SOFTWARE REQUIREMENTS SPECIFICATION

FOR

MULTI-DOMAIN QUESTION ANSWERING SYSTEM

PREPARED BY

KRISHNA PUROHIT	422030
AKASH VARMA	423064
AANSHI VERMA	422068
SHAAZ SHAIKH	422048

UNDER THE GUIDANCE OF

MS. P. D. MORE

BRACT's

VISHWAKARMA INSTITUTE OF INFORMATION TECHNOLOGY, PUNE.

Table of Contents

1. REQUIREMENT ANALYSIS

● Introduction

Question Answering (QA) system in information retrieval is a task of automatically answering a correct answer to the questions asked by human in natural language using either a pre-structured database or a collection of natural language documents. It presents only the requested information instead of searching full documents like search engine. As information in day to day life is increasing, so to retrieve the exact fragment of information even for a simple query requires large and expensive resources.

Now days there are many search engines available. All these search engines have great success and have remarkable capabilities, but the problem with these search engines is that instead of giving a direct, accurate and precise answer to the user's query or question they usually provide list of document related to websites which might contain the answer of that question. Although the list of documents which are retrieved by the search engine has lot of information about the search topic but sometimes it may not contain relevant information which the user is looking for. Also, chatbot are thought of as emerging technique to solve this issue, which are designed to be more natural as interactive than conventional search engines, which aim at making the user interested by engaging with them in a conversation. But the subtle limitations of natural language restrict chatbots from being general purpose.

Therefore, it is a necessity to develop an application which can have the accuracy, nature and conciseness of a chatbot, but can be expanded to multiple domains, if not all, like a general purpose search engine.

● Purpose

As we are acknowledged from the introduction about a general purpose question answering system, the proposed software application focuses on solving above issues by absorbing useful features of both, specific purpose chatbots, and conventional general purpose search engines.

The main aim of question answer system is to present short and concise answers to user query instead of searching list of document related to search topic. Users just have to ask the question and the system will retrieve the most appropriate and correct answer for that question and it will give to the user. In this way the user will not have to spend a lot time going through different documents online and get the accurate answer in one place.

● Intended Audience

This will be for general users who need accurate answers instead of links when they type a query . While working on normal search engines for e.g. google, people get to accurate answers only sometimes. But most of the time they recieve links in ranked order to search for the answer they need. They need to search multiple links for answer which is inefficient. In order to avoid this problem we will present user the accurate answer.

- **Existing applications**

As discussed in prior points, many existing systems provided answers to most of our daily life questions. A conventional solution is search engines, which give us links to all the relevant web pages. Search engines are by far most widely used websites on the internet. But with increasing innovation and prevailing limitations of search engines, people started developing domain specific chatbots for their websites, which would give whole website a personalised flavor, and make the user feel more natural and friendly with the application. But again, chatbots come with their own limitations and natural language has its own subtle problems which cannot be solved by pure mathematics. Such problems include context resolution, understanding figures of speech like irony, satire, metaphor, etc. These problems with natural language limit the accuracy of chatbots over a wider domain, that's why most chatbots present on websites are limited to a very narrow scope.

2. FUNCTIONAL REQUIREMENTS

- **Project perspective**

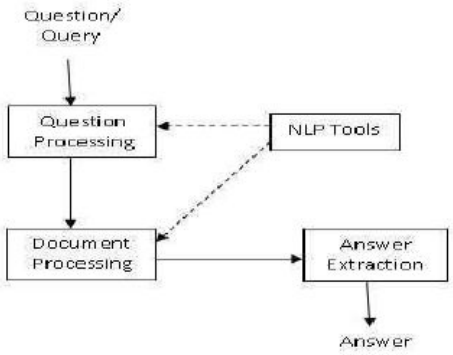
An Automatic Question Answering System (AQAS) is a platform which allows a user to input his question and makes use of information retrieval techniques and natural language processing to compute and present the user with the best possible answer. This system should give the user the impression that he is talking to the machine and for this to be possible the question should be posed in natural language.

The QA systems can be broadly divided into two types:

- Closed-domain question answering: Closed-domain question answering refers to specific domain related questions and can be seen as an easier task because NLP systems can provide domain-specific knowledge. It has very high accuracy but limited to single domain.
- Open-domain question answering: Open-domain question-answering deals with the questions which are related to every domain. In this systems, mainly have more data available from which the system extract the answer. It can answer any question related to any domain but with very low accuracy as the domain is not specific.

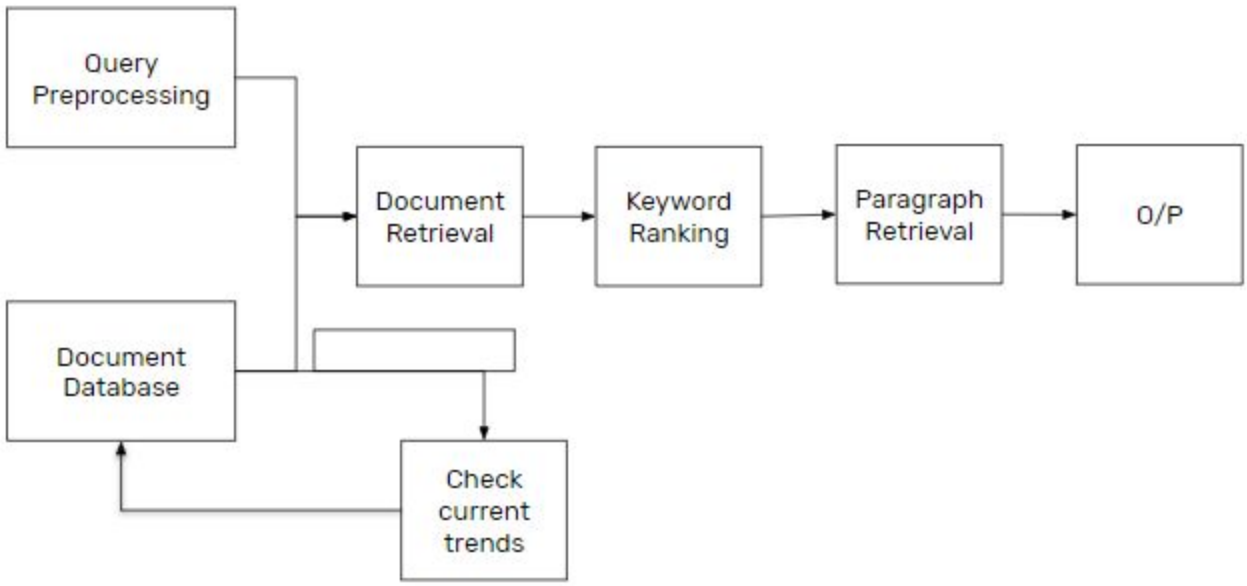
There are different projects on closed domain as well as open domain QA systems. But here the problem with open domain systems is low accuracy in dealing with different types of questions and in case of closed domain systems if the data in the knowledge base is outdated the QA system becomes obsolete .Here we aim at building a dynamic system which changes its knowledge base as per the current trends which is not the case with closed domain QA systems. It will give us a higher accuracy as well as it will not become useless in future .Here we consider some high level domains as politics, education, business, sports and entertainment. The knowledge base will have knowledge about the current trends in these domain.So knowledge base in of this system will consist of data about the current trends , that will make it more useful for the users , hence it will prove to be of greater advantage over the existing systems.

High level diagram is as follows :



- ## Modules of system

The functions this system will offer is that, it will give the answer to the query posted by the user as an input to the system. After after this the query will be processed and used to search the knowledge base and extract all the possible answers for the query.
The basic architectural diagram of the system is as shown below:



Basic steps involved are :

- **Query Preprocessing**: - This is the first step in QA system in which input to the system is question ask by user in natural language, the overall function of this module is to process and analyze the input question.

- **Query Classification**: - Here we will classify the query and identify that the query belongs to which domain as we have multiple broad different domain .This will facilitate the answer retrieval process.
- **Database Search**: - Here the search of the possible results is done in stored database, the related data that satisfy the given query with selected keyword are sent to the next process
- **Related Document**: - The result which was generated by the previous stage is stored as a document.
- **Answer Display**: - The result is stored as a document. Then the result is converted into accurate text for that the user is looking for and that answer displayed to the user. The front end is a Web Application which will be used by the users to interact with the System. Django framework will be used to design this web application.

● User classes and characteristics

There will be a general user class

Characteristics of user- User can ask question on any domain and accurate answer will be provided to the user. User will be able to see the current trends. If user asks question out of current trends then user will be provided with appropriate links for the same.

● Design and implementation constraints

The main constraint is that the knowledge base will be limited to current trends and if the user posts certain question that is not relevant or is not currently trending then the user may not get the answer with the desired accuracy. We cannot completely move to open domain systems because the accuracy is not great as systems with specific domains have. Also in open domain the time required to gather the information runtime will increase the processing time and if we won't have the correct information the answer won't be accurate so extending it to open domain will not be feasible.

As far as irrelevant query of user is concerned we will extract the answer from google at run time and we plan to display the top relevant links obtained from there to direct the user to some relevant answer.

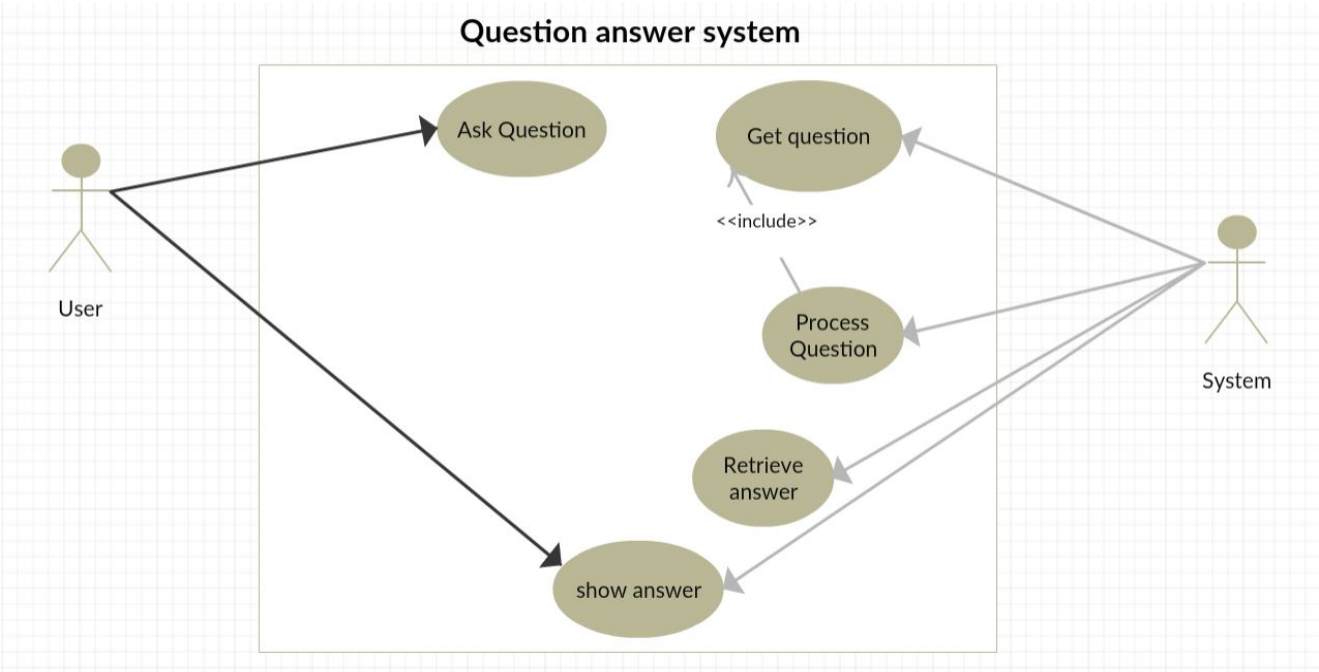
● Assumptions and dependencies

In this QA system we extract the information about the current trends from websites, we assume that the information that we want regarding a specific topic will be available at those websites. The websites must have specified information for the knowledge base to be constructed dynamically and with most accurate information which will be useful to answer the query.

3. EXTERNAL INTERFACE REQUIREMENTS

- **User interfaces**

  There will be graphical user interface which will show current trends. User will be able to ask question and expect answer in answer panel. Django web framework will be used to design user interface. It is written in Python, which follows the model-view-template architectural pattern.



Multi domain Question answer system use case diagram

- **Software interfaces**

  In software interface we would have a user interface designed in HTML, CSS and javascript which will consist of a text box from where a users question would be taken and a submit button for searching the answer in the database. In our system we will be using SQLite database which will be constantly updating itself constantly after a certain time for getting the latest news.

- **Communications interfaces**

  In question and answer system we need to have an active internet connection. We require internet connection to get the latest news, so for this we use scraping which requires active internet connection. The task of updating database at regular interval from trusted websites pre-selected for each domain will require an active internet connection. Other than this, no other type of communication will take place and everything will be carried out on server.

A client can access server through django webapp to use the graphical user interface of question answering system.

4. TECHNICAL SPECIFICATION

- **Operating environment**

We will use ubuntu 16.04/18.04 operating system. Our coding will be in python and we will use spacy library for natural language processing. Database used will be SQLite to store metadata and path for json file. Json file will store data regarding particular content. Django will be used as a framework to contain whole application. Since the proposed application is a web application, the user interface will be designed in HTML, CSS, and Javascript.

- **Programming languages and technologies**

1. Python

Python will be used to write programming logic for question answer system.

2. Django

We will be using django web framework for python as a container for our application. Since the software will be programmed as web application, user interface will be designed using HTML, CSS and Javascript. Django is based on MVC architecture, which makes it easy to develop frontend, backend, and middleware part in parallel since all 3 components are loosely coupled. Also, it supports object relational modelling (ORM), which will allow us to create and maintain database easily.

3. Database

Database used will be SQLite to store metadata and path for json file for retrieving answer. It will be updated under some specific time so that user will be able to get information about current trends in exact manner. Information will be divided in different files and in a particular file contents related to some specific topic will be stored.

Metadata will contain information like from which link the data was taken and when was it taken.The database will contain keywords and path of relevant documents. It will stored in the format as, keywords and the corresponding path of document which contains the keywords .

4. Wordnet

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations amongst these synonym sets or their members.

5. Spacy

SpaCy is used to prepare text for deep learning. It interoperates seamlessly with TensorFlow, PyTorch, scikit-learn, Gensim and the rest of Python's AI ecosystem. With SpaCy, you can easily construct linguistically sophisticated statistical models for a variety of NLP problems. SpaCy uses an object-oriented approach. Parsing some text returns a document object, whose words and sentences are represented by objects themselves. Each of these objects has a number of useful attributes and methods, which can be discovered through introspection.

● Performance constraints

Creating a question answer system would be only meaningful if it works ideally in real time or with a very low latency and processing time. To achieve this, it is really important to maintain database and process the question such that it requires very less time to generate a concise answer.

Following points discuss the performance requirements of our proposed project:

➔ **Creating and maintaining dynamic database:**

The application proposed focuses on answering multiple domain questions concisely, but to put an upper limit, the question answering system will only answer questions related to trending topics in various pre-selected domains, which include politics, business and economics, sports and entertainment. To handle this huge variety of data altogether is not possible. Also, mining relevant articles while searching an answer for a given question seems too time consuming. To solve the above problems, we'll automate the task of updating database using text mining from trusted websites which will be predefined for each domain. A simple algorithm will be used to check whether there are enough new trending topics to replace the older trending topics currently existing in the database.

➔ **Processing question:**

Before directly answering the question, the question will be preprocessed and metadata about question will be stored in a format which is appropriate for generating answer from relevant answers. This process of generating metadata related to question which will describe the domain of the question and type of question should be carried out as efficiently and fast as possible. The precision of this metadata will ultimately define the quality and conciseness and completeness of the answer generated. This is why generating precise metadata about the question will play a crucial role in the performance of software overall.

➔ **Generating answers:**

On the other hand, after the metadata of question is generated, it will be used to generate answers. It will require some essential information about data which will help the software in generating the answers. First, the software needs to figure out to which domain is the question related to, in order to find relevant articles from the database.