A Prototype of Personalized Question Answering System for Restricted Domain
This chapter presents the complete architecture of our proposed QA system underlying on both QA corpus and Web as information resource which involves three main phases of question processing, information processing and answer extraction that characterize most state-of-the-art QA systems.

7.1. Introduction

In recent past, increasing amount of data on Web has created a requirement for the mechanisms that help user to opt out of the irrelevant information efficiently. One of such popular phenomenon is question answering (QA), the retrieval mechanism which provide a precise answer of question instead of ordered list of documents. Given a question such as ‘Which university has highest ranking according to NIRF in 2016?’ a search engine present the user with pages from MHRD where as a QA system would try to directly present the answer with the name of an institute. On the Web a typical example of QA system is Wolphram Alpha, which allows users to ask questions in natural language. The system looks for the answer in its repository and returns the answer. But the system is more accurate in answering mathematical questions as build upon the tool Mathematica. Therefore a rational intention for an automated question answering is to present a string of answer instead of a list of documents. In this chapter we present a question answering system which combines shallow natural language processing (NLP) techniques, pattern learning and information retrieval techniques to provide response to the user’s question in real time. The QA system is designed to answer questions in higher education domain.

The major contribution of this chapter is an integrated framework for question answering. The QA system makes use of pattern learning in several parts of the processing. Patterns are used in (a) classifying questions (b) focus word identification (c) reformulation of question to query and (d) at answer extraction phase. The initial three tasks are performed using a pattern corpus for questions which aims to identify most sought pattern for a user question and generate its question class, focus word and reformulated query. The latest task, i.e., answer extraction requires implementation of another corpus which contains query and its respective answer. The user’s reformulated query pattern is matched against queries stored in the corpus according to its answer class mention by Singh et al.,(2015). However, we have also used SVM for question classification and structural query
reformulation for complimenting the pattern learning approach at question classification and reformulation phase respectively.

The chapter is organized as follows. Section 2 describes the general architecture for question answering system. Section 3 presents a comprehensive detail of the different components integrated to the QA system such as question classification, reformulation, learning patterns and answer extraction using similarity metrics and library of patterns. An evaluation is also presented using a manually constructed TREC like question collection for higher education domain as standard evaluation question set are not available for the same. Next section presents the related work to the QA system. Finally, section 6 presents the conclusion and future scope.

7.2. Proposed Method and System Architecture

The general architecture of QA system provides a framework which integrates shallow NLP techniques, pattern learning, query reformulation and information retrieval. The architecture is illustrated in Figure 7.1. This architecture has three phases: question processing, query similarity evaluation and extraction from corpus and answer extraction from Web. Each of these stages is described as:
7.2.1. Question Processing

Question processing is carried out to synthesize the question posed by user. This ‘synthesizing’ of the question requires several steps such as shallow linguistic processing of question, classification, reformulation and named entity or domain word recognition. This phase uses some components which are described as:

- **Pos tagger and Chunker**: The pos tagger tags the question according to Penntree bank set (Marcus et al., 1993) into noun, verb, adjectives etc. The chunker tries to chunk noun and verb tags to produce noun phrase and verb phrase respectively.

- **Pattern Corpus**: A pattern corpus is designed to accumulate most frequently occurring question patterns to be used in classification, focus word identification and reformulation. The structure of the pattern corpus has been discussed briefly in Chapter 3.

- **Taxonomy**: Question class taxonomy is presented for higher education domain which includes 9 coarse classes and 64 fine classes (Dwivedi et al., 2014). The taxonomy is crafted using generic terms. Answer class taxonomy is also presented for local information resource i.e., Question Answer corpus which has 8 classes.
based on domain specific words. This taxonomy helps in locating answer class thus reducing search space.

- **Related word and Domain Knowledge:** A manually constructed set of related word is employed during classification phase. Domain Knowledge is used to represent valuable domain specific information separately such as university name, designations, courses and their abbreviations, scholarships and many more at different stages of QA processing.

Usually, in restricted domain, a user put a question which comprise of specific terminologies. However, for education domain QA system, the user will not only pose specific terminologies but also so many abbreviations as name of courses, designation, subjects etc. for ease. Therefore, to use the domain knowledge efficiently and express the answer space distinctively, we have developed a dictionary specific to Higher education domain which includes domain word type as illustrated

- **WordNet:** QA system’s lexical resource.

With the help of above components, the module of question processing initially starts with question classification which categorizes questions as belonging to any of the coarse question class supported by our QA system (what, who, when, which, how, where and why) and then to any of the 63 fine classes. The primary classification if achieved using pattern matching will also accompanied with focus word. Further, a secondary classification is carried out to identify answer class to regulate answer search in QA corpus. The query reformulation is the next task to be done by the question processing phase. The query reformulation from question to query is also first attempted to be acquired using pattern matching but if failed to do so we would use structured reformulation to get the required query. The system then aim to named entity and domain word recognition and proceed to next stage of QA system.

For instance different aspects for the question, ‘What is the age limit for JRF applicant in UGC?’ can be retrieved from the snapshot shown in Figure 7.2 picked from question pattern corpus.
Figure 7.2 Snapshot of referenced pattern from Corpus

The question class is ‘Number’ and focus word is ‘age limit’.

Reformulated query from pattern matching module is ‘age limit for JRF applicant in UGC’

Answer class: Fellowship & Scholarship

7.2.2. Retrieved Information processing:

The reformulated query from previous module is posed to the QA system and effort has been made to extract relevant information either in form of direct answer from QA corpus or from Web. A QA corpus is essential to this stage and can be described as:

- **QA corpus**: The corpus is key to our proposed QA system. It is used as direct local information resource which can be easily referred while answering such questions which has less changing or static answers. The corpus contains query and their respective answer along with annotated snippet. (details in Chapter 3)

In view of the fact that, we have to extract information from either of the two resources; our information processing can be divided into two steps:
7.2.2.1. Answer extraction from QA corpus

After obtaining the reformulated query from previous step, the pattern matching procedure is used to find the question patterns which best match with the patterns stored in question answer corpus. The similarity evaluation procedure includes following steps:

Step 1: Word order similarity

This step is used for matching and refining all patterns which are relevant to the user question that has been obtained from previous sub step. The pattern with highest matching score is considered as the candidate pattern for answer extraction from local resource.

The user question transformed to query is initially considered as set of words and these words along with their order of occurrence are used to match with all queries stored in question-answer pair corpus. The word order similarity between user’s reformulated query and candidate questions in QA corpus has been calculated by the formula presented by Li et al. (Li et al., 2006) in Equation 5.2.

Step 2: Semantic similarity calculation

This is a feature that indicates whether a word and its synonym appear in a query candidate and a user’s query respectively. We use WordNet to determine semantic similarity by the Wu & Palmer’s Measure (1993) given as Equation 5.4:

Therefore, the total similarity score depending on the individual influence has been calculated as:

$$ Siml(Q, C) = \mu_1 W_i(Q, C) + \mu_2 S_i(Q, C) \quad (7.1) $$

Where $\mu_1 + \mu_2 = 1$. Apparently $0 \leq Siml(Q, C) \leq 1$. The similarity between query and answer is decided together by the above factors, the influence of $W_i(Q, C)$ is dominating, while semantic similarity $S_i(Q, C)$ has relatively less impact. According to the testing results to some queries, the rudimental parameters have been defined as: $\mu_1 = 0.7$, $\mu_2 = 0.3$.

A set of similarity measures applied to find the most resembled query in the QA corpus. These metrics rely on syntax and semantics of the query itself and different scores are calculated. If the combined score of similarity metrics is above the threshold value then
the answer for candidate query having highest score is extracted to give the response to
the user’s question. If none of the candidate query resemblance score is above the
threshold value then our system proceeds to the next phase.

7.2.2.2. Web answer Extraction

In this phase answers are extracted from Web resource. Search engine i.e., Google is used
to access the information from Web to bring down the list of documents. Different
metrics are used to refine the candidate answers from snippets of the listed documents
including the rank of the same. Finally, the answer having highest score is selected as the
answer for user’s question.

In addition to the similarity measures used by the system while retrieval from QA corpus,
Web based answer extraction module estimates two more similarity parameters. The first
parameter is the document or snippet rank $R_i(Q, Snip)$ as retrieved by the search engine
and second one is similar text pattern count $C_i(Q, Snip)$ across snippets as discussed in
chapter 5. The total effective similarity $S_i(Q, Snip)$ has been calculated as the weighted
sum of all these similarity measurements based on their individual impact as shown in
Equation 5.5.

7.2.3. Personalization

This phase particularly belongs to the personalization of proposed standard question
answering (QA) system according to the user’s interest. The personalization of the
retrieved data from standard QA system is done using implicit user information and
interest area. It is based on building a conceptual user profile and implementing it in re-
ranking of the search results. The user profile comprise of concepts obtained from
organized classification of previous transactions based on different metrics. Re-ranking
of the search result has been performed using several similarity metrics based on
attributes and values. These metrics consider both semantic and syntactic user
information. (Chapter 6)

Integrating all of the above modules to develop a unified framework of personalized
question answering system can be summarized as Algorithm 7.1. The algorithm provides
The complete workflow of our proposed personalized question answering system as shown in Figure 7.3.

Algorithm 7.1: QA Algorithm

- **Question Processing**
  1. Pre-Process user’s question q using POS tagger and chunker.
  2. The POS tagged pattern of q is matched against candidate patterns q\_c from pattern corpus.
  3. If, similarity coefficient is greater than threshold value λ, extract coarse & fine grained class, focus word and reformulated query q\_r.
  4. Else, apply generic pattern matching to filter first head noun and coarse class.
  5. Implement SVM to extract coarse & fine grained class.
  6. Identify answer class implementing secondary classification.
  7. Apply SQR to fetch the reformulated query q\_r.

- **Information Processing**
  8. Pose the reformulated query q\_r to Question-Answer corpus
  9. Calculate Query Answer similarity
  10. If similarity coefficient is greater than threshold, extract top k answer list
  11. Else, pose query to search engine & Extract snippet list.
  12. Calculate query snippet similarity

- **Answer Extraction**
  13. Answer list/ Snippet list are matched with the topics in User’s profile.
  14. Candidate answers are extracted and ordered by relevance to the query;

---

**Figure 7.3** Question Answering Algorithm

### 7.3. Automated pattern learning

This is another aspect of our proposed question answering (QA) system. We have explicitly embedded snippets annotated with focus word and answer along with query-answer pair in QA corpus. The snippets are attached to facilitate automated pattern learning for question pattern corpus by inclusion of new generic patterns. The pattern
learned from the annotated snippets is added to generic pattern list of question pattern corpus along with the coarse class.

Here, we give an example to illustrate how question pattern learning works. The annotated snapshot of snippet for our QA corpus can be illustrated as shown in Figure 7.4 Focus word (if exists) and Answer string are tagged as FW and AW respectively along with the POS tags.

```
<ID> 314</ID>
<QUERY>
BMU establishment year/BHU established in year/Founder year of BMU/BHU was founded in year
</QUERY>
<ANSWER>1916</ANSWER>
<SNIPPETS>
Established_VBN in_IN 1916_CD (AW) by_IN [Pandit_NNP Madan_NNP Mohan_NNP Malaviya_NNP], BHU_NNP is_VBZ one_CD of_IN [the_DT largest_JJS residential JJ universities_NNS]] in_IN Asia_NNP, with_IN over_IN 20,000_CD students_NNS .._.
</SNIPPETS>
```

Figure 7.4   Answer annotation example

Here, we will use only POS tagging information to generate new patterns as shown in Figure 7.5. The snippet is however chunked to have more generalized heuristics. Thereafter the snippet is labeled with focus word and Answer string as shown in Figure 7.6

```
<SNIPPETS>
VBN IN CD (AW) IN NNP NNP NNP NNP, NNP VBZ CD IN DT JJS JJ NNS IN NNP, IN IN 20,000_CD NNS .._.
</SNIPPETS>
```

Figure 7.5   POS tagging

```
<SNIPPETS>
VBN IN CD (AW) IN NNP, NNP VBZ CD IN DT NNS IN NNP, IN IN 20,000_CD NNS .._.
</SNIPPETS>
```

Figure 7.6   Chunked generalized pattern with answer annotation

Hence, we can obtain a training sample as illustrated above in example. The above synthesis gives a new pattern annotated with answer string and focus word.
7.3.1. Assessment of extracted pattern

Obviously, the above process of automated answer pattern extraction can result in large number of patterns and hence poor response time. The approach judges the answer pattern to be added in question pattern corpus and drops too specific ones. For verification, we will randomly consider at least 5 verified query-answer pairs from QA corpus having same question class as the candidate pattern has. The approach then poses the queries to the search engine, fetches snippets and applies the learned pattern to fetch the respective answer string. The approach records how often the new pattern could be used to extract a right answer \( (A_r) \) and how often it extracted a wrong answer \( (A_w) \). Thereafter, we calculate their precision and recall. Then patterns having precision above 0.6 are selected to be included in question pattern corpus. Therefore, the new pattern obtained this way will be added as generic pattern in question pattern corpus to enhance the overall QA processing. However, idea need to be more enhanced to include specific patterns automatically to question pattern corpus and can be considered as future work.

7.4. Summary

Summarizing, in this chapter, we describe the overall working of our proposed question answering system for restricted domain. Briefly, the questions in English are first classified to its respective question class and focus word is detected using pattern matching and SVM classification technique. A secondary classification mechanism has also been applied but this time to identify answer class of the local answer resource i.e., QA corpus which contains query and answer pair. The secondary classification reduces the answer search space with respect to the QA corpus. The QA system then looks for reformulated query which is again performed using pattern matching technique. But this time the reformulation mechanism is complimented using structural reformulation technique. Further, the QA system uses similarity metrics to extract answer from local answer resource and from Web as well. The task of automated generic question pattern generation has been also performed using the same QA corpus.