [42, 50], for POS tagging and chunking [51] for some of Indian languages including Hindi.

(f) **Ontologies**: Ontologies provide an explicit specification of a conceptualization in a structured knowledge representation formalism that can be used for measuring the similarity between any two fragments of text (a word, sentence, paragraph or document), deriving semantic relations, and finding semantically equivalent words. Some knowledge-based resources are thesaurus, ontology, Wiki, etc. Some ontologies have been constructed in Hindi in various domains such as Grocery [52], health [53, 54], University [55].

(g) **WordNet**: WordNet [56] is a large electronic lexical database of English developed at Princeton University, USA, by a team led by Prof. George Miller

[4]A Hindi stemmer, e http://research.variancia.com/hindi_stemmer/.

[5]Hindi POS Tagger, http://sivareddy.in/downloads#hindi_tools.

[6]Apache OpenNLP, http://opennlp.apache.org/download.html.

[7]Pre-trained models for OpenNLP, http://opennlp.sourceforge.net/models-1.5/.

with an aim to create a source of lexical knowledge. WordNet can be downloaded from the Website of Princeton University.[8] It has been used in numerous NLP tasks and applications with a remarkable success, such as POS tagging, Word Sense Disambiguation [57], Text Categorization [58], and Information Extraction [59]. Originally conceived as a full-scale model of human semantic organization, WordNet has become the most used ontological resource for Information Retrieval applications. It has a rich structure connecting its component synonym sets to each other [60]. Semantic relations in WordNet have been extensively used for query expansion [61], building named entity lexical resources [62], and Word Sense Disambiguation [63].

(h) **Hindi Wordnet**[9]: The Hindi Wordnet, like its English counterpart, is a system that provides lexical and semantic relations between different words in Hindi. Hindi Wordnet groups words according to similarity of meaning. For each word there is a synonym set, or synset representing one lexical concept. The current Hindi Wordnet contains 28687 synsets and 63800 unique words. Each entry of Hindi Wordnet describes sysnset (synonyms), gloss (concept) and its position in Ontology. Each synset in the Hindi WordNet is linked with other synsets through the well-known 16 lexical and semantic relations such as hypernymy, hyponymy, meronymy, troponymy, antonymy and entailment. Java APIs have been written to make Hindi WordNet accessible and searchable for Hindi words. A python implementation[10] of Hindi WordNet is publicly available. A broader version of Hindi WordNet called IndoWordnet[11] supports 19 major Indian languages including Hindi and English.

(i) **Hindi Wikipedia**: Wikipedia pages, after its launch in 2001, have been used extensively in English QASs [26, 27]. Hindi Wikipedia was started in 2003 and since then 116,595 pages[12] have been added to it. Hindi Wikipedia API has been used for cross language retrieval [64], query expansion [65].

(j) **DBpedia**: DBpedia is a community based project created to extract structured information from Wikipedia and make it available on the web. DBpedia has localized version in 125 languages, including Hindi. All these versions together describe 38.3 million things while the English version of the DBpedia knowledge base describes 4.58 million things, out of which 4.22 million are classified in a consistent ontology, including 1,445,000 persons, 735,000 places (including 478,000 populated places), 411,000 creative works (including 123,000 music albums, 87,000 films and 19,000 video games), 241,000 organizations (including 58,000 companies and 49,000 educational

[8]WordNet, http://wordnet.princeton.edu/wordnet/download/.

[9]Hindi WordNet, http://www.cfilt.iitb.ac.in/wordnet/webhwn/.

[10]Python implementation of Hindi WordNet, http://sivareddy.in/downloads#python-hindi-wordnet.

[11]IndoWordNet, http://www.cfilt.iitb.ac.in/indowordnet/index.jsp.

[12]Hindi Wikipedia, https://hi.wikipedia.org/wiki/विशेष:/Statistics, accessed on January, 25, 2017.

institutions), 251,000 species and 6,000 diseases.[13] Due to its strongly structured information base, DBpedia is a very useful source for question processing task of a QAS [66].

(k) **HindiWac corpus**: HindiWaC corpus[14] contains 65 million tokens crawled from the Hindi Internet and it is tagged [67]. This corpus can be used to design and train various NLP as well as machine learning based algorithms.

(l) **Treebanks**: A treebank is a highly structured corpus which is a linguistic resource that is composed of large collections of manually annotated and verified syntactic analyses of sentences that are carefully and accurately annotated. These annotations are very useful for the development of a variety of applications such as tokenization, POS tagging, morphological disambiguation, base phrase chunking, named entity recognition, and semantic role labeling [68]. Considering the importance of treebanks in Hindi NLP applications, Palmer et al. [69] developed a multi-representational and multi-layered treebank for Hindi and Urdu. The expected number of words in final version of this treebank is 400,000 Hindi words and 200,000 Urdu words.

(m) **Lucene**: Lucene[15] is an open source cross-platform text search engine library written entirely in Java. It can be used to index the documents in the corpus-based QASs. Several QASs have used Lucene in indexing [2] and document analysis phase [58, 70]. Lucene contains several classes[16] to perform analysis of Hindi texts.

(n) **GATE**: GATE[17] is an open source free integrated development environment for performing language processing tasks and developing Information Retrieval/NLP tools. GATE has been used for development of QASs [71], Information Extraction [72], ontology learning [73], corpus annotation [74] and other NLP tasks. GATE provides plugins for processing many non-English languages such as Arabic, Hindi, French, and German.

(o) **QANUS**: QANUS[18] is an open source, Java-based Question Answering framework developed at the National University of Singapore with an aim to assist new researchers in building new QAS quickly, and act as a baseline system for benchmarking the performance of new QASs. QANUS implements the typical pipeline architecture of QASs, and includes modules for NER, POS tagging and question classification. It provides the flexibility to the developers in adding/removing modules so that the newly developed system can be easily trained over different datasets and techniques. A fully functional factoid QAS called QA-SYS [75] has been built using the QANUS framework to

[13]DBPedia, http://wiki.dbpedia.org/about, accessed on January, 25, 2017.

[14]HindiWalC corpus, https://www.sketchengine.co.uk/hindiwac-corpus/.

[15]Lucene, http://lucene.apache.org/core/.

[16]Lucene classes for Hindi, https://lucene.apache.org/core/4_1_0/analyzers-common/org/apache/lucene/analysis/hi/package-summary.html.

[17]GATE, http://gate.ac.uk/.

[18]QANUS, http://www.qanus.com/.

demonstrate the practicality of this framework. QANUS has been used for developing individual components of a QAS such as passage retrieval [76], and it can be extended for non-English languages as demonstrated in [77].

6 Future Scopes

Due to efforts of some selected researchers, there has been some progress in research in Hindi QASs. However, considering the advanced level of work done in other languages such as English and some Asian languages, the progress in Hindi QASs is really very far from satisfactory level. This creates scope for several improvements in Hindi QAS. In this section, we are describing some of these scopes.

(a) **Design of Relevant Resources and Tools**: One of the major impediments in development of high quality Hindi QASs is the lack of availability of freely available NLP/IR tools and integrated development environments for new researchers. For example, in an experiment, it was shown that using the existing POS taggers [43], an accuracy of only 14.7% in named entity tagging could be achieved over Hindi tokens [78]. In order to fill this gap, tools (some of these tools are discussed in the previous section) were developed to accomplish some specific tasks such as POS tagging, named entity recognition, stemming etc. But, as these tools were designed by some researchers to perform very specific tasks in their projects, other researchers either could not avail them or had to borrow and assemble these tools to design QASs. Contrary to their English counterparts such as PowerAnswer [4, 79] and START [80] which utilize the deep NLP techniques namely natural language annotation of the knowledge base, semantic parsing, logic proving, word sense disambiguation and other deep NLP techniques, very few Hindi QASs have attempted to incorporate logical representation, discourse knowledge, and other deep NLP techniques. The consistently good performances of PowerAnswer in TREC and CLEF competitions have demonstrated that deep NLP techniques increase the Precision of the question answering process [81]. Hindi QASs can achieve similar level of efficiency if deep NLP and statistical techniques are tweaked and adopted to the need of Hindi information retrieval.

Open source tools are useful for a large number of researchers due to availability of source codes to the researchers. Some of the QASs in English such as ARANEA [29], QANUS [75] release their source codes to help research community in developing new QASs. These systems can serve as baseline systems for new researchers in order to develop and benchmark the new QASs developed by researchers. A similar practice in Hindi QAS research community will give necessary boost to new researchers to understand the design patterns in better ways.

(b) **Development of Non-factoid QASs**: As most of the questions asked on the Web are factoid questions, it was natural for researchers to focus more on factoid questions, and Hindi question answering research is also not an exception. However, users in many fields such as academic and scientific research, politics, arts, etc. require answers containing several paragraphs. These types of questions are called non-factoid questions and usually start with keywords what and why. For example, consider the question बुजुर्गों की बढ़ती जनसंख्या का क्या राजनीतिक निहितार्थ हैं? (What are the political implications of an increasingly elderly population?). To answer such non-factoid questions more accurately, a system may need to analyze several documents, extract multiple passages, and combine them to present the answers. The biggest challenge in the development of non-factoid QASs is the unavailability of training data and linguistic resources. To overcome this problem, most systems train on a small corpus built manually for the specific system [82] or questions collected from frequently asked questions (FAQs) [83, 84]. As the researches on even factoid Hindi QASs are not at par with the Latin languages, it is not surprising that there is virtually no work reported on Hindi non-factoid QASs. Thus, there is lot of scope for researchers to contribute in the field of non-factoid Hindi QASs.

(c) **Development of Collaborative Question Answering Systems**: Collaborative QASs (also called Community QASs) such as Yahoo answers [85] and Wiki Answers are becoming promising alternatives for information seekers on the web [86]. In collaborative QASs, users provide answers to the questions posed by other users and best answers are selected manually either by the asker or by all the participants by voting. Due to the presence of a large number of internet users, these systems cover a very high volume of questions as well as answers for both factoid and non-factoid questions. Secondly, the processing of these question-answer pairs is also relatively simpler than automated QASs. The only problem with these answers is the quality of answers which, if not controlled or filtered, can be highly irrelevant or even abusing too. Recently, some research has been carried out to rank the answers on collaborative QASs so that the quality of the best answers can be improved [87]. Surprisingly, there is no reported work related to the development of collaborative Hindi QASs in the literature. As the number of Internet users is growing rapidly and crossing 460 million in India, we believe that a collaborative QAS in Hindi will be very effective and helpful for information seekers in Hindi language. This will help users to get more relevant information, especially for non-factoid questions.

(d) **Development and Use of Semantic Web Resources**: The semantic web and ontology have become the key technologies in the development of QASs. The semantic web is a mesh of information linked up in a way that it is easily process-able by machines, on a global scale. Ontology is most widely used method to represent domain-specific conceptual knowledge in order to promote the semantic capability of a QAS. Semantic Web resources and Ontologies have been used extensively for query expansion, and they greatly improve the

performance of QASs in answering the questions like "Who wrote 'The pines of Rome'?" even if the user asks it in a different form. While expanding the query, most of the systems expand the query with words belonging to same POS; however, in several cases the words from different POS, but with equivalent meaning, are more useful. Hence, query expansion phase of a QAS must also include cross-POS semantically related words. Ontologies help in assisting to find the semantically related words from different POS. Thus, the development of computational linguistic applications depends a lot on the availability of the well-developed linguistic corpora such as language dictionary, ontology, or treebank. Therefore, the last decade has witnessed the development of domain specific QASs in all fields of life ranging from education [88] to Medical [89], Tourism [90], and Mobile service consulting [18].

Researches in Hindi QASs are seriously lagging behind in developing semantic web resources and exploiting their richness in development of QASs. There is no open source tool available for designing Hindi semantic web resources. With an exception of Hindi WordNet (HWN), there is not a single ontology resource available on the web for Hindi question answering research community and even HWN has not been used widely. Some researchers attempted to develop ontologies in the field of Grocery [52], health [53, 54], University [55]. Some researchers have developed domain specific QASs in Hindi also [32, 34, 36]. However, none of these QASs used the ontologies available in various domains. This gap stresses the need of development of more domain specific ontological resources in Hindi which should also be exploited in the design and development of Hindi QASs.

(e) **Development of Evaluation Standards and Test Beds**: In Sect. 3 of this chapter, we noted that most of the Hindi QASs are not evaluated properly, which will make it impossible to compare their performances with future improvements and proposals. The set of questions and documents used for evaluation of the QASs are entirely disjoint for different researchers, unlike their English counterparts where the systems are tested over a standard set of questions and document collections compiled by a well-accepted institution such as NSIT. However, a TREC style set of standard questions is needed to be developed and provided to research community so that the performance of the Hindi QASs can be benchmarked.

(f) **Use of Blogs and Social Media Data**: Since the last one decade, people across the world are expressing their views and opinions over the blogs and social media. This has resulted into an explosion of data over blogs and social media across the world and the Hindi language is not an exception. People working in different technical and non-technical fields are providing relevant information on their blogs or social media pages. The processing of information on blogs and social media is not a trivial task due to the relatively large presence of typographical, syntactic and semantic errors [91]. A new set of NLP resources, tools and methods are required for efficient handling of large volume of data. In social media such as Facebook and Twitter, users write their views and

comments using something called code-mixing where phrases and words of one language is embedded into another language. Code-mixing is a serious challenge to conventional QASs which deal contents in only one language. Some researchers [92] have taken up this challenge to develop a full-fledged QAs in code-mixed language. As the first step, they have used Support Vector Machine to build a question classification system that predicts answer type of a question written using code-mixing (Hindi and English). But, the progress in the social media based Hindi QAS is still far from the satisfactory level.

Thus, we can conclude that there is a lot of scope for research in the field of Hindi question Answering. Researchers in Hindi question answering can take inspiration from the developments in QASs in other languages across the globe. In this chapter, we have described some of these developments. This field also requires the development of software tools useful to the research community. The recent trend in the field of question answering is the development of QASs in the form of smartphone-based mobile apps as it happened in the case of True Knowledge which has been turned into mobile application Evi.[19] We expect that the similar mobile applications will be developed for Hindi QASs also in the near future.

References

1. Buscaldi, D., Rosso, P.: Mining knowledge from Wikipedia for the question answering task. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC'06), pp. 727–730 (2011)
2. Dolvera-Lobo, M.-D., Gutiérrez-Artacho, J.: Multilingual question-answering system in biomedical domain on the Web: an evaluation, Lect.e Notes Comput. Sci. **6941**, 83–88 (2011)
3. Moldovan, D., Harabagiu, S., Pasca, M., Mihalcea, R., Girju, R., Goodrum R., Rus, V.: The structure and performance of an open-domain question answering system. In Proceedings of the Conference of the Association for Computational Linguistics (ACL-2000), pp. 563–570 (2000)
4. Moldovan, D., Harabagiu, S., Girju, R., Morarescu, P., Lacatusu, F., Novischi, A., Badulescu, A., Bolohan, O.: LCC tools for question answering. In: Proceedings of the 11th Text REtrieval Conference TREC-2002, NIST, Gaithersburg (2002)
5. Efthimiadis, E.N.: Query expansion. Ann. Rev. Inf. Syst. Technol. **31**, 121–187 (1996)
6. Renals, S., Abberly D.: The THISLSDR system at TREC-9. In: Proceedings of 9th Text Retrieval conference, Gaithersburg, MD (2000)
7. Clarke, C.L.A., Cormack, G.V., Kisman, D.I.E., Lynam, T.R.: Question answering by passage selection (MultiText experiments for TREC-9). In: Voorhees, E., Harman, D. (eds.) Proceedings of the Ninth Text REtrieval Conference (TREC-9, pp. 673–683), NIST Special Publication (2000)
8. Araujo, L., Pérez-Agüera, J.R.: Improving query expansion with stemming terms: a new genetic algorithm approach. In: Proceedings of the 8th European Conference on Evolutionary Computation in Combinatorial Optimization, pp. 182–193 (2008)

[19]True Knowledge, http://www.evi.com/.

9. Li, X., Yang, W.Z.: Research on personalized document retrieval based on user interest model. In: Proceedings of 7th International Conference on, Computer Science & Education, pp. 1771–1773 (2012)
10. Lee, D.L., Chuang, H., Seamons, K.: Document ranking and the vector space model. IEEE Softw. **14**(2), 67–75 (1997)
11. Crestani, F., Lalmas, M., van Rijsbergen, C.J., Campbell, I.: Is this document relevant? Probably. A survey of probabilistic models in information retrieval. ACM Comput. Surv. **30**, 528–552 (1998)
12. Henzinger, Monika, R.: Hyperlink analysis for the web. IEEE Internet Comput. **5**(1), 45–50 (2001)
13. Brin, S., Page, L.: The anatomy of a large-scale hyper-textual web search engine. In: Proceedings of the Seventh International World Wide Web Conference, pp. 107–117, Elsevier Science, New York (1998)
14. Baeza-Yates, R., Davis, E.: Web page ranking using link attributes. In: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, pp. 328–329 (2004)
15. Lempel, R., Moran, S.P.: The stochastic approach for link-structure analysis (SALSA) and the TKC effect. Comput. Netw. Int. J. Comput. Telecommun. Netw. Elsevier North-Holland, New York **33**(1–6), pp 387-401 (2000)
16. Vallet, D., Fernández, M., Castells, P.: An Ontology-based information retrieval model. In: Gómez-Pérez, A., Euzenat, J. (eds.) Proceedings of the 2nd European Semantic Web Conference (ESWC 2005), Heraklion, Greece, Lecture Notes in Computer Science, vol. 3532, pp. 455–470. Springer (2005)
17. Tellex, S., Katz, B., Lin, J., Fernandes, A., Marton, G.: Quantitative evaluation of passage retrieval algorithms for Question Answering. In: Proceedings of the 26th Annual ACM SIGIR International Conference on Research and Development in Information Retrieval (SIGIR 2003), Toronto, Canada (2003)
18. Wang, D.S.: A domain-specific question answering system based on ontology and question templates. In: Proceedings of 11th ACIS International Conference on Software Engineering Artificial Intelligence Networking and Parallel/Distributed Computing (SNPD), pp. 151–156 (2010)
19. AbdelRahman, S., Elarnaoty, M., Magdy, M., Fahmy, A.: Integrated machine learning techniques for Arabic named entity recognition. Int. J. Comput. Sci. Issues **7**(4)(3), 27–36 (2010)
20. Katz, B., Lin, J.: Selectively using relations to improve precision in question answering. In: Proceedings of the EACL 2003 Workshop on Natural Language Processing for Question Answering, Budapest, Hungary, pp. 43–50 (2003)
21. Harabagiu, S.M., Pasca, M.A., Maiorano, S.J.: Experiments with open-domain textual question answering. In: Proceedings of the 18th International Conference on Computational Linguistics, Association for Computational Linguistics, Saarbrucken, Germany, pp. 292–298 (2000)
22. Chu-Carroll, J., Prager, J., Czuba, K., Ferrucci, D., Duboue, P.: Semantic search via XML Fragments: a high-precision approach to IR. In: Proceedings of the Annual International ACM SIGIR Conference on Research and Development on Information Retrieval, Seattle, pp. 445–452 (2006)
23. Shaalan, K.: Rule-based approach in Arabic natural language processing. Special Issue on Advances in Arabic Language Processing, the International Journal on Information and Communication Technologies (IJICT), vol. 3(3), pp 11–19. Serial Publications, New Delhi, India (2010)
24. Shaalan, K., Oudah, M.: A hybrid approach to Arabic named entity recognition. Journal of Information Science (JIS). vol. 40(1), pp. 67–87. SAGE Publications Ltd, UK (2014)
25. Oudah, M., Shaalan, K.: Person name recognition using hybrid approach. In: NLDB 2013, LNCS, vol. 7934, pp. 237–248. Springer, Berlin (2013)

26. Ray, S.K., Singh, S., Joshi, B.P.: Question classification & answer validation—a semantic approach using WordNet and Wikipedia. Pattern Recogn. Lett. **31**(13), 1935–1943 (2010)
27. Cao, Y.G., Liua, F., Simpsonb, P., Antieaua, L., Bennett, A., Cimino, J.J., Ely, J., Yu, H.: AskHERMES: an online question answering system for complex clinical questions. J. Biomed. Inf. **44**(2), pp. 277–288 (2011)
28. Zheng, Z.: AnswerBus question answering system. In: Proceedings of the Second International Conference on Human Language Technology Research, pp. 399–404 (2002)
29. Lin, J.: An exploration of the principles underlying redundancy-based factoid question answering. ACM Trans. Inf. Syst. **27**(2), 1–55 (2007)
30. Larkey, L.S., Connell, M.E., Abduljaleel, N.: "Hindi CLIR in Thirty Days," ACM Transactions on Asian Language Information Processing (TALIP), vol 2, Issue 2, pp. 130–142. ACM, New York, NY, USA, June 2003
31. Sekine, S., Grishman, R.: Hindi-English Cross-Lingual Question-Answering system. ACM Trans. Asian Lang. Inf. Process. (TALIP) **2**(3), 181–192 (2003)
32. Shukla, P., Mukherjee, A., Raina, A.: Towards a language independent encoding of documents: a novel approach to multilingual question answering. In: Proceedings of the 1st International Workshop on Natural Language Understanding and Cognitive Science, NLUCS 2004, pp. 116–125, (2004)
33. Uchida, H.: UNL Beyond machine translation. In: International Symposium on Language in Cyberspace, Seoul, Korea Systems. ICEIS Press (2001)
34. Surve, M., Singh, S., Kagathara, S., Venkatasivaramasastry, K., Dubey, S., Rane, G., Saraswati, J., Badodekar, S., Iyer, A., Almeida, A., Nikam, R., Perez, C.G., Bhattacharyya, P.: AgroExplorer: a meaning based multilingual search engine. International Conference on Digital Libraries (2004)
35. CLIA Consortium: Cross lingual information access system for indian languages. In: Demo/Exhibition of the 3rd International Joint Conference on Natural Language Processing, Hyderabad, India, pp. 973–975 (2008)
36. Kumar, P., Kashyap, S., Mittal, A., Gupta, S.: A query answering system for e-learning Hindi documents. South Asian Language Review, vol. XIII, Nos 1&2, Jan-June, 2003. pp. 69–81 (2003)
37. Sahu, S., Vasnik, N., Roy, D.: Prashnottar: a Hindi question answering system. Int. J. Comput. Sci. Inf. Technol. (IJCSIT) **4**(2) (2012)
38. Nanda, G., Dua, M., Singla, K.: A Hindi question answering system using machine learning approach. In: 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT) (2016)
39. Cucerzan, S., Yarowsky, D.: Language independent named entity recognition combining morphological and contextual evidence. Proc. Jt. SIGDAT Conf. EMNLP VLC **1999**, 90–99 (1999)
40. Li, W., McCallum, A.: Rapid development of Hindi named entity recognition using conditional random fields and feature induction (Short Paper). In: ACM Transactions on Computational Logic (2004)
41. Kumar, N., Pushpak, B.: Named Entity Recognition in Hindi using MEMM. In Technical Report, IIT Bombay, India (2006)
42. Saha, S.K., Chatterjee, S., Dandapat, S., Sarkar, S., Mitra, P.: A Hybrid Approach for Named Entity Recognition in Indian Languages. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, Hyderabad, India, pp. 17–24, January 2008
43. Avinesh, P., Karthik, G.: Part of speech tagging and chunking using conditional random fields and transformation based learning. Proc IJCAI Workshop Shallow Parsing South Asian Lang. India **2007**, 21–24 (2007)
44. Ray, P.R., Harish, V., Basu, A., Sarkar, S.: Part of speech tagging and local word grouping techniques for natural language parsing in Hindi. In: Proceedings of ICON (2003)
45. Singh, S., Gupta, K., Shrivastava, M., Bhattacharyya, P.: Morphological richness offsets resource demand-experiences in constructing a POS Tagger for Hindi. In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, Sydney, pp. 779–786, July 2006

46. Akshar, B., Chaitanya, V., Sangal, R.: NLP A Paninian Perspective. Prentice Hall of India, Delhi (1994)
47. Ramanathan, A., Rao, D.: A lightweight stemmer for Hindi. In: Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL), on Computational Linguistics for South Asian Languages (Budapest, Apr.) workshop (2003)
48. Reddy, S., Sharoff, S.: Cross language POS Taggers (and other Tools) for Indian languages: an experiment with Kannada using Telugu resources. In: Proceedings of the 5th Workshop on Cross Lingual Information Access (2011)
49. Brants, T.: Tnt: a statistical part-of-speech tagger. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, ANLC'00, Stroudsburg, PA, USA, pp. 224–231. Association for Computational Linguistics (2000)
50. Ekbal, A., Haque, R., Das, A., Poka, V., Bandyopadhyay, S.: Language Independent named entity recognition in Indian languages. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, Hyderabad, India, pp. 33–40, Jan 2008
51. Dandapat, S.: Part-of-Speech tagging and chunking with maximum entropy mode. In: Proceedings of SPSAL2007, IJCAI, India, pp. 29–32 (2007)
52. Chaware, S.M., Rao, S.: Ontology approach for cross language information retrieval. Int. J. Comput. Technol Appl. **2**, 379–384 (2011)
53. Bhatt, B., Bhattacharyya, P.: Domain specific ontology extractor for Indian languages. In: Proceedings of 10th Workshop on Asian Language Resources, COLING, Mumbai, pp. 75–84 (2012)
54. Mathur, I., Darbari, H., Joshi, N.: Domain ontology development for communicable diseases. CS & IT-CSCP **3**, 351–360 (2013)
55. Dwivedi, S.K., Kumar, A.: Development of University ontology for aSPOCMS. J. Emerg. Technol. Web Intell. **5**, 213–221 (2013)
56. Miller, G.A.: WordNet: a Lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
57. Segond, F., Schiller, A., Grefenstette, G., Chanod, J.-P.: An experiment in semantic tagging using hidden Markov model tagging. In: Proceedings of the Workshop in Automatic Information Extraction and Building of Lexical Semantic Resources, pp. 78–81 (1997)
58. G´omez-Adorno, H., Pinto, D., Darnes, V.A.: Question Answering System for Reading Comprehension Tests. Pattern Recognition Lecture Notes in Computer Science, vol. 7914, pp. 354–363 (2013)
59. Yue, J., Alan, C., Biermann, W.: The use of lexical semantics in information extraction. In: Proceedings of the Workshop in Automatic Information Extraction and Building of Lexical Semantic Resources, pp. 61–70 (1997)
60. Fellbaum, C.: WordNet(s). In: Brown, K. (ed.) Encyclopedia of Language and Linguistics, 2nd Edn. pp. 665–670. Oxford, Elsevier (2006)
61. Zhiguo, G., Chan, W., Leong, H.U.: Web query expansion by WordNet. Database Expert Syst. Appl. Lect. Notes Comput. Sci. **3588**, 166–175 (2005)
62. Attia, M., Toral, A., Tounsi, L., Monachini, M., van Genabith, J.: An automatically built named entity lexicon for Arabic. In: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2010), Valletta, Malta (2010)
63. Li, X., Szpakowicz, S., Matwin, S.: A WordNet-based algorithm for word sense disambiguation. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, pp. 1368–1374 (1995)
64. Sharma, V.K., Mittal, N.: Exploiting Wikipedia API for Hindi-english Cross-language Information Retrieval. In: Proceedings of Twelfth International Multi-Conference on Information Processing-2016, 19-21 Aug 2016, Bangalore, India, pp. 434–440 (2016)
65. Barman, U., Lohar, P., Bhaskar, P., Bandyopadhyay, S.: Ad-hoc information retrieval focused on wikipedia based query expansion and entropy based ranking. In: The proceedings of the Forum for Information Retrieval Evaluation (FIRE)—2012. Dec 2012, ISI, Kolkata, India (2012)

66. Adel, T., Okba, T.: DBPedia based factoid question answering system. Int. J. Web Semant. Technol. **4**(3), 23–38 (2013)
67. Kilgarriff, A., Reddy, S., Pomikálek, J., Avinesh, P.V.S.: A Corpus Factory for many languages. In: Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), 19–21 May 2010. Malta, Valletta (2010)
68. Habash, N., Rambow O., Roth R.: A toolkit for Arabic tokenization, diacritization, morphological, disambiguation, POS tagging, stemming and lemmatization. In: Proceedings of Second International Conference on Arabic Language Resources and Tools, pp. 102–109 (2009)
69. Palmer, M., Bhatt, R., Narasimhan, B., Rambow, O., Misra, D.S., Xia, F.: Hindi syntax: annotating dependency, lexical predicate-argument structure, and phrase structure. In: The Proceedings of the 7th International Conference on Natural Language Processing, ICON-2009, Hyderabad, India, 14–17 Dec 2009
70. Bilotti, M.W., Katz, B., Lin, J.: What works better for question answering: stemming or morphological query expansion? In: Proceedings of Information Retrieval for Question Answering Workshop, at SIGIR (2004)
71. Lopez, V., Victoria, U., Enrico, M., Michele, P.: AquaLog: An ontology-driven question answering system for organizational semantic intranets. J. Web Semant. Elsevier **5**(2), 72–105 (2007)
72. Derczynski, L., Field, C.V., Bøgh, K.S.: DKIE: open source information extraction for Danish. In: Proceedings of the meeting of the European chapter of the Association for Computation Linguistics (EACL), Gothenburg, Sweden (2014)
73. Maynard, D., Bontcheva, K.: Natural language processing. In: Lehmann, J., Voelker, J. (eds.) Perspectives of Ontology Learning. IOS Press (2014)
74. Sabou, M., Bontcheva, K., Derczynski, L., Scharl, A.: Corpus annotation through crowdsourcing: towards best practice guidelines. In: Proceedings of the Language Resources and Evaluation Conference (LREC) (2014)
75. Ng, J.-P., Kan M.-Y.: QANUS: An open source question-answering platform. http://wing.comp.nus.edu.sg/~junping/docs/qanus.pdf (2014). Accessed 1 May 2014
76. Ageev, M., Lagun, D., Agichtein, E.: The answer is at your fingertips: improving passage retrieval for web question answering with search behavior data. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, pp. 1011–1021 (2013)
77. Geirsson, Ó.P.: IceQA: Developing an open source question-answering system. http://www.ru.is/~hrafn/students/IceQA.pdfm (2013)
78. Gali, K., Surana, H., Vaidya, A., Shishtla, P., Sharma, D.M.: Aggregative machine learning and rule based heuristics for named entity recognition. In: Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages, pp 25–32 (2008)
79. Bowden, M., Olteanu, M., Suriyentrakorn, P., Clark, J., Moldovan, D.: LCC's PowerAnswer at QA@CLEF 2006. In Proceedings of CLEF 2006, pp. 310–317 (2006)
80. Katz, B., Borchardt, G., Felshin, S.: Natural language annotations for question answering. In: Proceedings of the 19th International FLAIRS Conference (FLAIRS 2006), Melbourne Beach, FL (2006)
81. Radev, D.R., Qi, H., Wu, H., Fan, W.: Evaluating web-based question answering systems. In: Proceedings of LREC, Las Palmas, Spain (2002)
82. Higashinaka, R., Isozaki, H.: Corpus-based question answering for why-questions. In: Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP), Hyderabad, India, pp. 418–425 (2008)
83. Brill, E., Dumais, S., Banko, M.: An analysis of the AskMSR question answering system. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, Pennsylvania, USA, pp. 257–264, 6–7 July 2002
84. Soricut, R., Brill, E.: Automatic question answering using the Web: beyond the factoid. J. Inf. Retr.—Special Issue Web Inf. Retr. **9**(2), 191–206 (2006)

85. Dror, G., Koren, Y., Maarek, Y., Szpektor, I.: I want to answer; who has a question?: Yahoo! answers recommender system. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1109–1117 (2011)
86. Adamic, L.A., Zhang, J., Bakshy, E., Ackerman, M.S.: Knowledge sharing and yahoo answers: everyone knows something. In: Proceedings of WWW '08, pp. 665–674 (2008)
87. Surdeanu, M., Massimiliano, C., Hugo, Z.: Learning to rank answers to non-factoid questions fromweb collections. Assoc. Comput. Linguist. **37**(2), 351–383 (2011)
88. Arai, K., Handayani, A.N.: Question answering system for an effective collaborative learning. Int. J. Adv. Comput. Sci. Appl. **3**(1), 60–64 (2012)
89. Cairns, B.L., Nielsen, R.D., Masanz, J.J., Martin, J.H., Palmer, M.S., Ward, W.H., Savova, G. K.: The MiPACQ clinical question answering system. In: Proceedings of AMIA Annual Symposium, pp. 171–180 (2011)
90. Kongthon, A., Kongyoung, S., Haruechaiyasak, C., Palingoon, P.: A semantic based question answering system for Thailand tourism information. In: Proceedings of the KRAQ11 Workshop, Chiang Mai, Thailand, pp. 38–42 (2011)
91. Baeza-Yates, R., Rello, L.: How bad do you spell?: the lexical quality of social media. In: Proceedings of the Future of the Social Web, WS-11–03 of AAAI Workshops, AAAI (2011)
92. Raghavi, K.C., Chinnakotla, M., Shrivastava, M.: Answer ka type kya he? Learning to classify questions in code-mixed language. In: The Proceedings of the International World Wide Web Conference Committee (IW3C2), pp. 853–858 (2015)