International Journal of Information Science and Management
Vol. 20, No. 1, 2022, 75-90
https://dorl.net/dor/ 20.1001.1.20088302.2022.20.1.5.8

Original Research

An Open Domain Factoid QA Framework with Improved Validation Techniques

Emmanuel Adebisi
Ph.D. student, Department of Computer Science
Federal University of Technology, Akure, Nigeria
emmanueladebisi8@gmail.com
ORCID iD: https://orcid.org/0000-0002-4855-2870

Bolanle Adefowoke Ojokoh
Professor, Department of Computer Science,
Federal University of Technology, Akure, Nigeria
bolanleojokoh@yahoo.com
ORCID iD: https://orcid.org/0000-0002-4995-6025

Folasade Olubusola Isinkaye
Senior Lecturer, Department of Computer Science,
Ekiti State University, Ado-Ekiti, Nigeria
Corresponding Author: folasade.isinkaye@eksu.edu.ng
ORCID iD: https://orcid.org/0000-0001-5012-7446

Received: 25 April 2020
Accepted: 10 March 2021

Abstract

The generic Question Answering (QA) framework processes questions by querying a knowledge base and extracting answers from retrieved passages using various Natural Language Processing techniques. The problem is validating whether the retrieved passages from the passage retrieval module contain expected answers to asked questions. Besides, extraction based on lexical and syntactic similarities alone is not enough coverage for scoring the correct answers in a QA framework. Therefore, this work aims to infuse validation techniques into the QA framework. Four similarity scores (Word Form (WF), Word Order (WO), Distance (DIST), and Semantic Similarity (SemSim)) were implemented for Answer Extraction. Instant snippets returned by the Google search engine were used as a corpus to generate candidate answer sets. On a dataset of 1370 factoid questions, the proposed method achieved an accuracy of 77.71%, precision of 77.91%, recall of 91.37%, and F1-measure of 91.37%. The results show that the inclusion of the validation techniques helps reduce the time spent by the system in analyzing passages without possible answers. The proposed system could be adapted for automatic QA Systems and grading factoid computer-based tests.

Keywords: Factoid questions, Question answering, Semantics, Open domain, Textual Entailment

Introduction

QA system is an effective automatic technique that provides appropriate answers to the queries issued by humans in a natural language pattern (Dwivedi & Singh, 2013). This can be achieved either by using a set of natural language documents or well-defined databases (Ansari, Maknojia & Shaikh, 2016). This research domain emanated from the field of Information Extraction and Retrieval (IE & IR), Pattern Matching (PM), as well as Natural Language Processing (NLP) (Preena & Shibily, 2019). Two forms of QA systems exist open-domain and closed domain QA systems (Mishra &Jain, 2016; Bouziane, Bouchiha, Doumi, & Malki, 2015). Open-domain QA systems are not limited to a specific domain; instead, they

depend on universal ontology and world knowledge (Allam & Haggag, 2012; Kumar & Sharma, 2014; Sun, Dhingra, Zaheer, Mazaitis, Salakhutdinov & Cohen, 2018). Closed domain QA systems are concerned with questions associated with specific domains (Reddy & Madhavi, 2017). Hence, they create and use formalized natural language processing systems for domain-specific ontology where answers could be searched within domain-specific document collections (Gupta & Gupta, 2012). A QA system's three key components comprise analysis of questions, analysis, and choosing answers (Aroussi, El Habib & El Beqqali, 2016). While question analysis involves breaking questions down, classifying, and reconstructing them, document analysis processes the given documents to discover correct answers (Ozyurt, Bandrowski & Grethe, 2020).

QA systems could be categorized according to the styles of questions issued by individual users. Some types of the categories of questions are (i) factoid (Sharma, Kulkarni, Pranavi, Bayomi, Nyberg & Mitamura 2018), (ii) list (iii) hypothetical (iv) confirmation, (v) causal, and (vi) complex questions. Factoid questions are direct and concise and are built on facts that need answers in comprehensible small phrases or sentences (Ojokoh & Adebisi, 2019). List questions require either a list of entities or a list of facts in response to questions (Wasim, Mahmood, Asim & Khan, 2019). Hypothetical questions try to find facts to gain a comprehensive insight into an imaginary event (Mishra & Jain, 2016). Causal questions need explanations concerning entities (ibid); Confirmation questions demand a yes or no answer; they require an inference technique, universal details to produce answers (Shinde, Singh, Shitole, Singh, Sumit & Harale, 2019). Complex questions are tough to answer (Saha, Pahuja, Khapra, Sankaranarayanan & Chandar, 2018); they frequently require deducing and integrating information from numerous documents to get different nuggets as answers.

QA has been broadly considered in computer science using information retrieval approaches. During the process of Information Retrieval (IR), some sets of retrieved paragraphs do not often contain the required answer, which usually leads to the generation of wrong candidate answers. Such errors from the Information Retrieval phase of the QA system always make the system produce wrong answers. Similarly, information retrieval systems that are majorly based on keyword matching are limited by their naïve view of the user's goal (Kanthavel, Maheswari & Padmanabhan, 2013). Rodrigo Perez-Iglesias, Penas & Araujo (2010) tried to improve Information Retrieval (IR) output in Question Answering Frameworks by expanding the IR search with extra information about the question. To increase the recall of the IR engine and the ranking given to promising candidate paragraphs, they introduced strict validation measures to remove paragraphs that showed evidence of incorrect answers. Their result showed that by infusing the validation method, some retrieved paragraphs with no correlation with asked questions from the IR system could be removed. The validation method targeted lexical entailment; hence, syntactic or semantic variations were dropped. Equally, recent answers on Answer Extraction proposed answer ranking based on similar features in both questions and retrieved answer sentences, such as the alignment of question terms and lexical distance between words (Lu et al., 2019). Le et al. (2016) merged different features to score candidate answers. These features were based on the lexical and syntactic similarities contained in both question and answer sentences. Their analysis showed some relationships between the pairs that required some semantic scorers to capture. Consequently, there is still a need to cover the semantic relationship between question and answer pairs to improve the quality of the extracted answers.

Therefore, this work aimed at infusing validation techniques into the QA framework to validate paragraphs beyond the lexical level and capture semantic variations in answer extraction techniques based on similarity features.

Related Works

Most times, users on the internet are interested in extracting precise and relevant information about factoid questions to retrieve a complete document on the Web (Olvera-Lobo & Gutiérrez-Artacho, 2010). Question Answering Systems (QAS) have proved effective in handling the task (Bouziane et al., 2015). QAS is an approach that extracts answers to users' questions specified in natural language from a massive databank of documents. QAS accepts their input questions in natural language, and their output can be an appropriate answer observed within a text or little text bits containing the answers (Alturani & Hamzah, 2018). Most of the works carried out in this domain were founded on NLP techniques (Morrissey & Zhao, 2013).

Cases of open-domain QA systems include the following: Webclopedia (Hovy, Gerber, Hermjakob, Junk & Lin, 2000), Mulder (Kwok, Etzioni & Weld, 2001), QAPD (Abdi & Ahmad, 2018), and the ones also in Lin et al. (2018) and Ryu, Myung-Gil & Hyun-Ki (2014). In addition, Wang et al. (2018) proposed and evaluated R3, an open-domain QA structure that incorporates IR and deep learning via Ranker and Reader. The IR module extracts the top-N passages based on the question given by the user. The Ranker and Reader modules were trained together by reinforcement learning to optimize extracting the ground-truth answer from the retrieved passages directly. In the work of Das, Dhuliawala, Zaheer, M & McCallum (2019), a framework for an open-domain QA where the retriever and the reader iteratively relate with each other was established. The retriever uses a fast nearest neighbor search to scale to corpora containing many paragraphs. A gated recurrent unit updates the query at each step according to the reader's state, and the reformulated query was used to re-order the paragraphs by the retriever. Likewise, Lee, Chang & Toutanova (2019) proposed an Open-Retrieval Question Answering system (ORQA) made possible by pre-training the retrieval with an Inverse Cloze Task (ICT), the retriever and reader were jointly learned end-to-end using only question-answer pairs and without an IR system.

The closed-domain QA system usually harnesses domain knowledge to build formalized ontology (Cuteri, Reale & Ricca, 2019). However, this type of QAS has very high accuracy, but it involves extensive language processing (Kamdi & Agrawal, 2015). Some examples of closed domain QA systems include VQA-LOL (Gokhale, Banerjee, Baral & Yang, 2020), Start (Katz et al., 2002), Naluri (Wong, 2007), and Webcoop (Benamara, 2004). QA systems developed by Katz et al. (2002), Chung et al. (2004), and Mishra & Jain (2010) extend their knowledge base by harnessing the huge information resident on the web for providing answers to questions via linguistic and rule-based approaches.

Textual entailment (TE) could be integrated into question answering to improve its tasks since TE could help decide if a natural language hypothesis could be deduced from a specific piece of natural language text (Shivade, Raghavan & Patwardhan, 2016). Harabagiu and Hickl (2006) proved that computational systems could be built to identify textual entailment, which can also be utilized to enhance the accuracy of current open-domain question answering (QA) systems. Abacha and Demner-Fushman (2019) examined question entailment about the medical domain and the usefulness of the end-to-end RQE-based QA approach by

evaluating the relevance of retrieved answers. Khot, Sabharwal & Clark (2018), demonstrated that linguistic structure could be exploited to discover the entailment association in datasets. Therefore, this work proposed an improved QA framework that authenticates entailment between asked questions and retrieved paragraphs using the textual entailment technique.

Methodology

The framework for the QA system is presented in Figure 1. The system takes factoid questions, and passages crawled by Google as input. The questions are pre-processed in three stages: the question-preprocessing stage, paragraph-validation stage, and answer-extraction stage. The operations of each of the stages are explained in the following sections.

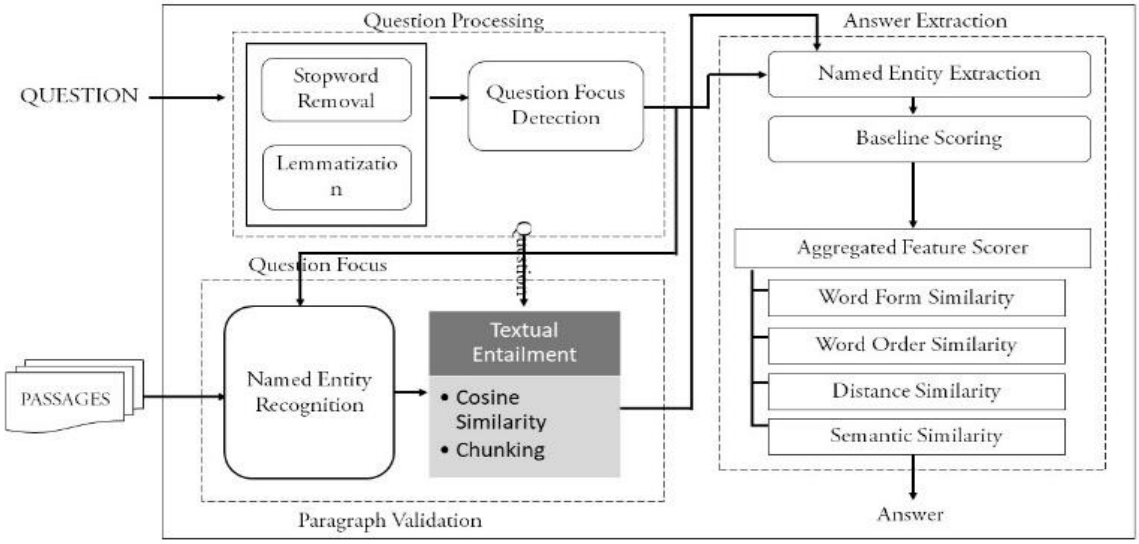

Figure 1: The Textual Entailment QA Framework

Question - preprocessing Stage

In the question-preprocessing stage, the input question and retrieved passages are prepared for use in the subsequent stages of the algorithm. In addition, stop words are removed from the query, and each question term is lemmatized. Part of Speech tagging is carried out on both the question and retrieved paragraphs for proper annotation using the Stanford POS tagger in English (Pakray, Neogi, Bhaskar, Poria, Bandyopadhyay & Gelbukh, 2011). This helps to identify the central question further to categorize the likely answer type to the asked question. Then named entity recognition is carried out on both the question and retrieved paragraphs to identify and tag named entities contained in both correctly. A named entity recognizer was developed using the Natural language toolkit to tag proper nouns, numeric expressions, and temporal expressions of each question and candidate paragraph. Besides, information about the type of the proper noun is included. For proper nouns, we have types PERSON, ORGANIZATION, and LOCATION. This information is used in the first step of paragraph validation to check if the retrieved paragraph contains the expected named entity, and it is also used in the answer extraction module to retrieve candidate answers contained in retrieved paragraphs. Table 1 shows a typical illustration of the named entity annotation as used in work.

Table 1
Named Entity Annotation

S/N	Text	Annotated Text
1	The best part of being a Yankee fan is living in New York	The best part of been a <organization>Yankee</organization> fan is living in <location>New York</location>
2	Albert Einstein was a German-born theoretical physicist who developed the theory of relativity.	<person>Albert Einstein</person> was a German-born theoretical physicist who developed the theory of relativity

Paragraph-validation Stage

The focus here is to eliminate paragraphs that do not satisfy some constraints enforced by a question. Then, aggregated similarity scores are used to classify each retrieved paragraph according to entailment. The extracted paragraphs and questions are re-formulated as a textual entailment problem adopting the methods of Novotný (2018) and Zhu, Wong and Chao (2014) for cosine similarity and chunking measure computation, respectively. Therefore, to calculate the entailment scores of the retrieved paragraphs against the asked question (serving as A and B, respectively in Cosine Similarity, and T and H respectively, in Chunking Scorer), we used the aggregated scores of the Cosine Similarity scorer in equation (1) and Chunking scorer in equation (2).

The Cosine Similarity θ is computed as the vectors between two attributes A and B, as:

$$Cos\ (\theta\ ) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_i^n (A_i)^2} \times \sqrt{\sum_i^n (B_i)^2}} \quad (1)$$

Chunking measure determines the number of similar chunks between the T-H pairs was computed as:

$$chunking(T, H) = \frac{1}{m} \sum_{n=1}^{m} simChunk(tn, hn) \quad (2)$$

m is the number of chunks in T, tn is the n chunk tag and content in the same order, and $simChunk(tn, hn) = 1$ if the content and annotation of the chunk are the same, and $simChunk(tn, hn) = 0.5$ if the content of the chunk is different but the chunk tag is still the same.

After building a vector of similarity scores with these two similarity measures to classify entailment, the entailed paragraphs and the expected answer types were supplied to the next phase, the Answer Extraction stage.

Answer Extraction Stage

Name entities identified in the entailed paragraphs stage were extracted to create an answer set in the answer extraction stage. Generally, given the question **q** and its candidate answer set $A_q = \{a_1, a_2, \ldots . a_n\}$, the best answer $a_1 \in A_q$ selected from the answer set is computed as:

$$A_1 = \arg\max_{a \in Aq} Score_{q,a} \quad (4)$$

Where, $\text{Score}_{q,a}$ signifies the corresponding degree of answer a and question q. Figure 2 describes the answer extraction process. The answers from the top of the flow are the extracted named entities from the retrieved and validated paragraphs. The candidate answers and questions are input into the similarity calculation block, where four feature scoring approaches were applied. The aggregated similarity scorers were further used to rank the candidate answers in the answer ranking block. Finally, the highest-ranked answer was returned. The following scores were used to perform similarity calculations to manage the syntactic and semantic diversities. Word form similarity, Word order similarity (Pawar & Mago, 2018), Distance Similarity, and Semantic Similarity (Blagec, Xu, Agibetov & Samwald, 2019). They are described as follows:

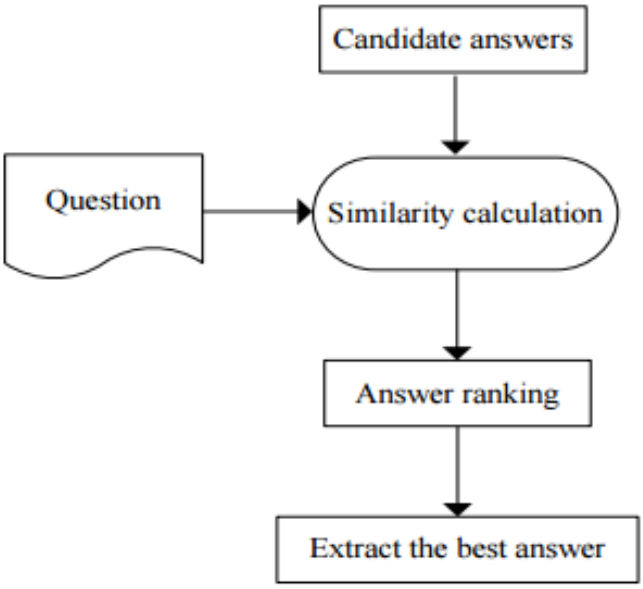

Figure 2: Flow of Answer Extraction

Word form (WF) similarity measured the number of common words in question and answer sentences. The score is calculated as in equation (5) (Wali, Gargouri & Hamadou, 2017):

$$score_{CA} = \frac{Same(q,a)}{Word(q) + Word(a) - Same(q,a)} \quad (5)$$

This scorer calculates the number of exact keywords in answer a and the question q. Word(q) denotes the number of keywords in question q, and Word(a) signifies the number of keywords in answer a. In word order (WO) similarity, the candidate answer is scored by matching the word order similarity between the question and answer sentences. The word order are compared based on important terms that make up the question. Equation (6) (Li et al., 2004) computes the score as:

$$score_{CA} = \frac{length_Q}{length_M} \quad (6)$$

Here, length_Q refers to the length of the question. The denominator length_Mrefers to the number of aligned hot terms in the question as shown in equation (7).

$$length_M = \frac{length_E}{count_E} \tag{7}$$

$length_E$ denotes the total length of alignment text segments and $count_E$ is the number of the alignment text segment. Distance similarity (DIST) measures similarity by calculating the distance of the question keywords found in the candidate answer text from the candidate answer itself. Equation (8) can be used to calculate the score.

$$score_{CA} = \frac{1}{|P_{CA} - P_{HT}|} \tag{8}$$

In the equation, P_{CA} and P_{HT} respectively denotes the position of the candidate answer and a key question term. The denominator $|P_{CA} - P_{HT}|$ is the absolute value of the distance between them. Equation (8) calculates the minimum distance between an answer candidate and a question term. More than one possible answer can be contained in a paragraph; the maximum distance calculation based on terms can help identify the named entity closest to any question term. The sum of the total distance of all question terms from the answer candidate gives us the maximum distance calculation for that answer candidate as shown in Equation (9):

$$score_{CA} = \sum_{i=1}^{n} = \frac{1}{|P_{CA} - P_{HT(i)}|} \tag{9}$$

If a keyword appears twice in the answer string, the one with the shortest distance from the candidate answer is considered. Finally, semantic similarity (SIM) Semantic similarity calculation is based on word semantics as used in WordNet. This is structured as a hierarchical net where words are organized into synonym sets (synsets) with semantic pointers to other synsets. Each word is expressed by a set of synsets (considered concepts) Syn(w) representing the possible meanings of the concerned word, w. The taxonomy "is a" is mainly used to measure the degree of similarity between concepts or words. Then, the maximum function was used to estimate the semantic similarity rate between words as in Equation (10) (Song, Feng, Gu & Wenyin, 2007; Pilehvar & Navigli, 2015):

$$SemSim(w1, w2) = \max(c1, c2)_{\in Syn(w1)\times Syn(w2)}\, Sim(c1, c2) \tag{10}$$

Also, given the question q and the answer a, q contains these keywords $w_{11}, w_{12}, \ldots . w_{1n,}$a contains these keywords $w_{21}, w_{22}, \ldots . w_{2m}$. So, the similarity between the keyword $w_{1i}(1 \le i \le n)$ and $w_{2j}(1 \le j \le m)$can be expressed as $Sim(w_{1i}, w_{2j})$. Semantic similarity between the question q and the answer a begins by finding the maximum word similarity score for each word in q with words in the same part of speech class in a. It applies the same procedure for each word in a with words in the same part of speech class in q. This is calculated using Equation (11):

$$\begin{aligned} SemSim(q, a) = \frac{1}{2}\Big(\frac{1}{n}\sum_{i=1}^{n} \max\{Sim(w_{1i}, w_{2j}) | 1 \le j \le m\} \\ + \frac{1}{m}\sum_{i=1}^{m} \max\{Sim(w_{1i}, w_{2j}) | 1 \le i \le n\}\Big) \end{aligned} \tag{11}$$

Experimental Description

This work used freely available Natural language tools like Natural Language Toolkit (NLTK), Scikit-Learn, and Stanford Named Entity Recognizer. WordNet and crawled Google results were used as the system's knowledge base. As a system performance benchmark, a dataset of 1370 testing open-domain factoid questions was used. As the answer source**,** returned results from Google search were used as the corpus to generate candidate answer sets. Returned results can offer valuable information because they have been ranked based on the PageRank algorithm. Some results were also returned during a search that contained no named entity; the link to such a result was visited and crawled for possible answers as long as the link fell within the top ten ranked Google results.

Evaluation Metrics

Standard evaluation metrics that are usually used for evaluating QA systems were used in this work. They include Accuracy metric, Recall, Precision, and F-measure (Soares and Parriras, 2020). The metrics are defined as follows: If CR represents the accurate positive answers, NP, true negative answers, WR, false-positive answers and NA false negative answers to questions that are embedded in a document collection, the accuracy can be expressed as in Equation 12:

$$Accuracy = \frac{NP}{NP+WR+CR+NA} \tag{12}$$

For recall, If CR represents the set of an appropriate answer to the questions that are embedded in a document collection and NA represents the false-negative answers seen by the system, Then the Recall, *R* of the answer set can be expressed as equation (13):

$$Recall = \frac{CR}{CR+NA} \tag{13}$$

Precision, on the other hand, is expressed as equation (14):

$$Precision = \frac{CR}{CR+WR} \tag{14}$$

Finally, F-measure is computed as equation (15):

$$F - measure = 2 * \frac{Precision*Recall}{Precision+Recall} \tag{15}$$

Where equal weighting is assigned to Precision and Recall.

Results

The results were obtained for 1370 open-domain factoid questions. Google's top 10 links were crawled to create a corpus for answering the questions for each question. A total of 33,700 paragraphs were used to answer the evaluation questions. Table 2 depicts a summary of the evaluation questions on the proposed QA framework.

Table 2
Evaluation of Questions Summary on Our QA Framework

	Total	CR	NA	WR
Who	223	183	14	26
When	129	95	9	25
Where	98	80	0	18
What	583	534	34	15
How	156	105	26	25
Other	181	129	29	23

The 1370 questions were further divided into question classes "who, what, when, where, how, and other," as shown in Table 2. The results obtained from the system are the number of correctly answered questions in each class, wrongly answered questions in each class, and finally, questions to which answers could not be found. The questions without answers stem from validation during the paragraph retrieval module. The "No answers" would have been due to the inability of the validation system to validate entailment between retrieved paragraphs and asked questions. As long as a paragraph is passed down from the paragraph retrieval phase that contains the probably named entity, the answer extraction module will always produce an answer, either correct or incorrect. Table 3 shows the result from evaluating the 1370 questions with the NER using Accuracy, Precision, Recall, and F-Measure.

Table 3
Metric Scores of the NER

Question Type	Accuracy	Precision	Recall	F-Measure
Who	0.82063	0.87560	0.92893	0.90148
When	0.73643	0.79167	0.91346	0.84821
Where	0.81633	0.81633	1.00000	0.89888
What	0.91595	0.97268	0.94014	0.95613
How	0.67308	0.80769	0.80153	0.80460
Others	0.71271	0.84868	0.81646	0.83226

As seen in Table 3, the system answered above 50% of the questions from each class. This is due to the structure of most of the retrieved paragraphs and the performance of the Named Entity (NE) Recognizer. The distance scorer outperformed all other metrics used for scoring named entities in situations where more than one NE is found in a paragraph. Some scorers attached the same score for two different named entities in the same paragraph, and only one answer is expected. The distance scorer calculated the named entity that is closest to answering the question asked. The chart in Figure 3 shows the summary of answers extracted from the NER in terms of Accuracy, Precision, Recall, and F-Measure.

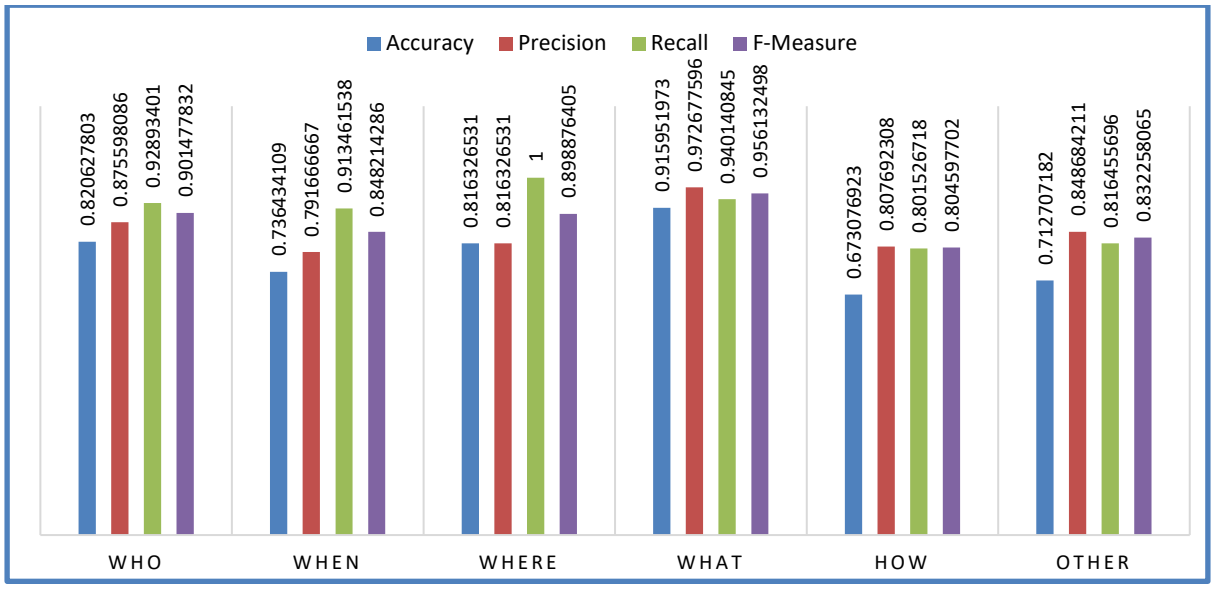

Figure. 3: NER Evaluation Result Chart

Table 4 compares the accuracy of the retrieved answers according to the similarity features used. This table shows the effect of including semantic similarity scorer in the Answer Extraction process.

Table 4.
Comparison between Accuracy of Similarity Scorers

Question Type	WF+WO+DIST	WF+WO+DIST+SIM
Who	0.762332	0.817628
When	0.697674	0.736434
Where	0.790918	0.811327
What	0.857633	0.915952
How	0.574923	0.671077
Other	0.660983	0.710707

Figure 4 shows the similarity scores aggregation (with and without the semantic scorer). The answer extraction accuracy increased with the inclusion of the semantic similarity scorer. The inclusion of this scorer enabled the recognition of answers embedded in paragraphs with a low lexical or syntactic relationship with asked questions. This is because the semantic similarity metric score answers

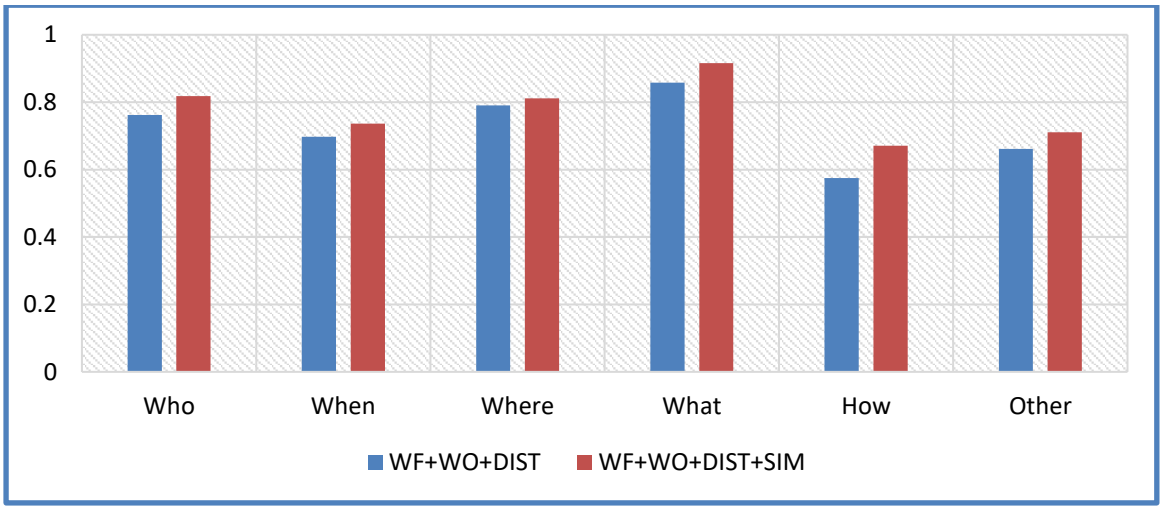

Figure. 4: Similarity Comparison Chart

By finding the maximum word similarity score for each word in q with words in the same part of speech class in a. It then applies the same procedure for each word in a with words in the same part of speech class in q.

We also compared the accuracy of generating answers based on similarity scores (our system) and generating answers according to the frequency of words. The other method ranks candidate answers using the frequency of occurrence of words in the retrieved paragraphs. The candidate answers contain all corresponding name entities taken from the paragraphs. Table 5 compares the accuracy of answers generated based on their frequency of occurrence and answers generated based on similarity scores.

Table 5.
Comparison of Accuracy of Answer Extraction Using Frequency Scoring and Similarity Scoring

Question Type	Frequency Scoring	Similarity Scoring
Who	0.714489	0.817628
When	0.620155	0.736434
Where	0.709286	0.811327
What	0.600343	0.915952
How	0.639026	0.671077
Other	0.495238	0.710707

The chart in Figure 5 compares the accuracy of answer extraction using Frequency scoring (frequency of words) and Similarity scoring. In order to evaluate the performance of our system, we compared our Aggregated Similarity Scorer for answer extraction against another Similarity scorer (Merged Feature Scorer (MFS)) (Lee et al., 2019) on a dataset of 400 testing open-domain factoid questions which belong to the main TREC QA tracks.

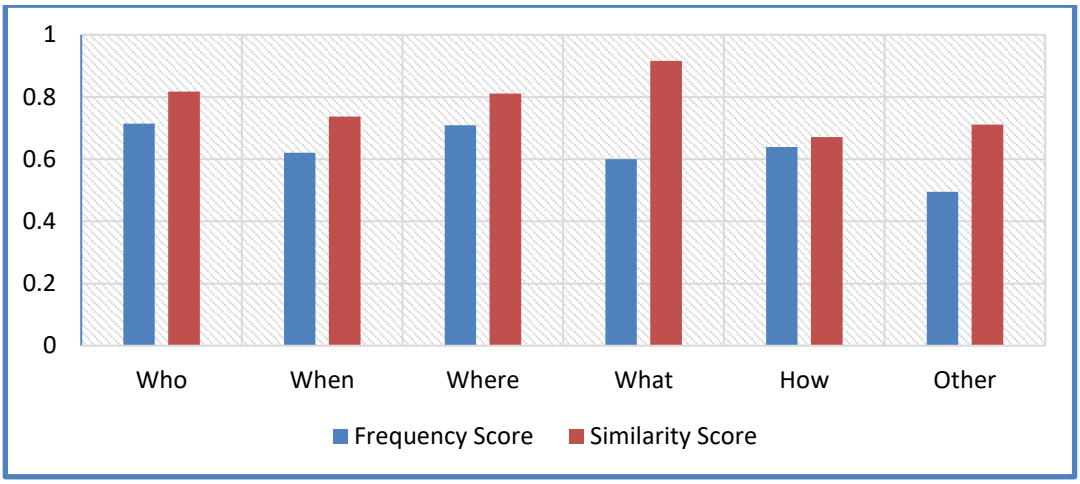

Figure 5: Accuracy of Frequency and Similarity score

The paragraphs crawled from Google were used, and the results are depicted in Table 6. Fig. 6 also depicts the comparative analysis of our proposed system with MFS.

Table 6
Comparative Analysis of the Accuracy of Our Method with another Similarity Scorer Method.

Class	Merged Feature scorer(Lee et al., 2019)	Our System
Person	0.887500	0.925000
Location	0.912500	0.937500
Organization	0.862500	0.887500
Number	0.925000	0.925000
Date	0.900000	0.912500

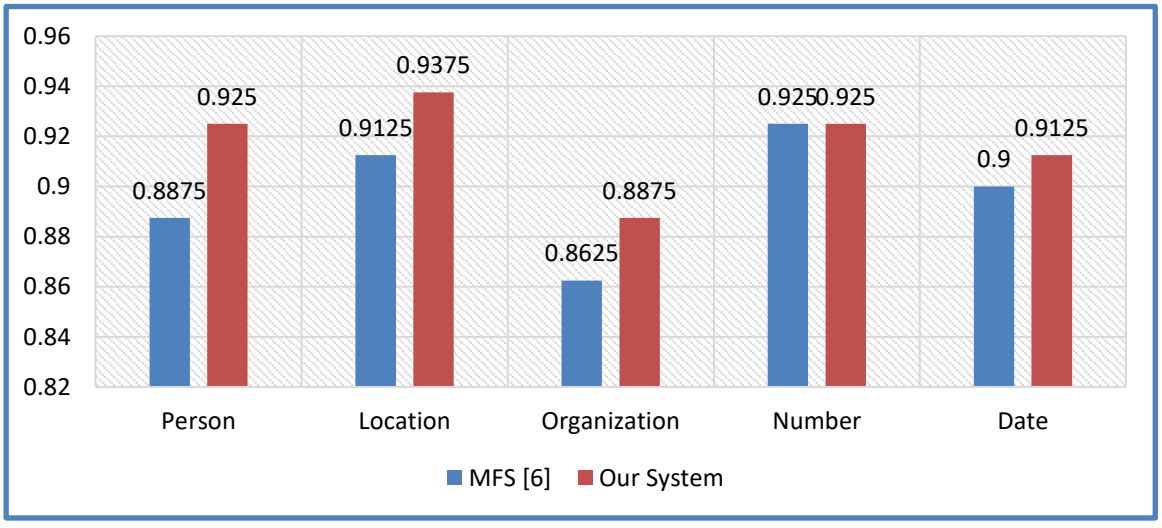

Figure 6: Comparative Analysis Chart

Discussions

The results obtained showed that the consideration of validation in the paragraph retrieval phase reduced the possibility of error in the downstream module (Answer Extraction). The merged scores from similarity features help improve the accuracy of the answer extraction module. This, therefore, proves the results from previous research on the merits of including validation steps in question answering system (Rodrigo et al., 2010) and the use of similarity features in scoring answers (Juan et al., 2016; Xianfeng et al., 2016; Xianfeng and Pengfei, 2016). Thus, the features considered showed significant relevance of use over one another (word form, word order, word distance, and semantic similarity) in different scenarios, but aggregating all the scores together helps cover the scenarios where some similarity features are not performing well. For example, the distance metric has a higher probability of scoring the correct answers when more than one candidate's answer is contained in a paragraph. Previous works on semantics in Question Answering have proved that semantic considerations have a higher probability of scoring the correct answers (Xu et al., 2014; Aouicha et al., 2016), as this work also establishes. Word form and order perform better when the asked question and the retrieved paragraphs bear close resemblance in word composition and alignment.

Conclusion

A Question Answering framework that promises better answer quality by including Paragraph Validation via Textual Entailment on retrieved paragraphs from the Paragraph Retrieval module was presented in this work. This work also demonstrated the need to consider the semantics of asked questions and retrieved paragraphs. The effect of computing similarity at the semantic level and its improvement effect on the output of the QA system were also considered. Evaluation with an existing system shows that the proposed system improves the accuracy of Factoid Question Answering systems. The use of instant material (Google ranked documents) and its closeness to asked questions also help increase the accuracy of the QA system. Therefore, the proposed system could be adapted for using automatic factoid QA Systems.

References

Abacha, A. B. & Demner-Fushman, D. (2019). A question-entailment approach to question answering. *BMC bioinformatics*, 20, 511. https://doi.org/10.1186/s12859-019-3119-4

Abdi, A., Idris, N., Ahmad, Z. (2018). QAPD: An ontology-based question answering system in the physics domain. *Soft Computing*, 22(1), 213-230. https://doi.org/10.1007/s00500-016-2328-2

Allam, A.M.N. & Haggag, M.H. (2012). The question answering systems: A survey. *International Journal of Research and Reviews in Information Sciences (IJRRIS)*, 2(3).

Alturani, I.M.I. & Hamzah, M.P.B. (2018). A new approach for open-domain question answering system. *International Journal of Computer Science and Network Security*, 18(6), 100-103.

Ansari, A., Maknojia, M. & Shaikh, A. (2016). Intelligent question answering system based on artificial neural network. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)* (pp. 758-763). IEEE.

Aroussi, S.A., El Habib, N. & El Beqqali, O. (2016). October. Improving question answering systems by using the explicit semantic analysis method. In *2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA)* (pp. 1-6). IEEE.

Benamara, F. (2004). Cooperative question answering in restricted domains: the WEBCOOP experiment. In *Proceedings of the Conference on Question Answering in Restricted Domains, 42nd Annual Meeting of the Association for Computational Linguistics*, Barcelona, Spain (pp. 31-38).

Blagec, K., Xu, H., Agibetov, A. & Samwald, M. (2019). Neural sentence embedding models for semantic similarity estimation in the biomedical domain. *BMC bioinformatics*, 20(1), 178. https://doi.org/10.1186/s12859-019-2789-2

Bouziane, A., Bouchiha, D., Doumi, N. & Malki, M. (2015). Question answering systems: Survey and trends. *Procedia Computer Science*, 73, 366-375. https://doi.org/10.1016/j.procs.2015.12.005

Cuteri, B., Reale, K., and Ricca, F. (2019). A logic-based question answering system for cultural heritage. In *European Conference on Logics in Artificial Intelligence* (pp. 526-541). Springer, Cham.

Das, R., Dhuliawala, S., Zaheer, M., McCallum, A. (2019). Multi-step retriever-reader interaction for scalable open-domain question answering. *arXiv preprint arXiv:1905.05733*.

Dwivedi, S.K. & Singh, V. (2013). Research and reviews in question answering system. *Procedia Technology*, 10, 417-424. https://doi.org/10.1016/j.protcy.2013.12.378

Gokhale, T., Banerjee, P., Baral, C. & Yang, Y. (2020). VQA-LOL: Visual question answering under the lens of logic. *arXiv preprint arXiv:2002.08325*.

Gupta, P. & Gupta, V. (2012). A survey of text question answering techniques. *International Journal of Computer Applications*, 53(4), 1-8. https://doi.org/10.5120/8406-2030

Hovy, E., Gerber, L., Hermjakob, U., Junk, M., Lin, C.Y. (2000). Question answering in webclopedia. In *TREC* (Vol. 52, pp. 53-56).

Kamdi, R.P. & Agrawal, A.J. (2015). Keywords based closed domain question answering system for Indian penal code sections and Indian amendment laws. *International Journal of Intelligent Systems and Applications*, 7(12), 57-67. https://doi.org/10.5815/ijisa.2015.12.06

Kanthavel, R., Maheswari, K & Padmanabhan, N. (2013). Information retrieval based on semantic matching approach in web service discovery. *International Journal of Computer Applications*, 64(16), 54-56.

Katz, B., Felshin, S., Yuret, D., Ibrahim, A., Lin, J., Marton, G., McFarland, A.J. & Temelkuran, B. (2002). Omnibase: Uniform access to heterogeneous data for question answering. In *International Conference on Application of Natural Language to Information Systems*, (pp. 230-234). Springer.

Khot, T., Sabharwal, A., Clark, P. (2018). SciTail: A textual entailment dataset from science question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kumar, P., Goel, R.K. & Sharma, P.S. (2014). A new architecture of automatic question answering system using ontology. *International Journal of Computer Applications*, 97(20), 1-4. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.681.843&rep=rep1&type=pdf

Kwok, C.C., Etzioni, O., Weld, D.S. (2001). Scaling question answering to the web. In *Proceedings of the 10th international conference on World Wide Web, (pp. 150-161).*

Lee, K., Chang, M.W., Toutanova, K. (2019). Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*.

Li, Y., Bandar, Z., McLean, D. and O'shea, J. (2004). A Method for Measuring Sentence Similarity and its Application to Conversational Agents. In FLAIRS Conference (pp. 820-825).

Lin, Y., Ji, H., Liu, Z., Sun, M. (2018). Denoising distantly supervised open-domain question answering. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 1736-1745).

Lu, X., Pramanik, S., Saha Roy, R., Abujabal, A., Wang, Y., & Weikum, G. (2019). Answering Complex Questions by Joining Multi-Document Evidence with Quasi Knowledge Graphs. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 105-114).

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. https://doi.org/10.1016/j.jksuci.2014.10.007

Morrissey, J. & Zhao, R. (2013, September). R/quest: A question answering system. In *International Conference on Flexible Query Answering Systems* (pp. 79-90). Springer, Berlin, Heidelberg.

Novotný, V. (2018). Implementation notes for the soft cosine measure. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 1639-1642).

Olvera-Lobo, M.D. & Gutiérrez-Artacho, J. (2010). Question-answering systems as efficient sources of terminological information: An evaluation. *Health Information & Libraries Journal*, 27(4), 268-276. https://doi.org/10.1111/j.1471-1842.2010.00896.x

Ojokoh, B. & Adebisi, E. (2019). A Review of Question Answering Systems. *Journal of Web Engineering*, 17(8), 717-758. https://doi.org/10.13052/jwe1540-9589.1785

Ozyurt, I. B., Bandrowski, A. & Grethe, J. S. (2020). Bio-AnswerFinder: A system to find answers to questions from biomedical texts. *Database: The Journal of Biological Databases and Curation*, 2020, baz137, https://doi.org/10.1093/database/baz137

Pakray, P., Neogi, S., Bhaskar, P., Poria, S., Bandyopadhyay, S. & Gelbukh, A. F. (2011). A textual entailment system using anaphora resolution. *Theory and Applications of Categories*. Retrieved from https://tac.nist.gov/publications/2011/participant.papers/JU_CSE_TAC.proceedings.pdf

Pawar, A. & Mago, V. (2018). Calculating the similarity between words and sentences using a lexical database and corpus statistics. *arXiv preprint arXiv:1802.05667*.

Pilehvar, M.T. & Navigli, R. (2015). From senses to texts: An all-in-one graph-based approach for measuring semantic similarity. *Artificial Intelligence*, 228, 95-128. https://doi.org/10.1016/j.artint.2015.07.005

Preena, M.P. & Shibily, J. (July 2019). Question answering using deep learning. In *proceedings of the International Conference on Systems, Energy & Environment (ICSEE)* 2019, GCE Kannur, Kerala. https://ssrn.com/abstract=3447734 or http://dx.doi.org/10.2139/ssrn.3447734

Reddy, A.C.O. & Madhavi, K. (2017). A survey on types of question answering system. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 19(6), 19-23. Retrieved from file:///C:/Users/Reza/AppData/Local/Temp/D1906041923.pdf

Rodrigo, A., Perez-Iglesias, J., Penas, A. & Araujo, L. (2010). A question answering system based on information retrieval and validation In *CLEF 2010 LABs and Workshops, Notebook Papers*, Padua, Italy.

Ryu, P., Myung-Gil J., Hyun-Ki K. (2014). Open domain question answering using Wikipedia-based knowledge model. *Information Processing & Management*, 50(5), 683-692. https://doi.org/10.1016/j.ipm.2014.04.007

Saha, A., Pahuja, V., Khapra, M.M., Sankaranarayanan, K. & Chandar, S. (2018). Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In *Thirty-Second AAAI Conference on Artificial Intelligence*. Retrieved from file:///C:/Users/Reza/AppData/Local/Temp/1801.10314.pdf

Sharma, V., Kulkarni, N., Pranavi, S., Bayomi, G., Nyberg, E. & Mitamura, T. (2018). BioAMA: Towards an end to end biomedical question answering system. In *Proceedings of the BioNLP 2018 workshop* (pp. 109-117). Retrieved from file:///C:/Users/Reza/AppData/Local/Temp/W18-2312.pdf

Shinde, K. Singh, A. Shitole, R. Singh, P. Sumit Harale, S. (2019). A survey on question answer system. *International Journal of Research in Engineering, Science and Management*, 2(4), 239-242.

Shivade, C., Raghavan, P., Patwardhan, S. (2016). Addressing limited data for textual entailment across domains. *arXiv preprint arXiv:1606.02638*.

Song, W., Feng, M., Gu, N. & Wenyin, L. (2007). October. Question similarity calculation for FAQ answering. In *Third International Conference on Semantics, Knowledge and Grid* (SKG 2007*)* (pp. 298-301). IEEE.

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*.

Wali, W., Gargouri, B. & Hamadou, A.B. (2017). Enhancing the sentence similarity measure by semantic and syntactico-semantic knowledge. *Vietnam Journal of Computer Science*, 4(1), 51-60. https://doi.org/10.1007/s40595-016-0080-2

Wang, S., Yu, M., Guo, X., Wang, Z., Klinger, T., Zhang, W., Chang, S., Tesauro, G., Zhou, B. & Jiang, J. (2018). R^3: Reinforced ranker-reader for open-domain question answering. In *Thirty-Second AAAI Conference on Artificial Intelligence*, (pp. 5981-5988). Retrieved from file:///C:/Users/Reza/AppData/Local/Temp/16712-76927-1-PB.pdf

Wasim, M., Mahmood, W., Asim, M. N. & Khan, M. U. (2019). Multi-label question classification for factoid and list type questions in biomedical question answering. *IEEE Access*, 7, 3882-3896. https://doi.org/10.1109/ACCESS.2018.2887165

Wong, W. (2007). Practical approach to knowledge-based question answering with natural language understanding and advanced reasoning. arXiv preprint arXiv:0707.3559.

Zhu, L., Wong, D.F. & Chao, L.S. (2014). Unsupervised chunking based on graph propagation from bilingual corpus. *The Scientific World Journal*, 401943. https://doi.org/10.1155/2014/401943