



A study on different closed domain question answering approaches

Srinivasu Badugu¹ · R. Manivannan¹

Received: 7 October 2019 / Accepted: 21 February 2020 / Published online: 11 March 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract

Question answering (QA) framework is a framework that gives answers to the inquiries raised by the client using the common language. The framework recovers minor portion of the content from the collection of the report, which contains the appropriate response for the client's inquiry. In order to retrieve such response from the repository, information retrieval techniques are needed and for further processing or comprehension of the client's inquiry, presented in the characteristic language, natural language processing techniques are utilized. However to make the recovering procedure increasingly hearty, snappy and accurate, the idea of knowledge-based classification also included in this work, for this reason, utmost care was taken in training the framework. using "Jaccard likeness", the closest answer for the client's inquiry was reached. In addition to this, "WordNet" was used to recover the appropriate response, depends on both syntactic and semantic similitudes. Utilizing these ideas we have actualized a QA framework on space "Hyderabad Tourism" which gives in general exactness of 92%. In this work, our main aim is to create a closed-domain question answering framework, which will give the precise and considerably short answer to all the inquiries that are related to the Hyderabad city, as a response, instead of giving a lengthy paragraph or document.

Keywords First keyword · Information retrieval · Knowledge-based classification · NLP · WordNet

1 Introduction

Question answering (QA) frameworks is a software engineering discipline that has a place with the fields of data retrievers recovery and characteristic language handling (NLP) (Ostapov 2011; Russell and Norvig 2016; Bishop 2006). These frameworks are worried about the making of the frameworks that can consequently respond to the questions presented by the people in a characteristic language (Cimiano et al. 2014). Question Answering System is a multidisciplinary field that implies, it is an accumulation of a few scholarly orders [for example, artificial intelligence, natural language processing (NLP) and information retrieval (IR)]. A question answering (QA) usage is a PC program, that develops the necessary answers by questioning or checking

an organized database of learning or data, which is called as an 'information learning base'. Generally, question answering (QA) frameworks can recover answers from an unstructured accumulation of common language documents (Ostapov 2011; Mishra and Jain 2016; Kaushik 2011)

Question answering (QA) research is worried about a broad scope of inquiry types including certainty, definition, list, why, how, semantically compelled, speculative, and cross-lingual inquiry. In any case, there are many web indexes available (Ferret et al. 2002; Marietto et al. 2013; Meadow et al. 1992). All these web indexes have incredible achievement and have amazing abilities. Still the issues with these web crawlers are that as opposed to furnishing a direct exact and exact response to the client's inquiry or question. They generally give rundown of record identified with sites which may contain the appropriate response of that question. In this way, so as to accomplish the necessary data or answer, the client need to experience all the website, report or document recorded by the internet searcher.

The following points focuses portray why it is critical to make a question answering system (QAS):

✉ Srinivasu Badugu
srinivasucse@gmail.com

R. Manivannan
drmanivannan@stanley.edu.in

¹ Department of Computer Science and Engineering, Stanley College of Engineering and Technology for Women, Chapel Road, Abids, Hyderabad 500 001, India

- This framework can diminish the time and exertion required for choosing an exact answer from the outcome given by the web index.
- The fundamental issue with these web crawler is that, as opposed to offer an exact response to the client's question. They more often than not give a rundown of archive identified with sites, which may contain the appropriate response, and expected the client to experience that outcome as the suitable answer.
- Looking inside the FAQs can be a dreary errand.

In each site, there will be an area called FAQ, in which a rundown of expected inquiry a client can have about the item or site alongside its answer is put away. If a client has any inquiry then they have to got through these FAQs segment to find its necessary solution. So, on the off chance that we supplant this FAQ segment with the QAS model, then it will be simpler for the client to type the inquiry as opposed to look through it in the rundown.

2 Literature survey

In Lende and Raghuvanshi (2016) has proposed a “Shut area question noting framework utilizing NLP systems” In Lende and Raghuvanshi (2016) authors have made a QAS on “Instruction act”. Authors have gathered the corpus about the Education Act in the UK. After that, authors performed preprocessing on the corpus in order to gather the watchwords using TF-IDF they have made a “List Term Dictionary”. When a framework gets an inquiry it will again perform preprocessing on it to gather significant catchphrases from the Question. Presently, the subsequent stage is to concentrate archive, which may contain the necessary response. For this they will recover the records, where the entire inquiry watchword is available. Then dependent on Jaccard closeness it will rank the report, at long last dependent on POS labeling they are recovering the Answer.

In Bhardwaj et al. (2016) talks about a question answering Framework for Frequently Asked Questions in which they have made an Open space question answering framework. Authors utilizes FAQs to respond to the client's inquiry. They have executed their methodology utilizing QA4FAQ from the site which presents in the CSV group as their data set. In Ferret et al. (2002) authors were proposed a methodology which joins two procedures. For example, Orthodox AND/OR looking with Combinatorics scanning for looking through the client's posted inquiry in the inquiry answers pair rundown store as their data set to recover the appropriate response as for the given inquiry is concerned.

In Fu et al. (2009) Introduce a QA framework on Music utilizing Database Ontology Knowledge where a client can pose any inquiry about music. In Fu et al. (2009) the creator

has offered two ways to deal with recover a response as follows: (i) FAQ module (ii) Ontology Knowledge. When a framework gets a client's inquiry it will initially look in FAQ module and in the event that it presents. At that point the combined answer with the coordinated inquiry is extricated and return to the client, else the framework will go for the second technique where it needs to examine the philosophy learning base for the suitable answer. For that it plays out the following advances: Question classification, Question Analysis, and Answer Extraction. The creator in Fu et al. (2009) has expressed that the main methodology which is FAQ has given great outcomes than the subsequent methodology.

In Bhoir and Potey (2014) have examined about “Question noting framework: A heuristic methodology” This is a shut space question answering System whose chose area is “Tourism”. For that, the Authors used the data collected about Pune, the travel industry as their corpus.

In Bhoir and Potey (2014) author has made after stride where right off the bat they have made a web Crawler utilizing Java and to that, they have given a rundown of the site, which contains the data identified with Pune. After that the data gathered by the crawler is preprocessed to get the catchphrase, when the QAF get a client question, that question will likewise get preprocessed to evacuate stop-word or noise and just significant watchwords ought to be there. After separating catchphrase from both inquiry and corpus, authors used the idea of Procedure programming language to recover the last answer. In request to surpass the time required for recovering the appropriate response, authors have additionally used the idea of ace sentence where from the information, the sentence wherein there is any number pursued by “km”, “miles” and so on will be considered as ace sentence on the off chance that the inquiry is identified with the separation, at that point the appropriate response will be return more precisely.

In Pragisha and Reghuraj (2014) authors have proposed their question answering system in the Malayalam Language which means their framework can address any inquiry posed by a client in Malayalam. This is a shut space QA framework where their area is Kerala sport. For this authors have put away gathering of Malayalam record about Kerala Sport as their data set. The usage of this framework starts with the Question Type Analysis module, where they distinguish the Malayalam question word and their importance. Next step is to perform Document Processing, for this Sentence tokenizer was used to part the record into sentences and put away in an exhibit, Then positioning of the sentence takes place depends on its likeness with the client's inquiry, later the beat rank sentences called as Answer Candidates are chosen. On that Answer up-and-comer, the assignment of Name Entity Recognizer is performed utilizing the TnT tagger. Finally, the last advance is Answer Extraction is where we recognize the normal tag of the inquiry word and that tag

is considered as the Answer key. Now if the Name Entity or Tag of Answer Candidate matches with the Answer key, then the Answer Candidate with most extreme matches are considered as the Final Answer.

In Sahu et al. (2012) authors have proposed a Question-Answering framework, which gives answers to the inquiries posed in the Hindi Language. The authors used a gathering of Hindi record about some particular theme. Initially authors have changed over the given client's Hindi inquiry in Query Rationale Language (QLL), which is a subset of Prolog utilizing created rules. This inquiry is sent to the database, which cross examines the put away data to separate an answer, which they will convert into Hindi before sending the appropriate response back to the client.

In Han et al. (2006) authors used a probabilistic model, consisting of three parameters, namely: (i) topic model (ii) definition model (iii) sentence (language) model. The goal is to find the probability that a sentence is a definition of the given topic (target) ($P(D, S | T)$).

In Sun et al. (2018) authors proposed a novel graph convolution based neural network, called GRAFT-Net (Graphs of Relations among Facts and Text Networks), specifically designed to operate over heterogeneous graphs of (Knowledge Base) KB facts and text sentences. To answer a question posed in natural language, GRAFT-Net considers a heterogeneous graph constructed from text and KB facts, and thus can leverage the rich relational structure between the two information sources.

In Hao et al. (2017) authors proposed a novel cross-attention based neural network (NN) model tailored to knowledge base question answer (KB-QA) task, which considers the mutual influence between the representation of questions and the corresponding answer aspects. The model leverages the global KB information, aiming at represent the answers more precisely.

In Cui et al. (2019) authors proposed a template representation model, whose task is to map the given question to the existing template question. To do this, the entity in the question is replaced by its concepts. This process is not trivial, and it is achieved through a mechanism known as Conceptualization. Using templates, a complex question can be decomposed into a series of question, each of which corresponds to one predicate.

3 Proposed work

A QAF can be created for any specific domain such as Education, Sports, Movies, Politics and Healthcare etc. Based on the literature survey conducted, classified categories are “List-based method” and “Retrieval-based method”. To get a proper understanding, an attempt was made to make a QAF to overcome the drawbacks in those two approaches and proposed a new method to extract correct response from a corpus using Rule-based Classification and Similarity Measures. Since there is no QAF for exactly answering the queries on the city “Hyderabad” (in the state of Telangana, India), which ensures the correct answers about history, famous monument, lakes, Amusement park of Hyderabad. Therefore the idea for developing a question answering framework on tourism of Hyderabad was proposed. The main goal of any QAF system should give short and accurate answer, which is nothing but the fine-tuned version of Information Retrieval System.

3.1 List based methodology

Question answering framework utilizing Rule-based methodology is an exceptionally straightforward methodology, wherein the accumulation of Question and Answer sets are gathered and put away as the data-set. Where Answers are created by basically looking through the offered inquiry from the put away Question Answer pair in the event that the inquiry is coordinated with the inquiry answer pair, at that point its individual answer is offered back to the User as a last Response. To make a Question Answering framework dependent on this methodology, the idea of AIML was used. AIML represents Artificial Intelligence Modeling Language. AIML is a XML based markup language intended to make counterfeit canny applications (Bird et al. 2009; Mishra et al. 2010).

First, as the main entry point for loading AIML files, it is standard to construct a start-up file called std-startup.xml. In this work, a simple file was built that fits one pattern and takes one action. In this case, match the pattern load aiml b, and have to load our brain in response.

3.2 Sample AIML file

```
<aiml>
<category>
<pattern>Define Artificial Intelligence </pattern>
<template>
Artificial Intelligence (AI) is a computer science sub-field, and
it is the science and engineering of making intelligent machines,
particularly intelligent computer programs.
</template>
</category>
</aiml>
```

Basic python code was used to construct a target object. This object learns the initialization file and then the rest of the target files are loaded. Now, it's ready to work, so we enter an infinite loop that will continue to ask the user for a response (Perera 2012; Unger and Cimiano 2011).

In the information retrieval approach, the key approaches are

1. Preprocessing
2. Vector creation

3.3 Basic code for creating AIML files in python

```
import os
import sys
# Create the part and learn AIML documents
part = aiml.Kernel()
kernel.learn("std-startup.xml")
kernel.respond("LOAD AIML B")
# Press CTRL-C to break this circle
while True:
    print (kernel.respond(input("Enter your message >> ")))
```

3.4 Information retrieval approach

In Retrieval based models, the framework utilizes some IR-Techniques to choose a legitimate reaction from a put away corpus. The developed QA framework uses the Python library like "Scikit-Learn" (Unger and Cimiano 2011; Yang et al. 2015) and extraordinary apparatuses like "NLTK" (Höffner et al. 2017).

3. Extracting answers

The algorithm used to pre-process that document in the corpus.

Algorithm 1: Algorithm for Preprocessing

```
Input: Input: Raw Text D;
Input: Output: Tokens T;
if  $D \neq$  then
    | Perform sentence segmentation;
    | perform tokenization;
end
for all tokens in D do
    | if token is not in stop words list then
    | | perform stemming;
    | end
end
```

3.5 Preprocessing

Preprocessing refers to the transformations applied to crawled data before feeding it to the algorithm. Whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. Therefore it needs preprocessing. In other words, Preprocessing is a technique that is used to convert the raw data into a clean data set.

3.6 Vector creation

Maximum tokens we found in all documents after pre-processing. Vectors were generated using the Term frequency and the Inverse Document Frequency of the term in document. It is also possible to represent the question entered by the user as a vector. followed by the vector creation process of user's questions, the similarity score between Question Vector and Sentence Vector was measured. The same (high similarity score) was used to obtain the final result.

3.7 Extracting answers

In this phase, similarity between the Question Vector and Document Vector is computed, then weight is assigned. Based on the similarity score, and on weights, close match/answer is extracted

The algorithm uses all the sentences in the corpus to construct vectors and finding answers.

To get a clear understanding, a question answering framework was created to extract response from a corpus with rule based classification and similarity score using syntactic and semantic approach. The developed QAF has to realize what sort of inquiry being posed by the client. In order to achieve this, the corpus were separated into sentences, and then grouped them into various inquiries classifications like WHAT, WHERE, WHO, LIST AND WHEN. Similarly user question also categorized. Now based on the question categories, retrieval of the sentences belonging to that specific class takes place. This set of sentences will be considered as candidate answer. Finally based on the similarity of the user's question with the candidate answer, the sentence which has higher score will be selected as final answer.

The corpus that collected from different websites based on domain will be split into sentences accordingly. Regular expressions were used for this purpose, where it was centered on the stated condition. After dividing the corpus into sentences, next step is to classify them into their specific question category.

If the sentences contain terms like “is a,” “means,” “define,” “identified as” etc, then the program will mark them with the “WHAT” question tag.

Algorithm 2: Vectors Creation and Answer Extraction

```

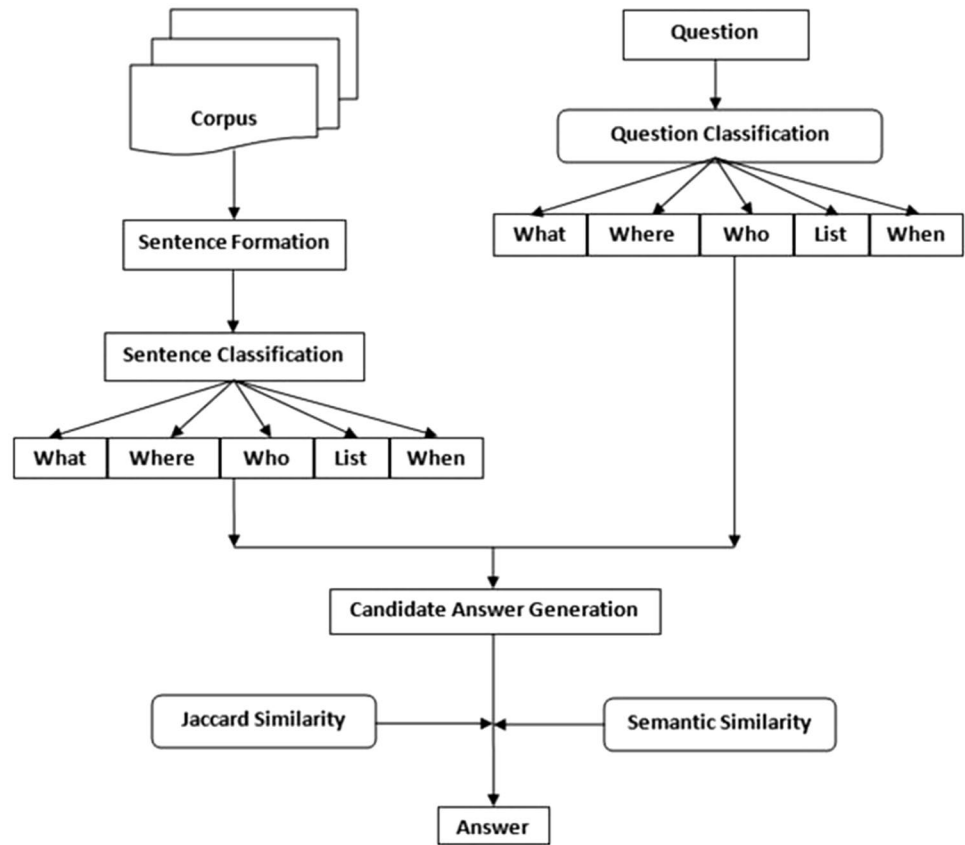
Input: Input: Tokens T query Q;
Input: Output: Answer Sentence S;
if  $T \neq$  then
    Perform Term frequency and inverse document frequency
    calculation;
    perform vector creation ;
    for all sentence vectors in V do
        | find the similarity between sentence vector and question vector;
    end
    find the highest similarity score sentence and return sentence as an
    answer;
end

```

3.8 New approach

Maximum care was taken in all aspects during the development of the current model. This was Implemented using Python language. The phases in each module contributes to the formation of the developed QAF (Fig. 1).

Fig. 1 The proposed system's framework



Algorithm 3: Sentence Formation

Input: Text document(D) ;
 output: Sentences(S) ;
for all documents in D **do**
 perform sentence segmentation in each document;
 $S \leftarrow S \cup s$;
end
 return S;

Algorithm 4: Classification of Sentence

```

Information = Knowledge Base K, Sentences S;
Output = Labeled sentences ;
Perform Preprocessing on Sentence Database to remove noise. And
  extract keywords.;
Find frequency of key-word  $W_i$  as a particular class in knowledge base.;
calculateclassscore (sentence, class_name, show_details = True);
if key - word present in class_word then
  | score  $\leftarrow$  score + 1 ;
end
Calculate the score of each sentence in all classes.;
Compare the score to discover which class the sentence have the most
  noteworthy score;
Add the class Tag to sentence which has most extreme score ;
return label sentences ;

```

3.9 Pseudo code: question classification**Algorithm 5:** Question Classification

```

input = Knowledge base K, User question q;
Output = Labeledquestion(tagQues) ;
Perform Preprocessing on user Question to extricate keywords;
Add every one of the watchwords in the Corpu_word2 show individually
  and furthermore include it in the Class_words2 list each for five classes.
;
calculate_class_score2(user_Ques, class_name2, show_details2 = True);
if watchword present in class_word2 then
  | score2 = score2 + 1;
end
Calculate the score of user_Ques in every five classes.;
Compare the similitude score of user_Ques with Question classes;
Add the q_class Tag to use_Ques to which it have greatest score ;
return tag_Ques;

```

3.10 Pseudo code: candidate answer generation**Algorithm 6:** Candidate Answer Generation

```

Input : labelsentences(t), labeledQuestion(tag_Ques) ;
Output : Candidateanswer(cand_Ans) ;
for eachsentences(s) in labelsentences(t) do
  | if q_class is available in sentence(s) then
  | | cand_Ans  $\leftarrow$  S;
  | end
end
return cand_Ans ;

```

3.11 Pseudo code: answer extraction utilizing Jaccard similarity approach

Algorithm 7: Answer Extraction

```

Input : Candidateanswer(cand_Ans), UserQuestion(user_Ques);
Output: Answer(ans1) ;
for eachsentCandincand_Ansanduser_Ques do
    perform preprocessing to extricate keyword ;
    Makeset1 = set(key_sent_Cand), set2 = set(key_user_Ques);
    Using two sets to measure the Jaccard similarity ;
    Measure the similarity of Jaccard and pick the sentence with the
    highest score of similarity and the lowest value of overlap.;
    Ans1 ← Ques, sent;
end
return Ans1;

```

3.12 Pseudo code: answer extraction utilizing Jaccard similarity and semantic similitude

Algorithm 8: Answer Extraction

```

Input : Candidateanswer(cand_Ans), UserQuestion(user_Ques);
Output : Answer(ans1)
for eachwordinuser_Ques do
    performpreprocessing(tokenization, stopword, lemmatization);
    for eachsentenceincand_Ans do
        | Preprocess keyword extraction;
    end
    for eachtokeninuseuser_Ques do
        | Find synonyms;
    end
    Calculate the User Query and Candidate Answer value;
    Createset1 = set(key_sent_Cand), set2 = set(key_user_Ques);
    Calculate the Jaccard similarity using two sets and add the score
    with it;
    Compare the final score of semantic and Jaccard similarity and
    select the sentences with highest similarity score.;
    Add the selected sentence to list Ans1;
end
return Ans1;

```

Test Question on Domain 'Hyderabad Tourism':

1. What is the number of inhabitants in Hyderabad?
2. What is the absolute zone of Hyderabad?
3. Where is chilkur balaji sanctuary?
4. In which spot chowmahalla royal residence is found?
5. What number religious spots are there in Hyderabad?
6. List out the notable royal residences in Hyderabad?
7. In which year qutb shahi tombs were constructed?
8. Who is the proprietor of falaknuma castle?

4 Results

Developed question answering framework together with the two previous solutions that we addressed the same domain earlier. Contrasted these three methods by giving them the same domain corpus, and analysis is carried out using the same and equal number of test data-set (Questions). Comparison and analysis of the final results obtained by the three methods were shown using confusion matrix (Tables 1, 2 and 3) and graphs (Figs. 2 and 3).

Table 1 Confusion matrix for sentence classification

50	A (WHAT)	B (WHEN)	C (WHO)	D (WHERE)	E (LIST)
A (WHAT)	16	0	0	0	0
B (WHEN)	0	7	0	0	0
C (WHO)	0	0	9	0	0
D (WHERE)	2	0	0	9	0
E (LIST)	0	0	0	0	7

Table 2 Final result of sentence classification

Class	Precision	Recall	F-measures
WHAT	0.88	1	0.93
WHEN	1	1	1
WHO	1	1	1
WHERE	1	0.81	0.89
LIST	1	1	1

Table 4 Different QA methods using fixed questions

QA methods	Total_ question	Correct	Wrong	Accuracy
AIML	50	50	0	100
IR-based	50	24	26	48
Class_Jaccard	50	42	8	84
Class_semantic	50	46	4	92

Five common different measures for the evaluation of the classification quality have been used: Accuracy, Error rate, Precision, Recall, and F-Measure. Accuracy is the proportion of the total number of predictions where correctly calculated.

$$Accuracy = (TP + TN) / (TP + FP + TN + FN) \tag{1}$$

$$\begin{aligned} \text{Overall Accuracy} &= (16 + 7 + 9 + 9 + 7) / 50 \\ &= 48 / 50 \\ &= 0.96 \end{aligned}$$

Error rate is the percentage of instances that were incorrectly predicted into the class they didnt belong to.

$$Errorrate = FP + FN / TP + TN + FP + FN \tag{2}$$

$$Errorrate = 1 - Accuracy \tag{3}$$

Precision is the ratio of the correctly classified cases to the total number of misclassified cases and correctly classified cases.

Table 5 Different QA methods using variable questions

QA methods	Total question	Correct	Wrong	Accuracy
AIML	50	0	50	0
IR-Based	50	24	26	48
Class_Jaccard	50	42	8	84
Class_semantic	50	46	4	92

$$Precision = TP / TP + FP \tag{4}$$

$$\begin{aligned} \text{Precision (WHAT)} &= TPA / (TPA + EBA + ECA + EDA + EEA) \\ &= 16 / (16 + 0 + 0 + 2 + 0) \\ &= 0.88 \end{aligned}$$

Recall is the ratio of correctly classified cases to the total number of unclassified cases and correctly classified cases.

$$Recall = TP / TP + FN \tag{5}$$

$$\begin{aligned} \text{Recall (WHAT)} &= TPA / (TPA + EAB + EAC + EAD + EAE) \\ &= 16 / (16 + 0 + 0 + 0 + 0) \\ &= 16 / 16 \end{aligned}$$

Table 3 Overall comparison of different methods

Methods	Domain	Corpus sents	Total ques	Ques phrase	Correct	Wrong	Accuracy (%)
IRBased	Closed	50	50	Open	24	26	48
AIML	Closed	50	50	Closed	50	0	100
AIML	Closed	50	50	Open	0	50	0
Classification (Jaccard similarity)	Closed	50	50	Open	42	8	84
Classification (semantic)	Closed	50	50	Open	46	4	92

Fig. 2 Overall comparison of different methods with fixed/closed question phrase

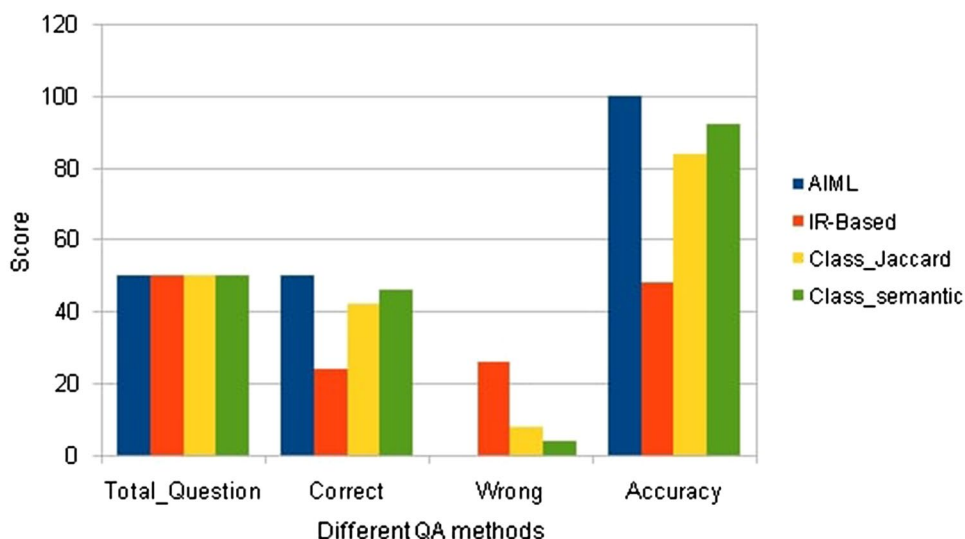
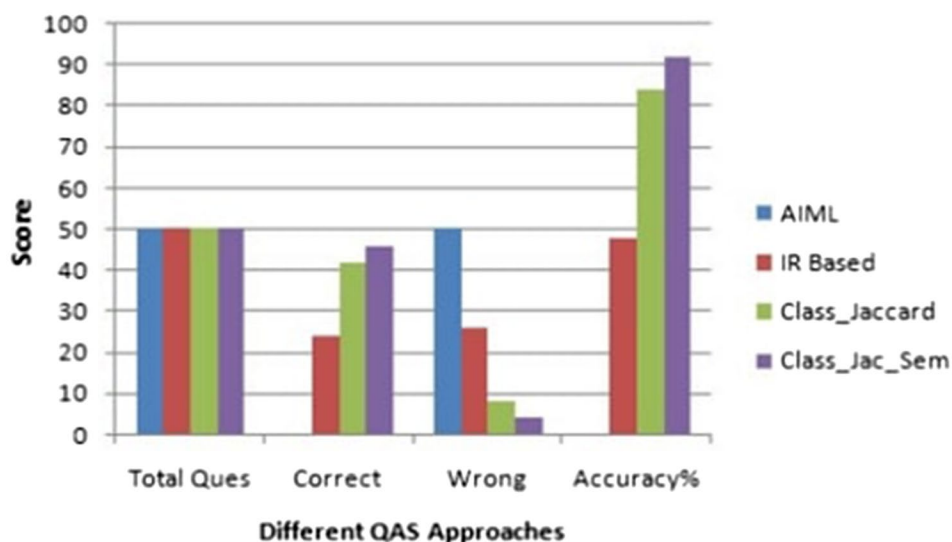


Fig. 3 Overall comparison of different methods with variable/open question phrase



= 1

The F-measure has been used to combine the recall and precision which is considered a good indicator of the relationship between them. F-measure is most desirable in cases with uneven class distribution (Tables 4, 5).

$$F - measure = 2 * ((Precision * Recall)/(Precision + Recall)) \tag{6}$$

$$F\text{-measure (WHAT)} = 2 * (0.88 * 1/0.88 + 1) = 0.93$$

Note: In the same way the recall for the all the remaining classes are calculated.

4.1 Observation

Figures 2 and 3 are the graphic representation of Table 3 showing us the general comparison of the different techniques adopted in the execution of the question answering framework. Where, Fig. 2 illustrates the contrast of strategies without modifying the phrasing of the question contained in the AIML database. In that case, AIML gives us the highest precision relative to other approaches. Whereas, in Fig. 3, if we change the way of formulating the problem, then AIML will have Zero Accuracy. When the classification method was used, 84% accuracy was achieved, which is

better than that of IR approach (which gave 48% accuracy). Finally, at long last by incorporating the Semantic similarity with the grouping approach, 92% of Accuracy was achieved.

5 Conclusion and future work

In light of the past methodologies and endeavors on Closed area question answering, it is found that before recovering the appropriate Answering, the framework should know the kind of the question posted by client. For this, the idea of information based classification was brought into picture. So that the developed framework, accurately categorizes the sentence to which it belongs. A mixture of strategies such as NLP, IR and Classification Techniques were incorporated in the QAF development. It is also found that, considering both syntactic and semantic likenesses gives the better outcomes. Similarly, the POS tag of the sentence can be used to fit them under various question categories. Utmost care must be taken while grouping the sentences and also to recover the appropriate response in an increasingly semantic manner.

References

- Bhardwaj, D., Pakray, P., Bentham, J., Saha, S., Mizoram, N. I. T., Gelbukh, A. (2016). Question answering system for frequently asked questions. In *Proceedings of the final workshop 7 December 2016, Naples* (p. 129).
- Bhoir, V., Potey, M. A. (2014). Question answering system: A heuristic approach. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)* (pp. 165–170). IEEE.
- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with python: Analyzing text with the natural language toolkit*. Sebastopol: O'Reilly Media Inc.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. Berlin: Springer.
- Cimiano, P., Unger, C., & McCrae, J. (2014). Ontology-based interpretation of natural language. *Synthesis Lectures on Human Language Technologies*, 7(2), 1–178.
- Cui, W., Xiao, Y., Wang, H., Song, Y., Hwang, S., Wang, W. (2019). Kbaqa: Learning question answering over qa corpora and knowledge bases. arXiv preprint [arXiv:1903.02419](https://arxiv.org/abs/1903.02419).
- Ferret, O., Grau, B., Hurault-Plantet, M., Illouz, G., Jacquemin, C., Monceaux, L., et al. (2002). How nlp can improve question answering. *KO Knowledge Organization*, 29(3–4), 135–155.
- Fu, J., Xu, J., Jia, K. (2009). Domain ontology based automatic question answering. In *2009 international conference on computer engineering and technology* (Vol. 2, pp. 346–349). IEEE.
- Han, K.-S., Song, Y.-I., Rim, H.-C. (2006). Probabilistic model for definitional question answering. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 212–219).
- Hao, Y., Zhang, Y., Liu, K., He, S., Liu, Z., Wu, H., Zhao, J. (2017). An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th annual meeting of the association for computational linguistics* (Vol. 1: Long Papers, pp. 221–231).
- Höffner, K., Walter, S., Marx, E., Usbeck, R., Lehmann, J., & Ngomo, A.-C. N. (2017). Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6), 895–920.
- Kaushik, S. (2011). *Artificial intelligence*. Cengage Learning India Private Limited. ISBN 9788131510995. Retrieved from <https://books.google.co.in/books?id=1KoAoQEACAAJ>.
- Lende, S. P., Raghuvanshi, M. M. (2016). Question answering system on education acts using nlp techniques. In *2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave)* (pp. 1–6). IEEE.
- Marietto, M. D. G. B., de Aguiar, R. V., Barbosa, G. D. O., Botelho, W. T., Pimentel, E., França, R. D. S., da Silva, V. L. (2013). Artificial intelligence markup language: A brief tutorial. arXiv preprint [arXiv:1307.3091](https://arxiv.org/abs/1307.3091).
- Meadow, C. T., Boyce, B. R., Kraft, D. H., & Barry, C. L. (1992). *Text information retrieval systems* (Vol. 20). San Diego, CA: Academic Press.
- Mishra, A., & Jain, S. K. (2016). A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, 28(3), 345–361.
- Mishra, A., Mishra, N., Agrawal, A. (2010). Context-aware restricted geographical domain question answering system. In *2010 international conference on computational intelligence and communication networks* (pp. 548–553). IEEE.
- Ostapov, Y. (2011). Question answering in a natural language understanding system based on object-oriented semantics. arXiv preprint [arXiv:1111.4343](https://arxiv.org/abs/1111.4343).
- Perera, R. (2012). Ipedagogy: Question answering system based on web information clustering. In *2012 IEEE fourth international conference on technology for education* (pp. 245–246). IEEE.
- Pragisha, K., & Reghuraj, P. C. (2014). A natural language question answering system in malayalam using domain dependent document collection as repository. *International Journal of Computational Linguistics and Natural Language Processing*, 3(3), 2279–0756.
- Russell, S. J., & Norvig, P. (2016). *Artificial intelligence: A modern approach*. Kuala Lumpur: Pearson Education Limited.
- Sahu, S., Vasnik, N., & Roy, D. (2012). Prashnottar: A hindi question answering system. *International Journal of Computer Science & Information Technology*, 4(2), 149.
- Sun, H., Dhingra, B., Zaheer, M., Mazaitis, K., Salakhutdinov, R., Cohen, W. W. (2018). Open domain question answering using early fusion of knowledge bases and text. arXiv preprint [arXiv:1809.00782](https://arxiv.org/abs/1809.00782).
- Unger, C., Cimiano, P. (2011). Pythia: Compositional meaning construction for ontology-based question answering on the semantic web. In *International conference on application of natural language to information systems* (pp. 153–160). Springer.
- Yang, Y., Yih, W., Meek, C. (2015). Wikiqa: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 2013–2018).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.