Question-to-Query Conversion in the Context of a Meaning-based, Multilingual Search Engine

Venkata Siva Rama Sastry K, Salil Badodekar, and Pushpak Bhattacharyya
Indian Institute of Technology Bombay, Mumbai
{sivaram,salil,pb}@cse.iitb.ac.in

***Abstract*—Question-to-Query conversion is to convert a grammatically correct interrogative sentence into one of the several potential syntactically correct declarative sentences or meaningful phrases. We present a multilingual system for Question-to-Query conversion in English, Marathi and Hindi. This is in view of integrating a multilingual forum with a Meaning-based multilingual search engine. We describe two approaches to this problem and the resultant algorithms. We wrote rules to cover different syntactic structure for English Question-to-Query conversion. In the absence of a parser and a POS-tagger for Marathi and Hindi, finding the syntactic structure of a question is difficult. For Marathi and Hindi, we delete word(s) from the question to obtain a query. Lack of a question corpus in Marathi and Hindi made the task challenging. Testing on TREC factoid questions gave encouraging results.**

***Index Terms*—Syntactic structure, Phrase structure grammar, Phrase structure rules, Interrogation, Postpositions, Morphology, Case-markers. General terms: Question, Query, Phrase.**

I. Introduction

WE define Question-to-Query conversion as converting a question into a syntactically correct and meaningful declarative phrase that contains no interrogative word. Syntactically correct means the output query must be according to the grammatical rules of the concerned language. We convert an interrogative sentence [10] into a declarative sentence [10] or a meaningful phrase [10] by performing one or more of the following operations on one or more words in the question.

- Morphological processing
1. Change form

- Syntactic processing
1. Change the order
2. Add
3. Delete

Question-to-Query conversion is useful in building a Question Answering system [14]. Therefore, researchers in Natural Language Processing, Information Extraction, and Information Retrieval are interested in Question-to-Query conversion [12]-[14]. To our knowledge, the work reported in this paper is the first ever Question-to-Query conversion system for an Indian Language.

Map: §2 describes how two important projects namely, *AgroExplorer*, a Multilingual, Meaning-based Search Engine and *aAQUA*, a Question-Answer Forum in the agricultural domain motivate this work. §3 presents different approaches to Question-to-Query conversion. §4 gives details of an English Question-to-Query conversion system. §5 describes Marathi and Hindi Question-to-Query conversion. §6 depicts the application of this work in integrating aAQUA with AgroExplorer. §7 presents the results. §8 presents concluding remarks.

II. Motivation: Integrating aAQUA with AgroExplorer

The problem of Question-to-Query conversion in English, Marathi and Hindi arose when we tried to integrate two independent systems called aAQUA and AgroExplorer. We present a brief description of aAQUA and AgroExplorer and then discuss the motivation behind Question-to-Query conversion.

A. AgroExplorer

AgroExplorer [2] is a Meaning-based, Multilingual search engine that considers the semantics of a query. It is unlike a keyword based search engine that matches only patterns. Universal Networking Language, which is often termed as UNL [3] facilitates meaning-based search and Multilinguality in AgroExplorer. A unique word in UNL represents each concept in a language. Therefore, UNL vocabulary is unambiguous. UNL is a language for semantic representation.

A software called EnConverter converts the source language text to UNL expressions. Fig. 1 shows the query (‘moneylenders exploit farmers’) and the UNL expression for this query. A software called DeConverter converts the UNL expression into the target language. Thus, the translation takes place via UNL. EnConverter converts both the query and the natural language corpus into UNL expressions. The search engine carries out the search on the corpus of UNL expressions. It retrieves a document that matches the UNL expression of the query. Thus, UNL facilitates meaning-based search in AgroExplorer.

A software called EnConverter converts the source language text to UNL expressions. Fig. 1 shows the query (‘moneylenders exploit farmers’) and the UNL expression for

this query. A software called DeConverter converts the UNL expression into the target language. Thus, the translation takes place via UNL. EnConverter converts both the query and the natural language corpus into UNL expressions. The search engine carries out the search on the corpus of UNL expressions. It retrieves a document that matches the UNL expression of the query. Thus, UNL facilitates meaning-based search in AgroExplorer.

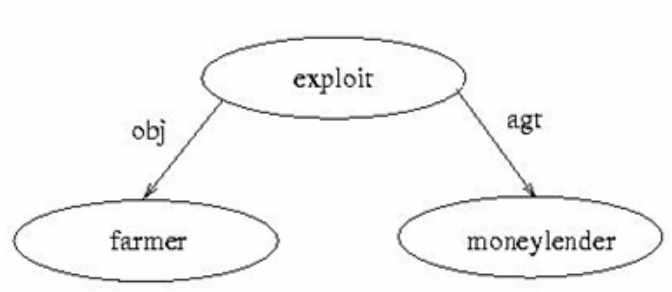

Fig. 1. UNL graph for the query

Enconverted Query

obj(exploit(agt>thing,obj>thing).@entry.@present, farmer(icl>occupation).@def.@pl.@topic)

agt(exploit(agt>thing,obj>thing).@entry.@present, moneylender(icl>occupation).@pl)

Matched UNL document

[s:10]
and(:02.entry, :01)
obj:01(exploit(agt>thing,obj>thing).@entry.@present.@progress.@complete, armer(icl>occupation).@def.@pl.@topic)
agt:01(exploit(agt>thing,obj>thing).@entry.@present.@progress.@complete, moneylender(icl>occupation).@pl)
tim:01(exploit(agt>thing,obj>thing).@entry.@present.@progress.@complete, still(icl>how))
agt:01(provide(icl>give(agt>thing,gol>thing,obj>thing)).@present, moneylender(icl>occupation).@pl)
obj:01(provide(icl>give(agt>thing,gol>thing,obj>thing)).@present, finance(icl>economy))
cob:01(provide(icl>give(agt>thing,gol>thing,obj>thing)).@present, :04)
mod:04(rate(icl>charge).@entry.@pl, interest(icl>profit))
mod:01(:04, exorbitant(mod<thing))
aoj:02(exist(aoj>thing).@entry.@present, cartel(icl>syndicate).@pl)
mod:02(cartel(icl>syndicate).@pl, trader(icl>occupation).@pl)
agt:02(pay(agt>thing,obj>thing,pur>thing).@present, cartel(icl>syndicate).@pl)
obj:02(pay(agt>thing,obj>thing,pur>thing).@present, little(aoj>thing))
man:02(little(aoj>thing), very(icl>how))
gol:02(pay(agt>thing,obj>thing,pur>thing).@present, produce(icl>result))
mod:02(produce(icl>result), they(icl>persons))
plc:02(pay(agt>thing,obj>thing,pur>thing).@present, :03)
man:02(:03, even(icl>how))
mod:03(mandi(icl>market).@entry.@def.@pl, recognized(mod<thing))
man:03(recognized(mod<thing), well(icl>how))
plc:03(mandi(icl>market).@entry.@def.@pl, country(icl>region).@def)
[/s]

B. aAQUA

aAqua [1] is an acronym for **a**lmost **A**ll **QU**estions **A**nswered. It is a Multilingual Forum. People from different communities and different languages can access it. A user posts a question relating to a particular domain. Human experts in the domain answer the question. If an answer already exists in the database of answers, the AgroExplorer Search Engine retrieves it. Otherwise, we convert the user's question into a query and pass it to the search engine. This is the point of integration. Fig. 2 illustrates this. It is the *query* that EnConverter converts to UNL and not the *question*.

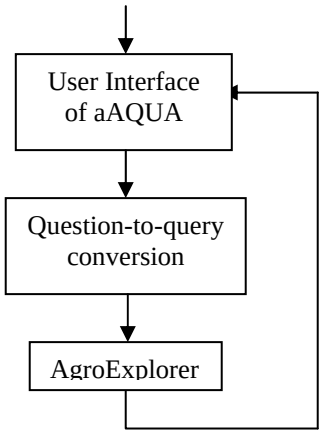

Fig. 2. Integration of aAQUA with AgroExplorer

III. Approaches to Question-to-Query Conversion

There are two approaches

1. Phrase structure rules
2. Deletion of question parts

A. Question-to-Query Conversion using Phrase Stucture Rules

Following this approach, we wrote phrase structure rules for different syntactic structures of questions. A rule consists of two parts. The first part identifies the syntactic structure of the question with the help of a Parts-of-Speech tagger and a Parser. The second part converts the question of the respective syntactic structure into a query.

IF question matches pattern P
THEN take action A
where A is one of the following.

1. Exchange the word positions in the question
2. Add some words to the question
3. Delete some words from the question
4. The combination of the above three methods

This approach is feasible for a language supported by rich NLP tools. In particular, it is feasible for English and currently

infeasible for Marathi and Hindi.

B. Question-to-Query Conversion by Deleting some part of the Query

This approach is applicable to a language that does not possess linguistic resources like POS-tagger and Parser. We delete a part of the question (Phrase to be deleted: DP) from the question. This leaves us with the query. This approach works for a language only if its syntax permits such an operation. The syntax of Marathi and Hindi does permit such an operation. We took into account the complete morphology of the phrase to be deleted i.e., all the inflections and the words derivable from the root of an interrogative word.

IV. English Question-to-Query Conversion

We took the approach of writing rules for different syntactic structures of the questions for converting English question to English query. Rule writing for English Question-to-Query conversion became feasible due to the availability of linguistic resources like POS-tagger and parser in English. Fig. 3 illustrates English Question-to-Query conversion.

We pass the question in English to both link parser [6] and Brill tagger [8]. Fig. 3 shows this. We merge the output of the link parser with that of Brill tagger's. We parse this information and identify the syntactic structure of the question. Next, we apply the respective algorithm on the question to transform it into a query.

A. Example

Fig. 4 illustrates the entire process of English Question-to-Query conversion using the question “What do farmers want?". We pass the question to both Link parser and Brill tagger. The output of Link parser is “What do NP VP". NP means noun phrase and VP means verb phrase. However, Link parser does not mark the syntactic category of the first instance of the word ‘do’. Therefore, we use the output of the Brill's tagger to determine the syntactic category of the word ‘do’ to be VBP. The parser does not categorize a Wh-word. We use output of Brill's Tagger to obtain the syntactic category of all the other unmarked words. We merge the two outputs to get. “What VBP NP VP". This is the syntactic structure of the question. We map it on to corresponding generic syntactic structure i.e., we map “What VBP NP VP" on to “What verb_plus noun_plus verb_plus". verb_plus means one or more verbs or verb phrases and noun_plus means one or more nouns or noun phrases. We wrote rules at the level of positive closure of phrases. Positive closure of a symbol X is the set of strings formed from X such that the length of the string formed is greater than or equal to one i.e.

Positive_closure(X):= {X, XX, XXX, ...}

Fig. 5 illustrates the exact meaning of rules written at the level of positive closure of phrases.

Taking positive closure of phrases as building blocks reduces the effort in writing phrase structure rules [5]. However, the effort did not actually reduce since developing an algorithm for each generic syntactic structure as shown in Fig. 5 took a lot of effort.

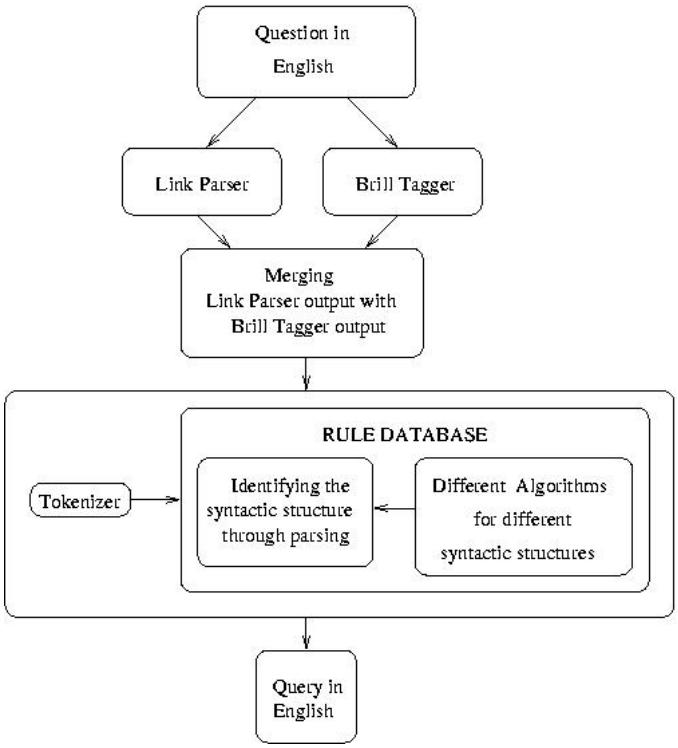

Fig. 3. English Question-to-Query Conversion

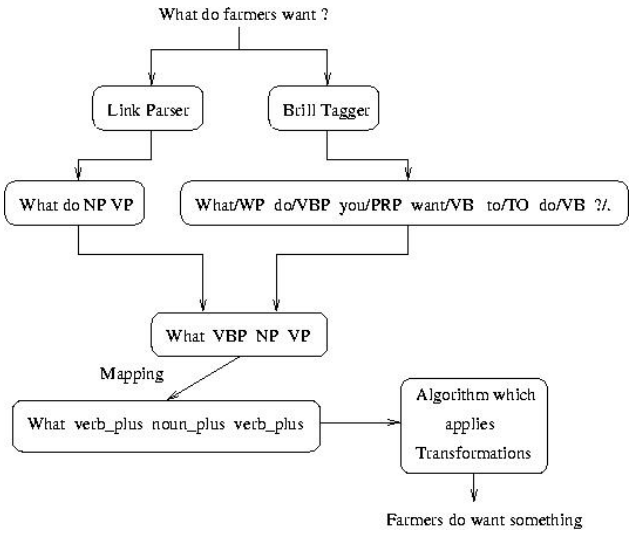

Fig. 4. An example of English question-to-query conversion

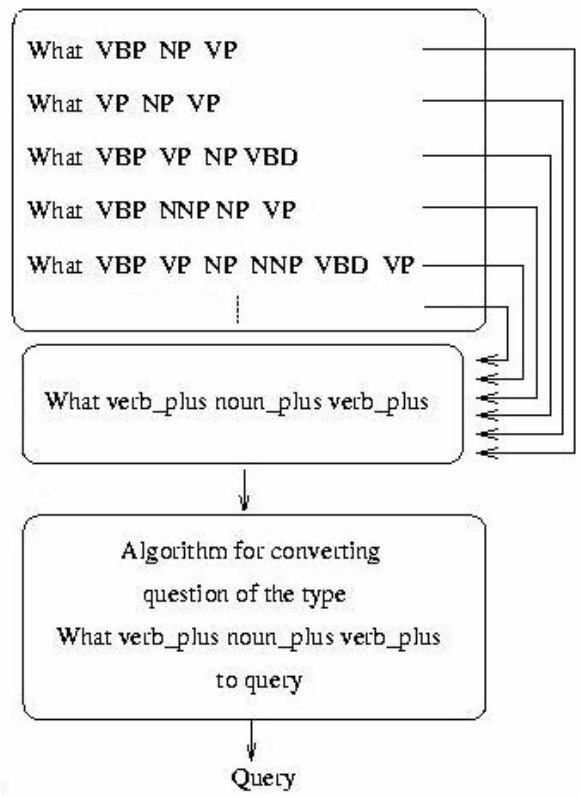

Fig. 5. Convergence of different syntactic structures to a single generic syntactic structure

We had to consider a large number of syntactic structures converging to a single generic syntactic structure. For all the questions that converge to a single rule, we had to write a generic algorithm that converts these questions to their respective queries.

B. Cases Handled

We have written approximately 1,100 rules for English Question-to-Query-conversion. We handle a question that falls into one of the following categories.

1. Yes/No: A question that can be answered in a 'Yes' or a 'No'. It starts with one of the following words.

{ am, is, are, was, were, ain't, isn't, aren't, wasn't, weren't, does, do, did, doesn't, don't, didn't, has, have, had, hasn't, haven't, hadn't, can, could, can't, couldn't, may, might, shall, should, shan't, shouldn't, will, would, won't, wouldn't }

Question: Is agriculture a risky business?
Query: Agriculture is a risky business.

2. Wh.: A question that starts with one of the following words.

{who, what, when, where, which, why, whom, how}
Question: What do farmers want?
Query: Farmers do want something.

3. Preposition: A question that starts with one of the following prepositions.

{for, from, in, to, at, after}
Question: In which soil does rice grow?
Query: Rice does grow in some soil.

4. Conditional: A question that involves 'If - then' construction.

Question: If I come, would it help?
Query: It would help if I come.

5. About: A question that starts with About

Question: About how many soldiers died?
Query: Many soldiers died.

6. Compound: A question formed by an ANDing or/and ORing of questions of the types above.

Question: Who sold DVD and who owns it?
Query: Someone sold DVD.
Someone owns it.

V. Marathi and Hindi Question-to-Query Conversion

We handle Question-to-Query conversion in Marathi and Hindi in a manner different from the one in English. Computational linguistic resources like POS-tagger and parser are not available for Marathi and Hindi. Therefore, writing rules on syntactic structures of questions is not possible in these languages. It is interesting to note that for converting a question to a query in these languages, there is no need for changing the word positions. Deletion of words suffices. We call the contiguous chunk of words to be deleted a Phrase to be Deleted (DP for short). We delete the DPs from the question. Fig. 6 provides an illustration.

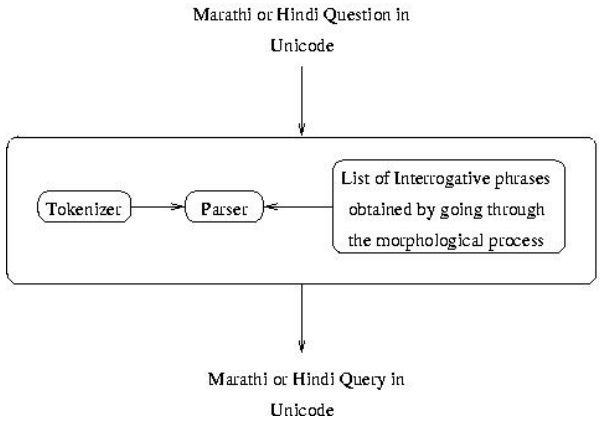

Fig. 6. Question-to-Query Conversion in Marathi and Hindi

A. Collecting Phrases to be Deleted through Morphological Processing

To convert a question in Marathi or Hindi into a query, we identify and remove the phrases to be deleted. We collected DPs in Marathi and Hindi at word level and at phrase level. This was a challenging task since there is no good corpus is available for the two languages. At word level, we collected all the forms. Marathi is rich in morphology. Table 1

Table 1. Phrase level forms of interrogation in Marathi and Hindi

T	U	G	N	C	P	CMB	A	O
R	कधी	केवढा	केवढे	केवढ्यात	कुठपर्यंत	कधीपासूनच्यांचे	कितवा	कितपत
N	कधीकधी	कोणाकोणाची	कशाकशांचा	कशाकशाने	कोणाकोणापर्यंत	कशाकशातून	कुठकुठला	कितीकितीदा
S	कधी कधी	[-]	[-]	[-]	[-]	[-]	[-]	[-]
H	[-]	[-]	[-]	[-]	[-]	[-]	[-]	[-]

illustrates the variety of roots of interrogative words and their inflected forms in Marathi using different root words.

Legend: T: Type, U: Uninflected, I: Inflected, G: Gender, N: Number, C: Case, P: Postposition, CMB: Combination of any of G, N, C, P, A: Adjectival, O: Odd, R: Root, RD: Reduplicated, S: Reduplicated and separated by Space, N: Reduplicated and separated by No space, H: Reduplicated and separated by hyphen.

The morphological inflections and derivations relate to gender, number, case and postpositions [11], [15].

At phrase level, we considered all the interrogative phrases consisting of two or more interrogative words. These phrases contain interrogative words joined together. Table 2 shows the rich variety.

Legend: C: Conjunction, D: Disjunction, WCC: Case-Case pair Without conjunction or disjunction in between-, WPP: Postposition-Postposition pair Without conjunction or disjunction in between

Table 2. Phrase level forms of interrogation in Marathi and Hindi

Type	Marathi	Hindi
C	कधी **आणि** कुठे	कब **और** किधर
D	कुठे **किंवा** कोणाकडे	किधर **या** किस के पास
WCC	कोणा**चे** कोणा**शी**	किस **के** किस **से**
WPP	कुठ**पासून** कुठ**पर्यंत**	कहाँ **से** कहाँ **तक**

At sentence level, we consider all the syntactic possibilities of the location of interrogative words and phrases.

B. Algorithm for Question-to-Query Conversion in Marathi and Hindi

Algorithm:

lang takes the value Marathi or Hindi.

Input: A question in lang.

Output: A declarative sentence or a meaningful phrase in lang.

Let each of A and B is a set of lang words.

X is a meaningful query or phrase obtained from the information content in A.

Y is a meaningful query or phrase obtained from the information content in B.

num is the number of phrases to be deleted (DPs) in the given question obtained by parsing the question using the grammar for a DP.

temp takes the value का or की in Marathi

temp takes the value या in Hindi

```
if num==0 then
  if question is of the form "A temp B?" then
    Output both X and Y.
  else
    Output the given question without question mark.
  end if
else if num==1 then
if question is of the form "if A then B?" then
    Output A and B.
else if question is of the form "A then B?" then
    Output A and B.
else if question is of the form "B if A then?" then
    Output A and B.
else if question is of the form “B if A?” then
    Output A and B.
else if question is of the form "B A then?" then
    Output A and B.
else if question is of the form "B, A then?" then
    Output A and B.
else
    Remove the single phrase to be deleted from the question.
end if
else if num>1 then
Remove all the phrases to be deleted from the question.
end if
End algorithm
```

The above algorithm takes as an input a Marathi or Hindi question. It generates as an output a declarative sentence or a meaningful phrase in the respective language. It considers the number of phrases to be deleted in the question. We use grammar for a phrase to be deleted to find the number of phrases to be deleted in the given question.

C. Phenomena

We handle the following phenomena.

1. Disjunction (A OR B)

1.1 Disjunction of verb phrases
1.1.1 Only a noun present in the B part
1.2 Disjunction of verbs
1.2.1 Only the auxiliary verb present in the B part
2. Conditionality
2.1 'IF A THEN B?'
2.2 'A THEN B?'
2.3 'B IF A THEN?'
2.4 'B IF A?'
2.5 'B, A THEN?'
2.6 'B A THEN?'
3. <mhaNaje>: DP is preceded by '<mhaNaje>' only
3.1 IP is preceded by '<mhaNaje> only
3.2 IP is preceded by '<mhaNaje> <adj_uninf>' only
3.3 IP is preceded by '<mhaNaje> <adj_inf>' only
3.4 IP is preceded by '<mhaNaje> <adj_inf> <adj_uninf>' only
4. Multiple Word DP
4.1 Form of DP: Interrogative word followed by an interrogative word
4.2 Form of DP: Interrogative word followed by a non-interrogative word
4.2.1 The non-interrogative word is a noun
4.2.2 The non-interrogative word is a verb
5. Nested interrogation: Requires #(DP)>1

D. Interesting Features

We handle the following features.

1. Nested/Embedded interrogation
2. The phenomenon of reduplication
3. Adjectival forms, case-markers, postpositions, number, gender
4. Multiple samanyarupas (form of a word before a suffix is attached to it)

Following situations may adversely affect the search.

1. A query may generate a sense not intended in the question.

Table 3. Query may generate an unintended sense

	sense of 'nakkI'
Question: rakkama nakkI kiwI Ahe?	exactly
Query: rakkama nakkI Ahe.	definite/fixed

2. A question that is not formal or grammatically correct may generate an empty query:

Question: javaLapAsa mhaNaje nemake kuTe?
Query: [empty]

VI. Integration of AgroExplorer with aAQUA

The motivation behind developing Question-to-Query conversion module was integration of aAQUA with AgroExplorer. aAQUA is a multilingual forum. AgroExplorer is a Meaning-based, multilingual search engine. Fig. 7 illustrates the integration.

Fig. 7 shows the following. The user posts a question on aAQUA. The reply by experts may take time. The user might wish an immediate reply instead of having to wait for an expert to reply. The Question-to-Query conversion module comes into play. The question is passed from aAQUA to Question-to-Query conversion module. Depending upon the language in which the question is posted, one of the English, Marathi or Hindi modules is activated and it produces a query in the respective language. Before passing on the query obtained to the search engine, we observe a very important thing: the EnConverter module may fail to generate a UNL query from the natural language query due to one of the following reasons:

1. The query obtained from the Question-to-Query conversion module may sometimes be syntactically incorrect.
2. No rules have been written for the EnConverter to handle certain types of queries.

In the above two cases, EnConverter cannot produce UNL query which means the search engine will not be able to produce results. To handle these two situations, keywords are produced as output along with the query from Question-to-Query conversion module. Fig. 7 shows this. Keywords are nothing but content words [9] in the given question. Removal of function words [9] from the given question produces Keywords.

Both the query and keywords are passed as input to the EnConverter. EnConverter gives Universal Words for the keywords. It may or may not produce UNL query from the given natural language query. If EnConverter produces the UNL expression for a query, then we pass the UNL expression directly to the AgroExplorer. This produces the search results. If EnConverter does not produce UNL expression for a query, then we pass Universal Words to AgroExplorer. It searches on the Universal Words and produces the search results. Thus, the search engine always produces the results.

The said search engine is a phrase-based search engine. Its place is between a keyword-based search engine and a question-based search engine: see Fig. 8.

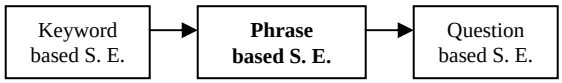

Fig. 8: The place of Phrase-based Search Engine

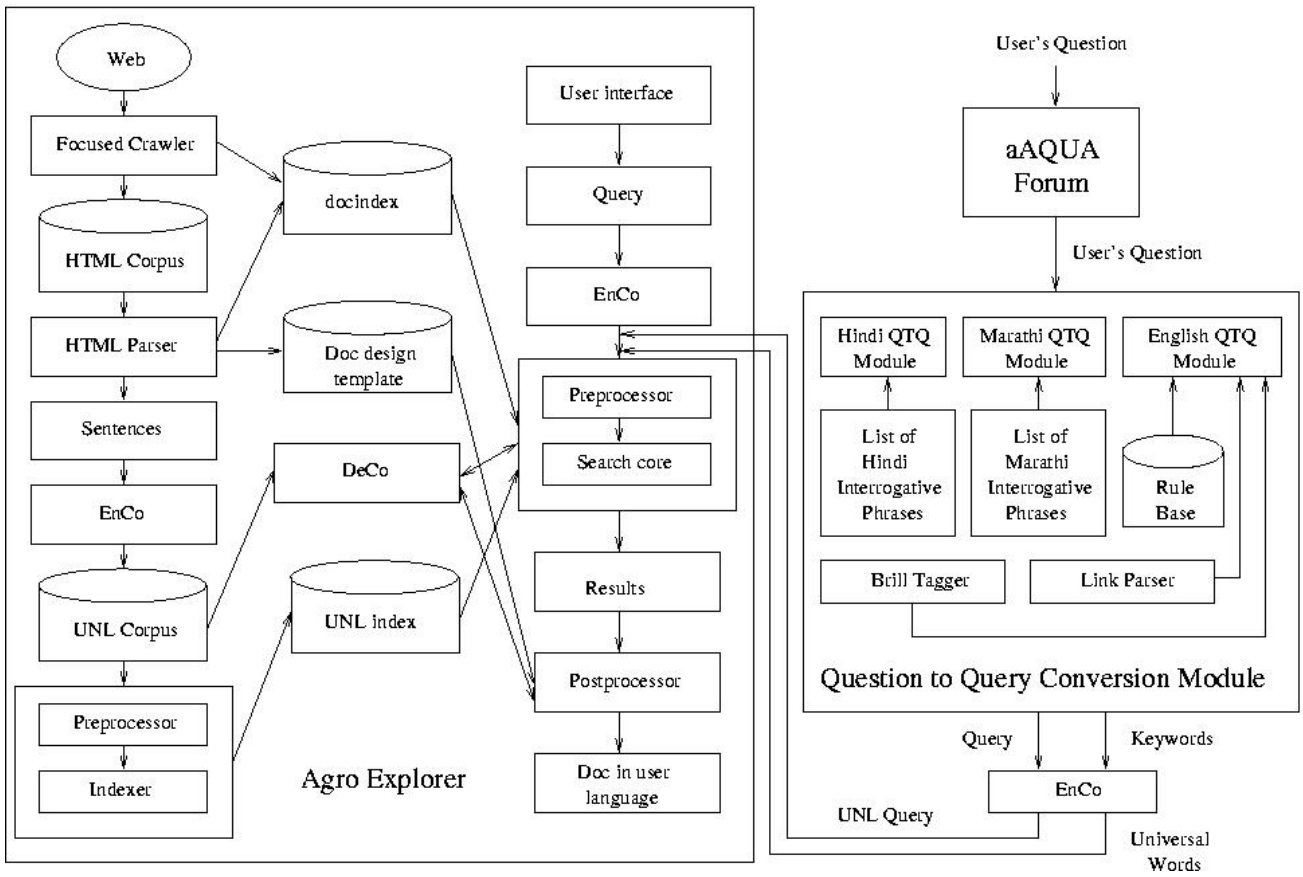

Figure 7: Block diagram showing the integration of aAQUA with Agro-Explorer

VII. Results

The English Question-to-Query conversion module was tested on TREC-Questions. TREC (an acronym for Text REtrieval Conference) is one of the prestigious conferences where question-answering systems are checked for accuracy.

A question set is used to test the question answering systems every year. The questions in the set are called Factoid questions. The English Question-to-Query conversion module was tested for accuracy on these Factoid questions. Table 4 shows the results.

Table 4: Accuracy of English Question-to-Query conversion

Set	No. of Questions	Accuracy (%)
TREC-1999	200	92.00
TREC-2000	692	84.68
TREC-2001	500	94.20
TREC-2002	500	86.40
TREC-2003	500	93.60

VIII. Conclusion

AgroExplorer is a meaning-based, multilingual search engine that performs meaning-based searches on the queries with UNL as the underlying technology. aAQUA is a multilingual forum. Unsatisfactory handling of interrogative sentences by EnConverter posed a problem for the integration of AgroExplorer and aAQUA. We attempted to solve this problem by developing a multilingual Question-to-Query conversion system. It converts English, Marathi and Hindi questions to syntactically correct and meaningful queries. Thus, it made the integration feasible.

In the ongoing work, we intend to improve the performance of Question-to-Query conversion system by the following actions:

For English, write more rules to cover

- The more complex syntactic structures
- More prepositions

For Marathi and Hindi, once a parser and a tagger are available

- Handle disjunctive questions
- Allow insertion if necessary

REFERENCES

[1] Krithi Ramamritham, Anil Bahuman, Ruchi Kumar, Aditya Chand, Subhasri Duttagupta, G. V. Raja Kumar and Chaitra Rao, *aAQUA - A Multilingual, Multimedia Forum for the community*, IEEE International Conference on Multimedia and Expo, 2004.

[2] Sarvjeet Singh, *Meaning Based, Multilingual Search Engine*, B. Tech. Thesis at IIT Bombay, 2003.

[3] Hiroshi Uchida, Meiying Zhu, and Tarcisio Della Senta, *UNL, A Gift for a Millennium*, UNU Institute of Advanced Studies, 1999.

[4] Hiroshi Uchida and Meiying Zhu, *EnConverter Specifications*, UNL Center, UNDL Foundation, 2000.

[5] *Phrase structure grammar for question.*. Available: http://www.scientificpsychic.com/grammar

[6] *Link Parser*. Available: http://www.link.cs.cmu.edu/link

[7] *Link Parser's Application Program Interface*. Available: http://www.link.cs.cmu.edu/link/api/index.html

[8] *Brill's Parts of Speech Tagger for English*. Available: http://research.microsoft.com/users/brill

[9] Steven E. Weisler and Slavko Milekic, *Theory of Language*, MIT Press, 2000.

[10] Wren and Martin, *High School English Grammar and Composition*, S. Chand, 1989.

[11] Damle, Moro Keshav and Arjunwadkar, Krishna Shrinivasa, *Shastriya Marathi Vyakarana*, Deshmukh and Company, 1970.

[12] Rohini Srihari and Wei Li, *Information Extraction Supported Question Answering*. In Eighth Text Retrieval Conference, 1999.

[13] Eric Brill, Susan Dumais and Michael Banko. *An analysis of the AskMSR Question Answering System*, 2002.

[14] Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. *FALCON: boosting knowledge for Answer Engines*. In proceedings of Ninth Text Retrieval Conference, pp. 479-488, 2000.

[15] Guru, Kamtaprasad, *Hindi Vyakarana*, ed. 22. Nagaripracharini Sabha, Varanasi, 1979.