

QUASE: AN Ontology-Based Domain Specific Natural Language Question Answering System

Vaibhav Mishra, Nitesh Khilwani



Abstract: Since early days Question Answering (QA) has been an intuitive way of understanding the concept by humans. Considering its inevitable importance it has been introduced to children from very early age and they are promoted to ask more and more questions. With the progress in Machine Learning & Ontological semantics, Natural Language Question Answering (NLQA) has gained more popularity in recent years. In this paper QUASE (QuesTion Answering System for Education) question answering system for answering natural language questions has been proposed which help to find answer for any given question in a closed domain containing finite set of documents. The QA system mainly focuses on factoid questions. QUASE has used Question Taxonomy for Question Classification. Several Natural Language Processing techniques like Part of Speech (POS) tagging, Lemmatization, Sentence Tokenization have been applied for document processing to make search better and faster. DBPedia ontology has been used to validate the candidate answers. By application of this system the learners can gain knowledge on their own by getting precise answers to their questions asked in natural language instead of getting back merely a list of documents. The precision, recall and F measure metrics have been taken into account to evaluate the performance of answer type evaluation. The metric Mean Reciprocal Rank has been considered to evaluate the performance of QA system. Our experiment has shown significant improvement in classifying the questions in to correct answer types over other methods with approximately 91% accuracy and also providing better performance as a QA system in closed domain search.

Keywords: DBPedia Ontology, Question Answering System, Question Classification, QUASE, Natural Language Processing, Machine Learning.

I. INTRODUCTION

In recent years, human's quest of information has made use of web much popular. Traditional search engines like Google and Bing provides the user with a set of links based on query asked and then user goes through these links and try to get the answer they are looking for.

Manuscript published on November 30, 2019.

* Correspondence Author

Vaibhav Mishra*, Research Scholar, Department of Computer Science & Engineering, Mewar University, Chittorgarh, Rajasthan, India, Email: vaib.mishra@gmail.com

Dr. Nitesh Khilwani, Technical Architect, Edifecs RoundGlass, Noida, India. Email: nitesh.nifft@gmail.com

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](https://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Answering System for Education, here after referred as Hence there is need of services which help users to filter irrelevant information quickly & provide the relevant answer. Question answering is such a service which provides correct answer to specific questions as opposed to providing list of documents. This paper describes a question answering system called -QuesTion QUASE. There are certain other services also available like google providing QnA based answers but they are limited to public documents. This research is proposing system for identifying answers from documents trained for closed domain.

QUASE is closed domain, factoid question answering system that takes questions queried in English and attempts to provide answers, if exist, in provided set of documents. The idea of QUASE is to use prediction abilities of Supervised Machine Learning algorithms to find expected answer type and use of this answer type in validating answers with help of ontology. NLP, ML & Ontology technology is used in developing this system. There are three major modules namely Question Processing, Document Processing & Answer Processing.

In question processing module input question is analyzed to identify the type of answer expected and question is framed into a query for search. It has two components Question Classification, also known as answer type prediction & Query Development. In question classification given question is categorized into distinct categories which indicates what type of answer this question wants. In query development phase, key information from the question is extracted which help in identifying the answer.

Different kinds of QA frameworks have distinctive models, the majority of them pursue a system in which question grouping plays huge role [1]. In Question Classification (QC) a question is classified into expected answer type. If expected answer type is known, then it can not only reduce search space needed to find the answer but also help in finding the true answer in a given set of candidate answers [2]. For example, knowing the class of the question —Who was the first female Prime Minister of India? is of type —human, one should only consider the name entities in candidate answer list, which is of type —human and leave the rest.

There are basically two different approaches for question classification: Rule based and Learning based.

Rule based approaches attempt to match the asked question with few manually handcrafted rules [3][4]. These approaches however, suffer from the need to define too many rules [5].

Rule-based approach may work well on a particular dataset, but may not on a new dataset and consequently it is difficult to scale them. The difficulties of rule-based approaches have been provided with the help of an example in [5].

All the following samples are same question which has been restructured in different syntactical forms:

- What are the tourist attractions in Shimla?
- Name the famous tourist attractions in Shimla?
- What do most tourists see in Shimla?
- What places attracts tourists to Shimla?
- What is worth visiting in Shimla?

All of the above questions refer to same class while they have different syntactical forms and therefore they need different matching rules. So it is difficult to make a manual classifier with a limited amount of rules.

Learning-based approaches on the other hand, perform the classification by extracting some features from questions, training a classifier and predicting the class label using the trained classifier. Machine learning help in learning based approach as it work on semantic features of text hence syntactic structure of text does not influence the outcome and so work well with different kind of data sets. Fig 1 displays a block diagram of a general QA system.

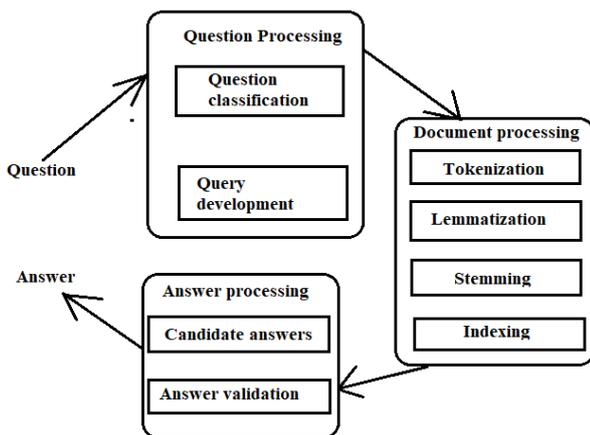


Fig. 1: A block diagram of QA system

It has been seen that the performance of Question Classification has significant influence on the overall performance of a QA system [6][7][8].

In query development phase a query is developed based on some logic. It is required because a question may have information which is not effective in searching answer relevant material so query should possess only information helpful in finding the relevant content. It may be represented as plain text format or in some query language. This query is used for search over documents to find candidate answers. Question Classification module in the proposed research has been developed based on Li & Roth question taxonomy [5].

Documents of search domain also need some kind of processing so that possibility of successful search can be increased. In most of the Question Answering Systems various NLP techniques like tokenization, lemmatization, stemming indexing etc. is used to organize data so that search can be made faster and productive [9].

In the current scenario, text of search domain has been normalized for faster and better search results. While in

Answer Processing module candidate answer received from Document Processing are validated based on Question Classification results. The whole QUASE system has been evaluated over CMU Wikipedia dataset [10].

Answer processing module is the final stage of Question Answering System where an answer is figured out from either a passage or set of possible answers received from document processing part. The candidate answer/s found in such a manner is subject to validation. For validation purpose these answer/s are matched with the question demand. Those who closely match are presented to user.

II. RELATED WORK

There has been a lot of research done in Question Answering field. Question answering research span is close to six decades. The initial work on QnA started in early 1960s which were database-centric [11][12]. Further improvements involved logical representations of Questions to better understand question before querying the database [13]. Some QAS used wh word classification and language processing techniques like lexical and semantic clues [14]. Later on, the focus of QAS development moved toward open domain QASs. TREC Evaluation campaign started open domain question answering research over unstructured data source which takes place every year since 1999 [15]. With rise of World Wide Web enormous text data become available on web which has been utilized as knowledge base [16]. Several web-based QAS has been developed [5]. Some of them are open domain [17][18][19]. The others are of restricted domain [20][21][22]. Most of the questions in restricted domain QASs are factoid questions.

Some of the QAS having similarity with QUASE are mentioned here.

QANDA is a question answering system close in nature to QUASE in intention and functionality. QANDA takes questions expressed in English and provide a short and concise answer (a noun phrase or sentence) [23]. QANDA incorporates three important technologies: first, Knowledge Representation (KR), second Natural Language Processing (NLP) and third Information Retrieval (IR). A question is represented as first order logic expression while in QUASE question is represented as query usually made up of noun or verb phrase. Moreover, QANDA does not use ontological relations like QUASE to validate answers.

Ontoseek is an information retrieval system grouped with ontology [24]. It seeks content instead of string during retrieval process. Search focus area is specific to yellow pages and product catalogues. Queries are represented as lexical conceptual graphs and as per the authors "the problem is reduced to ontology driven graph matching where individual node and arcs match if the ontology shows that a subsumption relation exist between them". These graphs weren't constructed automatically. The Ontoseek team created a semi-automatic approach in which the user verifies links among different nodes of the constructed graph by means of provided user-interface. QUASE is not representing the query as graphs. It is classifying the query through machine learning which is nowhere used in Ontoseek. Moreover QUASE is retrieving answer automatically.

MULDER is a web based QA system related to QUASE in terms of Question Classification [18]. Mulder's classifier recognizes three types of questions namely: nominal, numerical and temporal. Mulder extracts summaries from web document and generates list of possible candidate answers. However, it does not have an inference mechanism embedded like ontology which QUASE has. Moreover QUASE classifies questions based on rich set of Li & Roth taxonomy.

AQUA is also ontology driven Question answering system which convert natural language question in to a logical query through QLL (Question Query language) with help of some pre-defined rules which is different form QUASE [25]. In question classification AQUA is focusing on wh-words with help of sentence segmentation & WordNet [26]. On the other hand QUASE uses Li & Roth question classification taxonomy.

AquaLog is a portable QAS which accept queries given in natural language along with ontology as input & returns answers retrieved from one or more KBs (Knowledge Bases), which fill-up the ontology given as input with domain-specific information [27]. It converts NL question into first query-triple & then with help of Relation Similarity Service (RSS) in to Onto-triple. RSS also contains answer engine which resolves Onto-triple to infer the answer of the onto-triple. This answer is converted to English through templates before presenting to user. This QAS is different from QUASE in question classification techniques. AquaLog uses Operational Conceptual Modeling Language (OCML) to provide portability of ontologies to seek answer of the questions while QUASE does not use ontology to seek answer. In QUASE approach answers are searched form documents provided in advance to the system and DBPedia ontology is used for answer validation purpose.

Template mapping techniques also have been tried for question type identification, and on that basis a query is sent to a specific search engine which was built for this purpose [28]. Their system find detailed content blocks from pertinent web pages and consider them as answers but brief answers are not provided by this system. This does not happen with QUASE.

PowerAqua QAS is multi-ontology system which converts natural language query in to RF triples and map these triples on whole semantic web to find answer [29]. While QUASE is relying on Question Classification based on Li & Roth taxonomy & used multi domain ontology (DBPedia) to validate the candidate answers. It is searching answer in restricted predefined natural language content not on the whole web.

QAKiS points questions containing a Named Entity identified with the appropriate response considering each question will get a pattern [30]. It performs several techniques to map user question into SPARQL query which can be fired over DBPedia SPARQL endpoint to retrieve answer. QUASE approach is quite different from it. It use machine learning for question classification &

search provided documents through search engine. QUASE uses DBPedia endpoint for answer validation purpose.

A QA system has been developed which is quiet similar to QUASE in nature [31]. They have used Li & Roth taxonomy for question classification task and for focus

identification they also try to find noun phrase with focus head & modifier list. They search query on Google & pass retrieved set of documents to passage retrieval stage after some preprocessing. Here document is split in to set of passages that are ranked based on question n gram score matching similarity through tf-idf. For answer selection first question focus basis filtration & ordering (basis on similarity matching with question) is done. Then high order filtered sentences are matched with question type. Those who are matching are presented to the user. At many stages QUASE is different from this QAS as QUASE is doing classification for 5 coarse classes (among 6) namely HUMAN, LOCATION, NUMERIC, ENTITY & ABBREVIATION. But QAS uses only 4 classes for question classification hence scope of question classification is wider in QUASE [31]. It is working on closed domain so it has its searching universe with it which is provided by domain experts. QUASE is using in-built search engine which is not Google but based on Apache Lucene [32]. QUASE receives sentence which are similar in content with query after searching is over. These are possible candidate answers so no passage retrieval & answer filtration is involved. Now QUASE does answer validation based on question classification with help of DBPedia ontology [33]. To query linked data to identify the types supported by candidate answers here, SPARQL is used [34]. Those which are matching with question type are the answers presented to user. QUASE can work without internet while their system doesn't.

III. METHODOLOGY

A. Some Background Details

In this section basic building blocks required to develop QUASE has been described which will help in understanding the QUASE architecture.

- *Ontology*

In computer science, ontology is considered a technical term depicting an artifact which enables the modeling of knowledge base for some domain, whether real or imaginary [35]. An ontology includes a representation, proper naming, and define categorizations, characteristics, and relations among concepts (or ideas), data, and entities that exhibit one, many, or all domains. Every field makes ontologies to restrict intricacy and sort out data into information and learning. New ontologies development is ongoing which had shown progress in problem solving of respective domain as ontology help in mining the knowledge of that domain.

- *Machine learning algorithms used*

Proposed research is using Supervised Machine Learning Algorithms as mentioned below.

a) SGD Classifier (Stochastic Gradient Descent Classifier): These are linear classifiers (SVM, linear regression, logistic regression etc.) with SGD training. This estimator actualizes regularized straight models with SGD learning. SGD allows scaling these linear classifiers to much bigger training sets.

The model it fits can be controlled with the loss parameter [36].

b) Linear SVC (Linear Support Vector Classification): Linear SVC is a classification method of Support Vector Machine (SVM). It is a multiclass classification method which is handled through one-vs-rest (OVR) strategy [37]. This algorithm uses linear kernel which is a similarity measure function in SVM which help in finding decision boundary on a high dimensional feature space which was not possible to find in actual input feature space.

c) One-vs-Rest Strategy: Here Multiclass classification problem is reduced to same number of binary class classifications as number of classes in which classification has to be done. During training of this model Linear SVC deduce coefficients & intercept of kernel function for each binary classifier. Now when test data sample is given, this algorithm calculates score of that sample with respect to each binary classifier (based on sample features & coefficient, intercept of that classifier). The binary classifier which has maximum score is predicted as output of sample data.

B. QUASE Architecture

Fig 2 demonstrates the architecture of the proposed QUASE system.

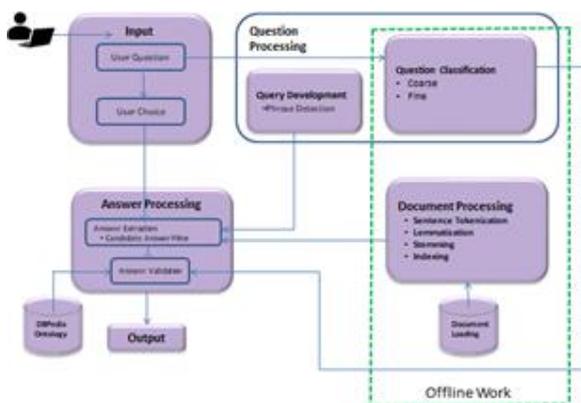


Fig. 2: QUASE Architecture

This section explains the proposed QUASE architecture (Figure 2). The model integrates various components such as machine learning, NLP techniques, indexing based retrieval mechanism, and ontology processing.

C. The Overall Algorithm

The overall steps of the algorithm have been mentioned below.

Step 1: Input

User will input her question in form of natural language text. At this stage no syntactic or semantic validation is performed so user is free to input any kind of text in English.

Step 1.1: User question: User poses a question in natural language text through this interface. If the user does not get satisfactory answer then question can be reformulated here.

Step 1.2: User choice: In QUASE, user has a facility of choosing either correct answer or nearly correct answer. Since QUASE is a closed domain QA system hence it

will find answer in a limited set of documents. So there may be a case where accurate answer may not be found by the system if it does not exist in documents searched. In that case a nearby answer will be provided to user which may satisfy user need.

Step2: Question analysis

This phase consist of two phases question classification & query development.

Step 2.1: Question Classification: In QUASE input question has been classified using Question Taxonomy as in [5]. The arrangement of questions grouping or classifying them is known as question taxonomy. A kind of taxonomy has been defined which consist 6 coarse-grained and 50 fine-grained classes [5]. Table 1 shows this taxonomy.

Table I: The coarse, fine grained categories of questions

COARSE	FINE
ABBR	abbreviation, expansion
HUM	description, group, individual, title
LOC	city, country, mountain, other, state
ENTY	animal, body, color, creation, instrument, disease, religion, food, currency, substance, letter, other, plant, product, event, sport, language, symbol, vehicle, term, technique, word
NUM	code, count, distance, date, money, period, other, temperature, order, speed, percent, size, weight
DESC	manner, definition, reason, description

So fundamentally question classification is an issue of Multi-Class Classification. To solve this problem two machine learning algorithms (of Scikit Python Library) have been used namely SGDClassifier & LinearSVC with different penalties.

Step 2.2: Query Development: In QUASE, query is developed by identifying phrase or clause detection. Phrases and clauses are meaningful chunk of text which contain the information pertinent to the anticipated answer and filter out irrelevant information. The question phrase is extracted through shallow parsing or chunking technique of NLP technology. The Noun Phrase chunk identifies the phrase/clauses which have to be found in document set for obtaining answer. If NP chunk is not available then VP chunk is extracted & used to obtain the answer set from documents.

Step 3: Document loading

Set of document from which answer has to be searched is loaded in QUASE system manually. Since QUASE is closed domain in nature hence answer searching will be restricted to these documents only. Based on user feedback domain experts can upload more document as and when needed.

Step 3.1: Document processing: In this section the loaded documents are processed in order to make them ready for searching. To process document following steps are followed.

- Each document is broken in to sentences with help of Java libraries. Each sentence is kept in separate file (hereafter called document) so that indexing can be done easily with help of Apache Lucene text search library. One copy of all these documents is also kept to keep a backup which is used in later stage.
- Query developed in earlier section & each document is lemmatized so that all of its words can reduce to its base form & easily findable to search engine. For this purpose StanfordCoreNLP pipeline is used.
- QUASE has used Porter Stemmer for stemming. Stemming has problem of under & over stemming. To overcome this problem QUASE first lemmatize tokens (of both document & query) then do stemming.
- QUASE has used Apache Lucene search library to create indexing of all document content.

Step 4: Answer processing

In this phase QUASE search the document to find the candidate answers and validate them based on the answer type extracted from Question Analysis module. It is a two-step process which is described below.

Step 4.1: Answer Extraction: The query developed, lemmatized & stemmed in earlier section is given to the Lucene search library as input. Lucene search this query over indexed documents to find matches. Matches will be lemmatized & stemmed sentence which will not work well with validation process as it may have lost its semantic importance due to reduction in base form. So for validation purpose original sentence equivalent of matched sentence is retrieved from backup and passed to answer validation phase. Search is based on user’s choice given at input stage. If the user wants exact match then searching will be done over all terms of the entered query. If user want nearest match then combinations of terms involve in query will be formed & search will be performed for them in narrow to wider way.

Step 4.2: Answer validation: In this phase QUASE will have one or more candidate answers. To find valid answers among them QUASE match these sentences with answer type extracted in Question Classification phase. This happens with help of ontology which keeps the semantic information of words available in the sentence. QUASE has used DBpedia Ontology. With help of SPARQL (an ontological query language) DBpedia ontology is queried and candidate answers are tagged in to coarse classes. These tagged classes are matched with answer type classes. Those sentences who are matching with answer type coarse classes are considered valid answers. Top five valid answers are taken in to consideration.

Step 4.3: Answer output display: QUASE output are sentences from the documents (provided by field experts) which are matching with the given query (based on user choice) and match with the answer type of given question. Top five validated sentences are shown to the user received from answer validation phase.

IV. EXPERIMENTS

A. Ontology used

DBpedia is a crowd-sourced community work which extracts structured or managed content from Wikipedia

content [38]. This DBpedia information base characterizes 4.58 million things, among which 4.22 million are arranged in a reliable ontology called DBpedia ontology. It is a shallow, cross- domain ontology. This ontology is queried via SPARQL ("SPARQL Protocol and RDF Query Language") endpoint [39]. A SPARQL endpoint allows users to request a knowledge base through SPARQL language. Results are retrieved in a format suitable for machine-processing. DBpedia is freely available to use under "Creative Commons Attribution- ShareAlike' 3.0 License and the "GNU Free Documentation License" [40].

B. Evaluation metrics

For the purpose of evaluating the answer type outcome, three evaluation metrics have been used. They are precision, recall and F1 measure. In a QA system, it is possible that the system may provide an answer which is not correct. The performance of a QA system depends on the proportion of the wrong answers provided by the system. In this regard, there are four important terms to be mentioned: TP (True positive), FP (False positive), TN (True negative), and FN (False negative). When a correct answer is identified by the user as correct, it is TP case. If a correct answer is identified as a wrong answer then it is FP case. When a correct answer is identified as wrong then it is TN case. Last of all, if an incorrect answer is detected as incorrect, then it is FN case.

Precision refers to portions of the answers detected by the system as correct answer. Recall describes the portion of right answers detected by the system. F1 score actually measures the correctness of a test. The precision, recall and F1 measure are calculated using the following equations.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP}) \tag{1}$$

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \tag{2}$$

$$\text{F1 Score} = 2 \cdot (\text{Precision} * \text{Recall}) / (\text{Precision} + \text{Recall}) \tag{3}$$

For the different kind of questions, the evaluation procedure of answer processing relies on TREC (Text Retrieval Conference), using Mean Reciprocal Rank (MRR) standards as shown in the following formula:

$$MRR = \frac{1}{n} \sum_{i=1}^n 1/r_i \tag{4}$$

Where, n is the number of the questions to be tested and r_i is the position of the first correct answer to the question number i among top 5 answers, if there is no correct answer available in candidate sentences, the value is considered to be 0.

C. Dataset

For QUASE system evaluation CMU Wikipedia Question/Answer dataset [10] has been used. However for Abbreviation type no specific dataset has been found hence a set of 25 questions has been created from Newspaper Wikipedia articles.



QUASE: AN Ontology-Based Domain Specific Natural Language Question Answering System

D. Experiment 1: Evaluation of question classification module

In QUASE, Question Classification (QC) module is tested for 5 Coarse & 46 fine grained classes. DESC classification has been left as QUASE focus is for factoid questions. QC module is trained over 4290 question sets & tested over 362 question set provided by [5][41].

Due to high dimensional question text data in space, overfitting situation is encountered which cause lesser generalization. To achieve generalization, L1, L2 & Elastic Net penalties are used. L1 penalty is also known as Lasso Regression [42]. It helps in better prediction by shrinking the less important feature's coefficient to zero thus forces non-essential features to get removed from feature vector. The alternative name of L2 penalty is Ridge Regression [43]. It spreads the penalty term among all the features. Both L1 & L2 have their own pros & cons. Thus an algorithm has been developed known as the Elastic Net, which penalizes both the L1 and L2 norms with individual tuning parameters, as a way to achieve the finest of both LASSO & Ridge [44].

QC prediction model has been trained on SGDCClassifier with L1, L2 & Elastic Net penalty while Linear SVC is used with L1, L2 penalty & L1 based feature selection.

Outcomes of both SGDCClassifier & LinearSVC have been checked with mentioned penalties and taken only those outcomes which were most common among them. Results have been presented in Table 2 & 3.

Table II: Answer Type Outcome Evaluation—Coarse Grade

Coarse Category	True Positive	False Positive	False Negative	Precision	Recall	F1 Measure
HUM	62	5	3	0.93	0.95	0.94
NLM	102	2	11	0.98	0.90	0.94
ENTY	84	19	10	0.82	0.89	0.85
LOC	72	8	9	0.90	0.89	0.89
ABBR	8	0	1	1.00	0.89	0.94
Average						0.91

Table III: Answer Type Outcome Evaluation—Fine Grade

Coarse Category	Fine Category	True Positive	False Positive	False Negative	Precision	Recall	F1 Measure	
HUM	description	1.00	0.00	2.00	1.00	0.33	0.50	
	group	5.00	1.00	1.00	0.83	0.83	0.83	
	individual	53.00	4.00	2.00	0.93	0.96	0.95	
	title	0.00	0.00	1.00	NA	0.00	NA	
HUM Average								0.76
LOC	city	16.00	0.00	3.00	1.00	0.83	0.91	
	country	3.00	0.00	0.00	1.00	1.00	1.00	
	mountain	3.00	1.00	1.00	0.75	0.75	0.75	
	other	42.00	2.00	8.00	0.95	0.84	0.89	
	state	4.00	5.00	3.00	0.44	0.57	0.50	
	LOC Average							
ENTY	animal	11.00	3.00	5.00	0.79	0.69	0.73	
	body	1.00	2.00	1.00	0.33	0.50	0.40	
	color	10.00	0.00	0.00	1.00	1.00	1.00	
	currency	5.00	1.00	1.00	0.83	0.83	0.83	
	disease	1.00	3.00	1.00	0.25	0.50	0.33	
	event	1.00	3.00	1.00	0.25	0.50	0.33	
	food	3.00	0.00	1.00	1.00	0.75	0.86	
	instrument	1.00	0.00	0.00	1.00	1.00	1.00	
	language	2.00	0.00	0.00	1.00	1.00	1.00	
	other	6.00	29.00	6.00	0.23	0.50	0.32	
	plant	2.00	0.00	3.00	1.00	0.40	0.57	
	product	1.00	0.00	3.00	1.00	0.25	0.40	
	sport	1.00	1.00	0.00	0.50	1.00	0.67	
	substance	6.00	4.00	9.00	0.60	0.40	0.48	
	technique	1.00	0.00	0.00	1.00	1.00	1.00	
	term	7.00	3.00	0.00	0.70	1.00	0.82	
	vehicle	2.00	0.00	2.00	1.00	0.50	0.67	
	symbol	0.00	0.00	0.00	NA	NA	NA	
	religion	0.00	0.00	0.00	NA	NA	NA	
	letter	0.00	0.00	0.00	NA	NA	NA	
	creation	0.00	2.00	0.00	0.00	NA	NA	
	word	0.00	2.00	0.00	0.00	NA	NA	
	ENTY Average							
NUM	count	9.00	3.00	0.00	0.75	1.00	0.86	
	date	46.00	1.00	1.00	0.98	0.98	0.98	
	distance	8.00	0.00	8.00	1.00	0.50	0.67	
	money	1.00	0.00	2.00	1.00	0.33	0.50	
	other	5.00	1.00	7.00	0.83	0.42	0.56	
	percent	3.00	2.00	0.00	0.60	1.00	0.75	
	period	8.00	4.00	0.00	0.67	1.00	0.80	
	speed	6.00	0.00	0.00	1.00	1.00	1.00	
	temperature	4.00	0.00	1.00	1.00	0.80	0.89	
	weight	3.00	0.00	1.00	1.00	0.75	0.86	
	code	0.00	0.00	0.00	NA	NA	NA	
	order	0.00	0.00	0.00	NA	NA	NA	
	size	0.00	0.00	0.00	NA	NA	NA	
	NUM Average							
ABBR	abbreviation	1.00	0.00	0.00	1.00	1.00	1.00	
	expansion	7.00	0.00	1.00	1.00	0.88	0.93	
ABBR Average								0.97

E. Experiment 2: Evaluation of QA system performance

Figure 3 below shows the scored value of MRR for several types of questions tested on QUASE for answer processing (extraction & validation). Corresponding MRR values have been presented in the table.

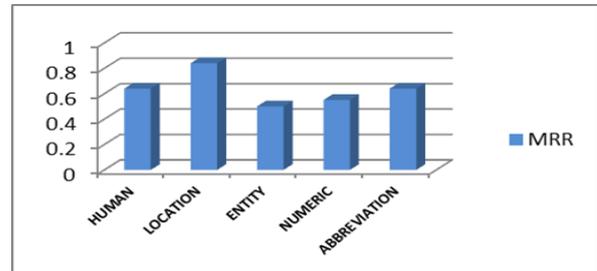


Fig. 3: Evaluation of QA performance

Table IV: MRR Result of QUASE Evaluation

Questions Type	Number of Questions	MRR
HUMAN	25	0.64
LOCATION	25	0.84
ENTITY	25	0.5
NUMERIC	25	0.55
ABBREVIATION	25	0.64

F. Experiment 2: Comparison of answer type evaluation performance with related systems

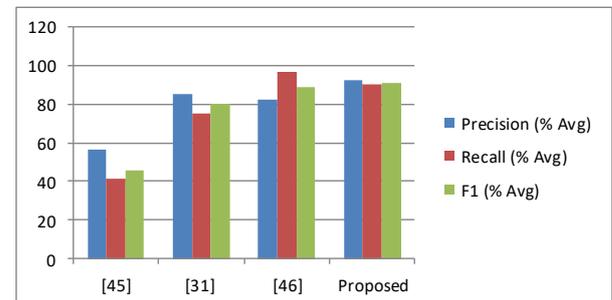


Fig. 4: Comparison of answer type evaluation

Performance of the answer type evaluation of the proposed QUASE has been compared with other related and popular systems. The research works selected for comparison are [45], [31] and [46]. The values of evaluation metrics of the proposed system has been compared with other systems as given below. The metric values with respect to the above chart have been presented in the table below.

Table V: Metric values comparison with other QAS

Reference	Precision (% Avg)	Recall (% Avg)	F1 (% Avg)
[45]	56.9	41.6	45.6
[31]	85	75.2	80
[46]	82.6	96.7	89.1
Proposed	92.6	90.4	91

V. DISCUSSION

QUASE has been tested over five coarse categories which include human, location, numeric, abbreviation and entity. A total of 125 questions with 25 questions for each question type have been used to test QUASE.

From Table 2, it can be observed that precision of ABBR questions are highest having 100% success rate though the ABBR question dataset is author developed containing 25 questions only.

Among other question types tested on standard dataset, NUM type questions have highest precision of 98% and ENTY question has lowest precision of 82%.

HUM type questions have the highest recall. The overall system classification accuracy is 91%. So the proposed system performs pretty well on ABBR, HUM, NUM and LOC type questions and provides highly acceptable performance for ENTY question type.

Question Classification module has been developed based on Li & Roth question taxonomy [5]. It has given very good results (approx. 91% accuracy).

For evaluating answer processing phase, the metric Mean Reciprocal Rank (MRR) has been used. With LOCATION question type the method performed very well and that is because the variety of locations were limited to city, country, mountain, state & other which were distinct in nature & possess less ambiguity. However, for the ENTITY question type, there are 22 fine grained classes & there QC outcome is not so good. ENTITY average is least among five coarse classes. Hence question validation itself become in efficient due to wrong answer type identification.

Answer processing module candidate answer received from document processing are validated based on question classification results. The whole QUASE system has been evaluated over CMU Wikipedia dataset [10] which shows LOCATION based questions giving most (approx. 84%) accuracy while ENTITY based questions gives least (approx. 50%) accuracy.

In comparison to other related systems, it is observed that the proposed system has the highest precision of 92.6% and second highest recall of 90.4% with highest classification score of 91%. Therefore it can be concluded that the current QA system outperforms most of the other related QA systems.

VI. CONCLUSION AND FUTURE WORK

This paper has presented and evaluated QUASE question answering system to find different types of answers including human, location, numeric, abbreviation and entity with respect to factoid questions by applying Machine Learning & POS tagging over questions and several NLP techniques like sentence tokenization, lemmatization, indexing over document preparation and finally extracting candidate answers based on Lucene search and validating them through DBpedia Ontology with help of SPARQL query.

The system is asked natural language questions to get back precise answers which can help the students/learners for the purpose of gaining knowledge by their individual effort. A learner can obtain basic knowledge on a new

topic or he/she can obtain answers to questions in advanced level.

The experimental result shows that the proposed system performs exceptionally well in detecting answer type and outperform other related QA systems with respect to question classification accuracy.

Question Classification module has given very well results (approx. 91% accuracy). The evaluation result of the whole QUASE system shows LOCATION based questions giving most (approx. 84%) accuracy while ENTITY based questions gives least (approx. 50%) accuracy.

However there is scope of improving accuracy of the system. Some areas on which work can be done in future can be improving the lack of co-reference resolution, performing query expansion to widen search area for similar semantics and enhancing lack of answering Boolean (Yes/No) type of questions.

REFERENCES

1. Ellen M. Voorhees, "Overview of the TREC 2001 question answering track", In Proceedings of the Tenth Text REtrieval Conference (TREC, pages 42–51, 2001
2. Loni, B. (2011), "A Survey of State-of-the-Art Methods on Question Classification", Literature Survey, Published on TU Delft Repository
3. David A. Hull. Xerox, "TREC-8 question answering track report", In Voorhees and Harman, 1999
4. John Prager, Dragomir Radev, Eric Brown, and Anni Coden, "The use of predictive annotation for question answering in trec8", In NIST Special Publication 500-246: The Eighth Text REtrieval Conference (TREC 8, pages 399–411. NIST, 1999
5. Xin Li and Dan Roth, "Learning question classifiers", In Proceedings of the 19th international conference on Computational linguistics, COLING '02, pages 1–7. Association for Computational Linguistics, 2002
6. A. Ittycheriah, M. Franz, W. J. Zhu, A. Ratnaparkhi, and R. J. Mammone, "IBM's statistical question answering system", In Proceedings of the 9th Text Retrieval Conference, NIST, 2001.
7. Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin yew Lin, and Deepak Ravichandran, "Toward semantics-based answer pinpointing", 2001,
8. Dan Moldovan, Marius Pasca, Sanda Harabagiu, and Mihai Surdeanu, "Performance issues and error analysis in an open-domain question answering system", ACM Trans. Inf. Syst., 21:133–154, April 2003
9. Daniel Jurafsky and James H Martin, "Speech and Language Processing", (2nd Edition) (Prentice Hall Series in Artificial Intelligence) Prentice Hall, 2008
10. Research supported by NSF IIS-0713265 (to Smith), an NSF Graduate Research Fellowship (to Heilman), NSF IIS-0712810 and IIS-0745914 (to Hwa), and Institute of Education Sciences, U.S. Department of Education R305B040063 —Question-Answer Dataset [Online] Available: <http://www.cs.cmu.edu/~ark/QA-data> [Accessed: May 26, 2018]
11. Green, B.f., Chomsky, C., Laughery, K., 1961. BASEBALL, "An automatic question answerer", Proceedings of the Western Joint Computer Conference, New York: Institute of Radio Engineers, pp. 219–224
12. Woods, W., 1973, "Progress in natural language understanding – an application to lunar geology" Proc. of the American Federation of Information Processing Societies (AFIPS), 42: 441–450, AFIPS Press, <WIKIPEDIA.ORG/wiki/Question_answering>
13. Androustopoulos, I., Ritchie, G.D., Thanisch, P., 1993, "MASQUE/SQL – an efficient and portable natural language query interface for relational databases", In: Proc. of the 6th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. Gordon and Breach Publisher Inc., Edinburgh, pp. 327–330
14. Riloff, E., Thelen, M., 2000, "A rule-based question answering system for reading comprehension tests" ANLP/NAACL Workshop on Reading Comprehension Tests as Evaluation for Computer- Based Language Understanding Systems, vol. 6, pp. 13–19

15. Ellen M. Voorhees, "Overview of the TREC 2001 question answering track", In Proceedings of the Tenth Text REtrieval Conference (TREC), pp. 42–51, 2001
16. Soricut, R., Brill, E., 2006, "Automatic question answering using the web: beyond the factoid", Journal of Information Retrieval-Special Issue on Web Information Retrieval, vol. 9(2), pp. 191-206.
17. Hovy, E., Gerber, L., Hermjacob, U., Junk, M., Lin, C., 2000, "Question answering in webclopedia", Ninth Text REtrieval Conference, Vol. 500–249 of NIST Special Publication, Gaithersburg, MD, National Institute of Standards and Technology, pp. 655–664
18. Kwok C., and Etzioni O., and Weld D.S., "Scaling Question Answering to the Web", World Wide Web, pp. 150-161, 2001
19. Zheng, Z., 2002, "The AnswerBus Question Answering System", Proceedings of Human Language Technology Conference HLT, San Diego, CA. March 24–27
20. Katz, Felshin, B., Yuret, S., Ibrahim, D., Temelkuran, B., 2002, "Omnibase: uniform access to heterogeneous data for question answering" Proc of the 7th International Workshop on Applications of Natural Language to Information Systems (NLDB)
21. Wong, W., 2004, "Practical approach to knowledge- based question answering with natural language understanding and advanced reasoning (MSc thesis)", Kolej Universiti Teknikal Kebangsaan Malaysia.
22. Benamara Farah, 2004, "Cooperative question answering in restricted domains: the WEBCOOP experiments", Workshop on Question Answering in Restricted Domains, 42nd Annual Meeting of the Association for Computational Linguistics, Barcelona, Spain, pp. 31–38
23. Breck E., and House D. and Light M., and Mani I, "Question Answering from Large Document Collections", AAAI Fall Symposium on Question Answering Systems, 1999
24. Guarino N., "OntoSeek: Content-Based Access to the Web", IEEE Intelligent Systems, pp. 70-80, 1999
25. M. Vargas-Vera, E. Motta, and J. Domingue, "AQUA: An ontology-driven question answering system", M. T. Maybury, editor, New Directions in Question Answering, pp. 53–57, Menlo Park, CA, USA, 2003. AAAI Press
26. "WordNet A lexical database for English", Princeton University, [Online] Available: <http://wordnet.princeton.edu/> [Accessed: April 26, 2015]
27. Lopez, V., Motta, E., Uren, V., "Aqualog: an ontology-driven question answering system to interface the semantic web", NAACL on Human Language Technology, New York (2006) 269 – 272
28. Parthasarathy, S., and Chen, J., 2007, "A Web-based Question Answering System for Effective e-Learning", Proceedings of the Seventh IEEE International Conference on Advanced Learning Technologies (ICALT 2007), pp. 142 – 146
29. Vanessa Lopez, Victoria Uren, Marta Reka Sabou, and Enrico Motta, "Cross ontology query answering on the semantic web: an initial evaluation", In K-CAP '09, pp. 17–24, 2009
30. Cabrio E., Cojan J., Aprosio A. P., Magnini B., Lavelli A. and Gandon F., "QAKiS: an open domain QA system based on relational patterns",. In Proceedings of the ISWC 2012 Posters & Demonstrations Track. CEUR Workshop Proceedings, vol. 914 (2012)
31. A. Waheeb, A. Babu, "An automatic web-based question answering system for e-learning", Information Technologies And Learning Tools, vol. 58, no. 2, 2017
32. Langer, S., & Beel, J. (2017), "Apache Lucene as Content-Based-Filtering Recommender System: 3 Lessons Learned", 5th International Workshop on Bibliometric-enhanced Information Retrieval (BIR) at the 39th European Conference on Information Retrieval (ECIR)
33. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z., " DBpedia: A nucleus for a web of open data", The Semantic Web. Volume 4825 of LNCS. Springer Berlin (2007), pp. 722–735
34. Harris, Steve (2013, March 21), "SPARQL 1.1 Query Language", Retrieved from <https://www.w3.org/TR/sparql11-query/>[Online; Accessed: 19 Sep 2018]
35. T. R. Gruber, "A Translation Approach to Portable Ontology Specification," Knowledge Acquisition, vol. 5, no. 2, pp. 199-220, April 1993
36. sklearn.linear_model.SGDClassifier, Retrieved from http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html [Online; Accessed: 25 Sep 2018]
37. sklearn.svm.LinearSVC, Retrieved from <http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html> [Online; Accessed: 26 Sep 2018]
38. Christian Bizer. DBpedia version 2014 released. <http://blog.dbpedia.org/2014/09/09/dbpedia-version-2014-released/>, 2014. [Online; Accessed: 20 Sep 2018]
39. Bu, Q., Simperl, E., Zerr, S., Li, Y., "Using microtasks to crowdsource DBpedia entity classification: a study in workflow design", Semant, Web J. (2016)
40. Pablo N. Mendes, Max Jakob, and Christian Bizer, 2012, "Dbpedia for nlp: A multilingual cross-domain knowledge base", In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)
41. Research supported by NSF grants IIS-9801638 and ITR IIS-0085836 and an ONR MURI Award
42. Tibshirani R., "Regression shrinkage and selection via the Lasso", J. R. Stat. Soc. Ser. B, 1996, vol. 58, pp. 267-288
43. Hoerl A.E., Kennard R.W., "Ridge regression: biased estimation for non-orthogonal problems", Technometrics, 1970, vol. 12, pp. 55-67
44. Zou H., Hastie T., "Regularization and variable selection via the elastic net", J. R. Stat. Soc. Ser. B , 2005, vol. 67, pp. 301-301
46. Yang-Yin Lee, Yong-Siang Shih, Chih-Chieh Shao, Yen-Pin Chiu, Hsin-Hsi Chen et al, "Question Type Analysis for Question-Answering Applications in Education", (2014). Proceedings of the 22nd International Conference on Computers in Education
47. ALAA MOHASSEB, MOHAMED BADER-EL-DEN and MIHAELA COCEA, "Question Categorization and Classification using Grammar Based Approach", Information Processing & Management, Vol. 54, Issue 6, pp. 1228-1243, 2018

AUTHORS PROFILE



Vaibhav Mishra received his B.Sc. Degree in 2001 in Physics and Math. He completed his M.C.A. in 2005 from Kanpur University. He has 14 years of industry work experience in Embedded Technologies. Currently he is pursuing his Ph.D. from Mewar University, Rajasthan. His Research Interest areas include Mobile Framework , Natural Language Processing, Ontology , Machine Learning , Deep Learning and Internet of Thing (IoT).



Dr. Nitesh Khilwani received his B.Tech. degree in 2007 in Materials & Metallurgy, Ph.D. in 2010 from Loughborough University, UK. He has 9 years of industry research experience in Web Semantics, Ontology, Knowledge Management & Text Analytics. He has published 8 Papers in national & international journals. His Research Interest areas are: Semantic web technologies and ontology development, Data management and analytics, Semantic web technologies and ontology development, Text analysis and mining using natural language processing and machine learning, Algorithms and AI Techniques.