A Graph Model with Integrated Pattern and Query Based Technique for Extracting Answer to Questions in Community Question Answering System

Bolanle Ojokoh (bolanleojokoh@yahoo.com)
Federal University of Technology
Tobore Igbe
Chinese Academy of Sciences
Bamidele Afolabi
Federal University of Technology
Oladunni Daramola
Federal University of Technology

Research Article

Keywords: community question answering, answer extraction, information retrieval, question query, graph model, pattern recognition

Posted Date: April 1st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1489202/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Social Network Analysis and Mining on March 5th, 2023. See the published version at https://doi.org/10.1007/s13278-023-01046-3.

Abstract

In community question answering (CQA) systems, topical comments are very valuable to provide information for users. However, it becomes cumbersome going through all these in order to decipher the correct answers to particular questions. Hence, extracting a particular answer to a question becomes vital to avoid reading every comment in the forum. This paper is an extension of of our previous research work that extracted questions from an online forum to develop a system for answer extraction to questions. This system is based on a graph-based method by building answers for related questions using nKullback–Leibler (KL) divergence to obtain ranked answers to a question. The process of extracting answers to questions involves; question core, building question query, query-based answer extraction (QBAE), pattern-based answer extraction (PBAE), and combined answer extraction. The source data for this work were already existing data from ResearchGate, a socio-academic networking website that provides researchers the platform collaborate, ask question and offer answers to question. The performance for answer extraction for 2786 questions shows that when 80% of patterns and keywords were considered, QBAE and PBAE extracted 2765 and 2766 correct answers respectively, while the QBAE + PBAE method extracted 2782 correct answers. Also, when 90% of patterns and keywords were utilized, QBAE and PBAE extracted 2782 and 2784 correct answers, whereas the QBAE + PBAE method extracted 2786 correct answers. Our method was able to identify 229 questions without answers. Finally, the evaluation of our model reveals high-performance accuracy and precision.

1. Introduction

The current method of finding information and providing knowledge about different domain has been influenced by Community Question Answering (CQA). Users' comments that include questions and responses posted in CQA are generally organized into a hierarchical structural pattern. Advances in internet and associated technologies provide more avenues for user interaction, knowledge sharing, collaboration and other social activities. A lot of applications allowing such interactions include Twitter, Facebook, Yahoo! Answers (Y!A), and Flickr. A CQA is different from traditional question answering system (QAS) that uses documents as information sources, that relies on user-generated content to provide the answers. CQA systems allow authorized users to post questions as well as answer other users' questions. As a result, a question can have different answers posted for it. On the CQA platform too, users can to find questions that are related to their research needs and possibly search for the right answers to them. Hence, a CQA system that allows human-generated answers is expected to provide better answers that meet users' needs in comparison to a traditional QAS. (Allahbakhsh et al. 2013). Extracting high-quality question and answer pair from the CQA forum dates back to when QAS was introduced in the 1960s (Green et al. 1961). Initially, QAS implementation was restricted to some structured domains, with certain processing capability to handle natural language task (Kamp, 1984). In recent times, advances in the fields of Information Retrieval (IR), Information Extraction, computational linguistics, and internet technology, has widened research in QA to include unstructured textual documents in open domain context, and with users' collaboration (Hong et al. 2012). Advancements in this domain also include emanating evaluation forums producing large-scale research methodologies like the Text REtrieval Conference (TREC) (Voorhees 2004) and the Cross-Lingual Evaluation Forum (CLEF) (Peñas et al. 2010).

Online forums are repositories that are rich in knowledge because they contain discussions and solutions to different problems and needs posed by various users. It is, therefore, important to mine such content. Solving this text mining problem has several applications. The results produced by these platforms can be leveraged upon by online Question Answering (QA) services such as Y! A, Answers.com, and other users of automatic answer systems would benefit from the application of extracted content from collaborative discussion boards. The extracted content can serve as potential solutions or suggestions when users ask questions similar to what people have discussed on forums (Jurczyk and Agichtein 2007). Therefore, eliminating the time users wait for answers and enrich the knowledge base of the QA services as well since discussion boards have a more extended history than QA services and also own a much more considerable amount of user-generated content. These platforms could be avenues for users seeking specific expert knowledge in particular areas. (Bouguessa et al. 2008). Another application is in the use of the CQA answer quality analysis predict the answer quality. (Allahbakhsh et al. 2013; Liu 2011).

In a CQA system, users are allowed to express their thought and opinion with text. This liberty could affect the quality of the content; sometimes abusive content used by the user. This challenge makes the tasks of filtering and ranking in such systems more complicated than in other domains. Dali et al. (2009) proposed a system with restricted predetermined template structure in terms of the question types. In Liu et al. (2009), the implementation did not evaluate the hybrid method that was proposed. Mudgal et al. (2013) and Gupta and Gupta (2012) investigated specific question types which include Why, Which, Where, Where, What, Who and How (5W1H). Toba et al. (2013) research was restricted to performing feature selection. They failed to consider different varieties of the content generated by users in forum discussion. Ramprasath et al. (2016) considered pattern approach for answer extraction, but their method was dependent mainly on manual construction of patterns for a fixed set of question types. Manual construction of question query yields in poor recall (Ramprasath and Hariharan, 2016). After taking cognisance and broad review of the limitations and challenges of these abovementioned works, this research work focuses on developing a system that will tackle these shortcomings. This work is, therefore, motivated by the need to face these challenges by an integrated query and pattern-based approach to provide a high-quality QA system.

To achieve quality answers in the CQA system, we extended our previous research, where we presented a system to accommodate new question categories and a pattern classification model for identification and extraction from an online forum (Ojokoh et al. 2016). To obtain answers to extracted questions, we proposed an integrated query and pattern-based approach. Classifying questions into answer type can help provide answers to questions. The following are the main sections in our technique for extracting answers to questions: Question-Answer construction, answer extraction, quality answer inspection, and answer ranking. Question-Answer construction section provides a guide that maps a question type or class to the expected answer. The answers to a question can either be a word, list or description. The question class is obtained from the extensible question configuration module, which specifies the question category to extract from the forum. The answer extraction section is responsible for searching and extracting answers to questions. There are some input parameters from the question extraction system that is prerequisite for this section to extract answers. The answer inspection section consists of successive stages that perform analysis on the extracted answers for a given question. The answer ranking section prioritizes the answers obtained from the inspection module using a collective-statistics method. The source data for our implementation is obtained from the research gate online community forum. A greedy crawling approach is used to capture all the content to create an offline version of the forum.

2. Related Works

With the increasing data content and knowledge available on the Web, the task of search engines becomes more intelligent than before. Currently, many users do not want to go through many documents to find the information they need. Therefore, it is often preferred to give the user a concise and short answer. QA systems aim to respond to a question with the correct information. In the review of related works, we examined methodologies that have been considered in achieving answer extraction to question. The review is divided into two categories: Query-based method and pattern-based approach. The query-based methodology uses a basic format to search for answers to questions, while the pattern-based method uses a knowledge base to find answers to questions.

2.1. Query-Based Methods

Cong et al. (2008) described a web application forum for accepting discussion posts from users in a specific field, such as technology, sports, recreation, and travel. Over time, this forum consists of large user-generated content, such as text and pictures about different topics. Therefore, it is desirable that the human knowledge found in a forum can be mined and used to find answers to questions. Their research was motivated by the need to address the challenge of extracting QA pairs from online forums due to the enormous amount of valuable user-generated content. Resolving this challenge would enable extracting question-answer pairs to enrich the knowledge-based on community-based QA service and instant answer service. Another application that uses semantic graphs was implemented for a QA system by Dali et al. (2009). Besides using natural language for finding answers, a graphical representation to explain the answers is part of the system feature. The representation is achieved with a list of subject-verb-object triplets and their summaries. Furthermore, Liu et al. (2009) adopted distributional similarity and the word relevance model to identify and extract semantically related words from a large question archive, for a given queried question. The word relevance model was also adopted for non-stop word expansion in questions. The objective of the research is to analyze question collection to find semantically related questions. The model from the study was expected to improve answer extraction for related questions.

Wang et al. (2009a) were motivated by the need to address the challenges of the spam contained in user-generated content. They also aimed at improving answer ranking in CQA. Therefore, they developed an analogical reasoning-based model to find the relationship between the new QA linkages and those of some previous relevant knowledge, which contains only positive links, with the assumption that there exist varied latent links between answers and their questions. They assumed that high-quality answers imply positive links, while negative links indicate incorrect; or user-generated spam. Albaham et al. (2012) focused on retrieval in online forums that are quality based. Their research work solved the problem encountered to perform a search in a forum using three methods: keywords matching, database backend full-text search, and customized web search. The similarity score between all posts' and user query computations is done when a user query against posts matches the result of the database search. Later, it returns the highest-scoring posts to users. The heterogeneous way of online forums might limit this method.

Nakov et al. (2015) proposed two subtasks. The first subtask was identifying answers to question in a discussion thread without considering useful text that can provide more information to educate the users about the question. Also, this subtask, which was available in two languages Arabic and English language was designed to automate the answer search task. The second subtask handled special question that requires YES/NO answer. The proposed an extreme summarization exercise to extract YES/NO answers by considering all possible correct answer from the first subtask. The two subtasks were implemented on a CQA application setup based on the Qartar living forum. However, the IR component in the model was decoupled to focus on the important aspect relevant to semantic evaluation. G'omez- Adorno et al (2015) proposed a CQA answer selection technique that transformed the answers into a graph-based representation that is specific for each class, with associated features related to their lexical structure, morphology and syntax. Their research work was motivated to solve the challenging effort of going through all the possible answers to select the most accurate for every question. Their research objectives are to use textual information to predict the quality of the answers using a graph-based representation model referred to as Integrated Syntactic Graph and the soft similarity measure (cosine measure). Answers for a given question are classified into good, potential, or bad (bad, dialogue, non-English, and other) and decide whether the comprehensive answer to the question should be yes, no, or unsure, based on the individual right answers.

2.2. Pattern-based Methods

Dan et al. (2005) explored involving open-domain factoid question answering using patterns. The research utilized a pipelined structure, a scoring function, or some statistical-based methods for the answer extraction module. In another similar technique, Molla (2006) proposed a graph-based pattern approach to QA. It utilized a rule-based system in which the rules are automatically learned from a training QA corpus with annotated answers. This kind of rules was also in the work that obtained the best accuracy in the 2001 Text REtrieval Conference (Voorhees, 2001). Furthermore, Hong and Davison (2009) explored the problem of QA extraction from discussion boards. In their research, they addressed the QA task as classification problems. For question detection, they used the following features Question mark, 5WIH words, the total number of posts within one thread, authorship, N-gram. Moreover, for answer detection, they used natural language techniques such as the position of the answer post, authorship, N-gram, stop words, and query likelihood model score. Their research was motivated on how to retrieve information in the forum automatically and effectively, which is still a non-trivial task. Classification based approach was used in a ranking scheme for question-answering in discussion boards. The focus of the research was on only answer post in the discussion that is directly related to the question and did not consider post or questions within the same forum. Also, Conceptual graph formalism (CGF) was proposed by Salloum (2015) to model knowledge in documents and questions. Open NLP was adopted for natural language processing, and syntactic and semantic modelling was realized using VerbNet and WordNet.

Toba et al. (2013) proposed a framework built on a hybrid hierarchy-of-classifiers to model QA pairs, by analyzing the question type as a guide to right answer selection They result of the question analysis was used to predict the expected answer features and used to train the type-based quality classifiers to aggregate an overall answer quality score hierarchically. Moreover, the work identified features for distinguishing the quality of answers. The framework was tested on a dataset of about 50,000 QA pairs from Yahoo! Answer. Their method was designed to solve user-generated answers. This method may produce a bias in the quality of the content. Hence, there was a need to balance between efficiency and accuracy. Similarly, Ahmed et al. (2016) proposed an answer extraction module that aimed at returning only those passages of a document that correctly answer a given user question. This module contains components that can analyze the text structures (syntactical analysis) and extract the meaning too (semantic analysis), and a search engine able to use the resulting knowledge base. The purpose of their research is to develop a QA system and return answers from a knowledge base of structured data and unstructured text. Their method which adopts several techniques for extracting different types of answers including human, location, numeric, description and entity by applying named entity recognizer, regular expressions or semantic similarity between sentences and question to extract the answer.

3. Methodology

This section presents the analysis and design of the proposed CQA system using an integrated query and pattern-based approach. It highlights the system components and a description of answer extraction, which was achieved using combined techniques of query-based answer extraction (QBAE), pattern-based answer extraction (PBAE), and combined answer extraction. Also, the classification of questions into question type to provide answers to such questions was presented. In this section, we present the system architecture of our system for answer extraction and the detailed process of extracting answers to questions.

3.1 Structural Design for Answer Extraction

Figure 1 shows the visual description of answer extraction. The design of our system is structured for extracting answers to questions from blogs. It consists of four distinct modules: question-answer construction module, answer extraction, answer inspection, and answer ranking module. The design is also connected to our previous model of question extraction defined in Ojokoh et al. (2016) with three connectors A = Extensible Configuration Module for Question, B = Content Pre-processing Unit and C = Question Classification Module.

3.1.1 Question-Answer Construction Module

To get the correct answers to questions, we define what type of information is required; the appropriate response to any question is dependent on information requested by the question. The answer configuration module of our QA system provides a significant guide that maps a question type or class to the expected answers. Answers to a question can either be a word, list, or description. The question class is obtained from the extensible question configuration module, which specifies the expected question category to extract from a social media blog. This module extends a question configuration module by augmenting each question class with the expected answers; this will assist the answering system in extracting possible answers from sentences in the blog. Besides, the question category not configured in the question configuration module will not be setup in this module. The answer extraction module uses this module to filter and extract answers to a question, which is used for further structural categorization and selection to determine the correct answer for the question.

3.1.2 Answer Mining

This module represents the core for the QA system because it is responsible for searching and extracting answers to questions. There are some input parameters from the question extraction system that is prerequisite for this module to extract answers. These parameters include the blog group where the question was extracted and the extracted questions for which answers are to be determined. The blog group consists of the text (discussion) obtained from a forum thread where the question was extracted; the concept is to extract the correct answer from the same blog group where the question was extracted. The "search for an answer to questions in a blog group" performs the answer extraction. It takes a question to determine the required answer to a question from the QA construction module and the answers. The relationship between questions and scored-answers is saved for further analyses to determine how to prioritize the answers for presentation.

3.1.3 Answer Inspection

To ensure accurate answers to questions, we examine the properties of answers returned from the answer extraction module to provide the best answer as results. The inspection module consists of successive stages that perform analysis on the extracted answers for a given question. The first step, "punctuation and typo checking", performs spelling, spacing density between words, and average use of punctuation to estimate the quality of the answer. The next step, analyses and visualizes the complexity of the syntactic and semantic patterns of text in the answer sentence, which helps to determine the readability or difficulty level of the answer. Syntactic complexity utilizes naïve parameters such as the occurrence of syllables in words, the rate of characters per word and the average number of words in sentences. In contrast, the semantic complexity considers the structure of the text. The last stage is the "Grammaticality", which assesses the language-oriented features of the answers such as formality score, identification of out-of-vocabulary words to determine the grammatical quality of the answers.

3.1.4 Answer Ranking

This module prioritizes the quality answers obtained from the inspection module using a collective-statistics method. To create a rank of extracted answers, the score value for each answer obtained from the quality answer inspection module from computing different measurements. A QA map that holds key-value pair is created, where the keys refer to the likely answers to the question, and the value represents the score for the answer obtained from the measurement. Applying sorting procedure on the map's values produces a sorted list of answers, where the best quality answers for the question are located at the top of the list.

3.2 Process of Extracting Answers

In our research, answer extraction is achieved with a combined technique of pattern-based and automatic query methods. The challenge of determining questions without answers and establishing a relationship between questions for better answer retrieval, which hitherto have been omitted, are addressed. To achieve this task, we constructed different interdependent components to handle the process of taking a question and returning possible answers in a ranked list format. The following subsections describe the process of extracting answers to questions.

3.2.1 Question Core

The dynamic nature and pattern of asking questions create a challenge for retrieving answers; hence, it becomes essential to create a dynamic system capable of handling these challenges. For example, a 'who' question usually get a person or group of people (organization) as an answer. Other question keywords are not straightforward with expected answers such as: What time is the bus arriving? What city is the bus stopping at? What is the name of the driver of the bus? In another example, question keywords do not start a sentence, and two different keywords could be asking the same question: For example, when was the president of Nigeria born? and in what year was the president of Nigeria born? Considering these conditions, we define the "question core" for every category to be considered. The answer to a question is determined by the keywords in the question, which is referred to as the "question core". This concept is similar to Cooper (2000) and Mudgal et al. (2013), the question core disambiguates questions with the same keyword and emphasize the type of answer expected.

Although, Cooper (2000) and Mudgal et al. (2013), considered 5W1H (who, what, where, when, and how) keywords used for asking a question, this research is configurable, extensible and focuses on all categories of question extracted. Table 1 shows an answer category table for creating a question-answer hub, which was used in Radev et al. (2005) and Mudgal et al. (2013).

Table 1
Answer type table from Radev et al. (2005) and Mudgal et al. (2013)

Person	**Place**	**Date**	**Number**	**Definition**
Organisation	Description	Abbreviation	Known for	Rate
Length	Money	Reason	Duration	Purpose
Nominal	Others			

Creating a question-answer core requires that the system assign one or more answer categories to a question category. Table 2 shows examples of the question categories, q_m. Based on the number of question categories considered, the system provides a configuration section that assigns answer type to q_m to create the question-answer core.

$$Q_c = q_m \times \left\{ num_{j \in \mathrm{n}} \left(h_n \right) \right\} (1)$$

where Q_c is the question-answer core, h_n is answer type from Table 1, Q_c forms a Cartesian product of question category and answer type. The answer type is *n* and size of questions is *m*, where $num_{j \in \mathrm{n}}$ is a function that returns one or more answer type for a question category which is formed in Q_c.

Table 2
Question category considered for answer extraction to questions (Ojokoh et al. 2016)

ID	Question Category
1	Any
2	Ask
3	Can
4	Could
5	Do
6	Excuse me
7	Have
8	How
9	Is
10	Let
11	May
12	Might
13	Not
14	Perhaps
15	Please
16	Suppose
17	Were
18	What
19	When
20	Where
21	Who
22	Why
23	Would

Table 3
Example of generated question core

	Question category	Answer Category
1	Who	Person, Organisation
2	Where	Location
3	Which	Person, location, time

3.2.2 Query-based Answer Extraction (QBAE)

To achieve answer extraction with a query-based method, the construction of the question query string that will be used for detecting possible answers is necessary. The query string consists of keywords drawn from the question. Keywords in a question refer to significant words in the question that defines what the question is about, for example: What is the name of the driver of the bus? The keyword in the question will be the name, driver, and bus. These keywords give further clarification of the type of answer that is required. For a given question q_i, the first step is to prune q_i to remove stop words (such as the, a, and so on obtained from Fox (1992) stop word list). Next, a statistical approach is applied to extract keywords from q_i. Although, Cooper and Ruger (2001), Yen et al. (2013), Hong and Davison (2009) and Mudgal et al. (2013) applied the use of question query for question extraction, the approach for building the query was either constructed manually or poorly defined and stem words for the keywords were not considered. Stem words are root words without affixes, considering the root words will help decipher the meaning and the usage of words in the question (Davis, 2010). Also, in our research, words in a sentence are given weight value based on their position in the sentence to question category word. This methodology is adapted from Kaur and Gupta (2010); some few words are automatically added to the query because of their relevance; these include digit, uppercase words, and words not found in WordNet dictionary (Miller, 1990). This procedure is described in equations (2) and (3).

$$E\left(q_i\right) = max_{k \in \mathrm{L} \geq \beta} PW\left(w_k, R\left\{q_i\right\}\right) (2)$$

$$Q\left(q_i\right) = E\left(q_i\right) + SP\left(q_i\right) + WN\left(E\left(q_i\right)\right) (3)$$

where $E\left(q_i\right)$ is the extracted text from question q_i, w_k and L refer to a word and the number of words in q_i respectively before generating root words. R is a function that removes stop words from q_i, and PW is a function that returns a w_k if its score value is greater than or equal to β and the value of β is a configurable threshold value. The score value returned by PW increases as the distance of w_k from question category increases based on the number of words. Therefore, $max_{k \in \mathrm{L} \geq \beta}$ refers to the maximum score of w_k in q_i. $Q\left(q_i\right)$ is the question query that will be used for searching for answers for q_i. SP is a function that extracts special characters from q_i such as digits, upper case words and words not found in WordNet dictionary. WN is another function that generates available root words from $E\left(q_i\right)$.

To efficiently detect and extract answers, the IR method is cleverly adapted. Mostly used IR models include Cosine Similarity, Query Likelihood model using Dirichlet smoothing and KL-divergence language model. The performance of Query Likelihood model using the score ranking feature did not achieve impressive results. The low performance of the model is as a result of the approach of the retrieval model, which is based on finding only the important informaton relevant to the question posts. Although, in a forum, all the post related to a question may be more or less connected to the question. The major task is to rank post and find the best answer based on comparision of similarity measurement (e.g., cosine similarity). Furthermore, Lafferty and Zhai (2001) showed that the KL-divergence language model outperforms the query-likelihood model for IR. Therefore, to extract an answer from the forum for q_i, the estimation of KL-divergence score is between $Q\left(q_i\right)$ and the subsequent sentences s_i extracted from the blog group (discussion thread) where q_i was obtained.

$$.KL\left[s_i \middle\| Q\left(q_i\right)\right] = \sum_{w}^{Tq} \frac{p\left(w \mid s_i\right) \times \log\left(p\left(w \mid s_i\right)\right)}{p\left(w \mid Q\left(q_i\right)\right)} (4)$$

where $KL\left[s_i \middle\| Q\left(q_i\right)\right]$ is the KL-divergence score for sentence s_i to question q_i and w is a word in s_i. Tq is the word count in $Q\left(q_i\right)$ and $p(w|s_i)$ represents the probability of w in s_i.

A configurable threshold is set for KL-divergence score returned for s_i that determines the limit of answers returned from the computation. To show ranked answers, two other parameters need to be considered in addition to the KL-divergence score.

The first factor considered is the authorship of s_i, this is computed by considering the ratio of the replies and post for the author of s_i with the maximum value of the answers from other authors in the blog group for the same question q_i.

$$author(s_i) = \frac{(\#reply_{s_i})^2 / \#start_{s_i}}{max_{j \in K}\left((\#reply_j)^2 / \#start_j\right)} \quad (5)$$

where $author\left(s_i\right)$ is the score for the authorship of s_i, the total replies by the author s_i is represented by $\#reply_{s_i}$ and $\#start_{s_i}$ is total forum thread author of s_i initiated.$max_{j \in K}$ selects the maximum score among all the authors that replied to q_i except the author of s_i, with the highest value for the ratio of the square of replies to post started by the author. An increasing value of $author\left(q_i\right)$ indicates the degree of the authority and the tendency to ask more supporting and relevant questions to the previous questions. K is the total authors that responded to q_i, and $author\left(s_i\right)$ is given a value of 0 if $K \le 1$.

The second factor considered is the answer category form, HF_s. The purpose of this factor is to define the type of answer required for a question. The answer could be digit or number, single word, sentence, list and so on. This is derived from the question category and question query

$$HF_{s_i} = TT_q\left[Q\left(q_i\right)\right] \rightarrow Q_c (6)$$

where HF_{s_i} is the score for HF_s and TT_q is a function that analysis the relationship $(\rightarrow)$ between the content of the question query $Q\left(q_i\right)$ and the question category Q_c to determine the answer type requested for question q_i. The closer HF_s matches s_i, the higher the score for HF_{s_i}. To obtain ranked answers $answer\left(Sq_i\right)$ to a question, it requires the accumulation of the KL divergence score, authorship of the answer and the answer category from the score.

$$answer\left(Sq_i\right) = KL\left[s_i \middle\| Q\left(q_i\right)\right] + author\left(s_i\right) + HF_{s_i} (7)$$

The score value obtained for each answer, $answer\left(Sq_i\right)$ to question q_i are stored in a map that has the format of a table, capable of holding two values as a single entity, described as key-value pair. $Q_{map} < s_i, answer\left(Sq_i\right) >$ is a map that describes the relationship between each answer and their corresponding scores. A value-entity sorting is applied to $Q_{map} < s_i, answer\left(Sq_i\right) >$ which rearranges the content of the map with the values sorted in ascending order. The best answer to the question are answers at the uppermost in the map. The map is created and value-entity sorting operation is performed only if more than one answer is available.

3.2.3 Pattern-based Answer Extraction (PBAE)

The purpose of the QA system is to respond with a precise answer to the user's questions. A sentence consists of a sequence of words; the answer for a question is found within a sentence and there is a relation between core word extracted from a question and sentence containing the answer (Ramprasath et al. 2016). Where the expected text is not a sentence, but a word, the word is considered as the answer. In our work, a question-driven approach is considered for dynamically generating patterns for detecting answers for different types of question categories. The pattern used to find answers to a question will depend on the text or words found in the question. The following procedure is used for constructing a pattern for searching for an answer:

a. Get the extracted question whose answer is to be determined. Get the category the question belongs and the extracted sentences or text from the forum where the question was extracted. The question extracted from a discussion is classified into a category to determine the expected answer. An example of a question could be "*What is the name of the driver of the bus*", belongs to the category "*what*".
b. Isolate core terms from the question; these terms are important words in the question. We used Eq. (2) and the function $WN\left(E\left(q_i\right)\right)$ in Eq. (3) to obtain all the terms from the question. From the sample question in step (a), the following are some of the extracted terms: driver, drive, drove, driven motorist, car driver, bus, van, car, motor, vehicle name, title, label, brand.
c. Construct all possible combination of a pattern from the terms to get the answer(s). The terms are marked-up and stop words and determinant are not considered in the construction. The system can heuristically build multiple structural answer patterns from the terms to form the basis for the system to train, adapt and extract an answer from a sentence in a forum. An example pattern generated: *<bus > < driver > < answer>, <car > < driver > < answer>, <answer>, <vehicle > < driver > < answer>, <answer > < drove > < bus>, <driver > < bus > < answer>*, and so on
d. The search engine uses the generated pattern to search for answers to question from the text extracted from the forum. The search engine uses a recursive-superimposed technique based on the patterns generated. The technique applies the marked-up tags in each pattern over the sentence (possible answer) to uncover the answer. The type of answer returned or extracted depends on the configuration in Eq. (1) for the category of the question obtained in step (a).
e. The patterns that return matches or answers represented as $P(answer)$ are rated based on the level or percentage of the matched patterns found in the answer. These patterns that returned answers are saved, as well as the question and category for reuse in similar circumstances, it will be given priority for subsequent search operation before considering the generated patterns. Therefore, while searching for answers, the system is also learning
f. At the end of the search operation within the text from the forum thread, if no answer is found, other forum threads with constructed similarity with the question and category are considered. The relationship between questions is defined in section 3.1. If it returns *NULL* after the search process, then the question is regarded as having no answer. Figure 2 Illustrates the sequential process of extracting answers using patterns generated from the question.

3.3 Combined Answer Extraction

It is important to get an answer to a question and as well as extract accurate answers. Here, a combined QBAE and PBAE approach is constructed to find answers for a given question. Therefore, $answer\left(Sq_i\right)$ which is defined in Eq. (7)is the ranked answer returned by the QBAE method and $P(answer)$ described in section 3.2.3 is an orderly arranged answer returned by the PBAE method. To determine the answer to a question from combining these techniques involves a comparison of text quality information. The final answers for a question will be an ordered text quality score of the answers from both techniques. Where this score is not sufficient to evaluate the answer hierarchy, the extracted answer is displayed with the label of the technique appended.

$$AE = Fanswer\left(answer\left(Sq_i\right) + P(answer)\right)(8)$$

where AE is answer extracted for a question and $Fanswer$ is the function that computes the text quality score in terms of Visual Quality, Grammaticality, Readability Index for answers returned by both techniques.

3.4 Discover Questions Without Answers

To provide answers to questions from a forum requires a systematic approach that can take as input a question and comments from a forum and return one or more suitable answers. Despite the development and achievement recorded in recent times for question and answers detection and extraction, there is a need to investigate questions within the same thread to create a relationship between questions to improve answer detection, because there could be more than one question from a forum thread that share similar answers. Therefore, in this paper, a collective graph-based method was used to determine the connection between questions from the same forum thread. This technique creates a graph relationship between questions. Also, it assigns a weighted score to the relationship so that answers to a question can be used for another question.

3.5 Build Answers for Related Questions with Graph-Based Technique

The graph-based technique used in our work has been successful in Web search (Han et al. 2011), digital media review (Zoidi et al. 2015). However, it is adopted in this research to determine the relationship between questions from the same thread for answers reuse. Naturally, if a question is related (or similar) to another question with very small or no difference in the content of the questions, it is possible to apply the answers of the previous question to the latter. Given a forum thread *d*, and *Dq* is question set from *d*, a directed graph pattern defined as (C, H) is built with an approximation model $I : H \rightarrow R$, where *C* is a group of vertices and *H* is logical formed directed edges and $I(u \rightarrow c)$ is the approximation value associated with edge $u \rightarrow c$. Each question in *Dq* corresponds to a vertex in *C*. The next step will be to create the logical edge set *H* between the questions in *Dq*. Given two question q_0 and q_1, KL-divergence language model KL(q_0| q_1) is used to determine whether there will be an edge $q_0 \rightarrow q_1$. The use of KL divergence language model is motivated by its application in extracting keyphrase from different language models (Tomokiyo and Hurst, 2003), webspam identification (Martinez-Romo and Araujo, 2009), and statistical modeling of language information (Liu and Croft, 2005). Consider the following example for two extracted questions from the same forum thread q0: *I found that there are many topic modeling techniques like those of LDA, pLSA, PAM etc. Which technique is the latest and provide the best results?* q1: *what modelling technique do I use to get needed result, I am working on a new project and I need assistance*. The answers for question q1 can be applied to question q0, but not necessarily vice versa. This condition is because q0 is concerned with searching for the best modeling technique independent of the type of project. After all, the type of project could affect the choice of modeling technique.

$$KL\left(q_0 \mid q_1\right) = \sum_w p(w|q_0)\log(\frac{p(w|q_0)}{p(w|q_1)})(9)$$

where $KL\left(q_0 \mid q_1\right)$ is the KL divergence between q_0 and q_1, $p(w|q_0)$ is the probability of the word w in q_0. For two identified questions q_0 and q_1, an edge is formed between the questions if $^1/_{1 + KL(q_0|q_1)}$ is larger than or equal to a given threshold θ. It is said that q_0 is a generator of $q_1$and q_1 is an offspring of q_0. Also, an edge can be formed from q_1 to q_0 by comparing the result of $^1/_{1 + KL(q_1|q_0)}$ and θ. Moreover, one question q can be a generator of multiple other questions and q can have no generator. There are special cases when the graph propagation is turned off because there are no connections in the question graph. Figure 3 shows the directed graph relationship between questions for answer extraction. The edge between q_1 and $q_0$in the figure shows that q_0 is a generator of q_1 and q_2, while $q_1$is also the generator of q_0. q_3 has no generator and q_4 to q_n is the offspring of q_2 .

Furthermore, after considering the connection between questions, the next step is to determine if the answers detected for one question can be applied to another. To achieve this, an additive interpolation of the KL-divergence score and two other factors. The first factor is the distance between the questions is considered; it is observed that post far away from each other may not be related to the same topic in a forum. H Hence, a digraph between the two questions is generated. An evaluation of the distance between both questions, denoted by $d(q_0, q_1)$ is computed. The distance between the questions is obtained by the summation of all the thread posts between the questions $q_0 and q_1$. The second factor considered is the authorship of the question q_1, this is computed by considering the ratio of the author of q_1 with the max value of the other authors in the forum thread.

$$author(q_1) = \frac{(\#reply_{q_1})^2 / \#start_{q_1}}{max_{j\in I}\left((\#reply_j)^2 / \#start_j\right)} \qquad (10)$$

where $author\left(q_1\right)$ is the score for the author of q_1, $\#reply_{q_1}$ refers to the number of replies by the author of q_1 and $\#start_{q_1}$ is the number of forum thread started by the author of $q_1 . max_{j\in I}$ selects the maximum value among all the authors in the same forum thread as the author of q_1 with the highest value for the ratio of replies to post started. An increasing value of $author\left(q_1\right)$ indicate the degree of the authority and tend to ask more supporting and relevant questions to the previous questions.

3.6 Related Answers to a Question

Given two questions q_0 and q_1, the estimation of the edge $q_0 \to q_1$is achieved from the aggregation of three computations, which include the authority of the author of question q_1, the value of KL-divergence $KL(q_0| q_1)$, and the distance between q_0 and q_1.

$$w\left(q_0 \to q_1\right) = \frac{1}{1 + KL\left(q_0 \mid q_1\right)} + \frac{1}{d(q_0, q_1)} + author\left(q_1\right) (11)$$

where $w\left(q_0 \to q_1\right)$ is the weight of the edge between $q_0 and q_1$, if the weight is equal to or exceeds the threshold β, then it can be concluded that answers for a question can be applied to another question because of the connection between the questions. Eq. (11) shows the aggregated answers for question q_i. The answers for q_{i+1} is appended to the answer to question q_i if and only if there is a relationship between q_i and q_{i+1}.

$$answer\left(Sq_i\right) + = \begin{cases} if\left(w\left(q_i \rightarrow q_{i+1}\right)\right) \geq \beta answer\left(Sq_{i+1}\right) \\ elsenull \end{cases} (12)$$

To obtain the answers, $answer\left(Sq_i\right)$ represents the aggregated answers for the question q_i. This approach tends to provide more answers to a question and where there are no answers for a question, and there seems to be a relationship between questions within the same thread, the answers for a question can be applied to the related question.

3.7 Question without Answer

There may be cases where a question may not have answers, due to no of response, the difficulty of the question, the question is incorrect, irrelevant or such question does not require an answer. The approach in Eq. (13) provides a solution to the problem of discovering questions without answers. However, if there are no answers for a given question q_i, and there is no relationship between that question and other questions within the same forum thread, then it is considered to have no answers.

$$status\left(q_i\right) = \begin{cases} \sum [EA] > 0 Positive \\ elseNegative \end{cases} (13)$$

where $status\left(q_i\right)$ defines the status for question q_i, its value can either be positive or negative derived from the size of the answers for the question. A positive value means that the number of answers for q_i is greater than zero (0), which implies there are answers to the question. A negative or zero value indicates that there are no answers to the question.

3.8 Text Quality Information

Computed information features can describe the quality, status, and composition of a question's answers. These features, although they do not play any role in defining the validity of the answers, offer further insight and information about the answers used for assessment and rating of answers, especially when there is more than one answer to a question.

3.8.1 Visual Quality

The first factor to be considered is the visual quality of the text. It deals with the punctuation and typos in the answer extracted. The entropy of the answer is computed to determine how much information or understanding is in the answer. Entropy measures the degree of uncertainty; the higher this value, the lower the visual quality of the text (higher typo error). This approach was used in Ospanova (2013) for calculating the information entropy of language texts. The entropy approach depicts how well the next word of a text can be predicted when some of the proceeding text are known; it measures the probabilistic-linguistic relations in a given text. N-gram is the size of the text to be considered; the entropy of N-gram probabilistic method is considered for predicting the approximate occurrence value of a contiguous sequence of n words in a text (Jurafsky and Manning, 2015). In this paper, the maximum size of N-gram is four; the N-gram probabilistic method for a text is computed with Eq. (14) and Eq. (15) estimate the entropy of computed N-gram of a text.

$$D_N = \frac{\text{evaluatedfrequencyofN} - \text{gramintheanswertext}}{\text{totalN} - \text{gramsinthebloggroup}} (14)$$

$$H(X) = -\frac{1}{j}\sum_{i=1}^{j} D_N log_2\left(D_N\right) (15)$$

where $j =$ total N-gram in X, D_N is the probability occurrence of N-gram in X and $H(X)$ is the entropy value for N-gram of the text contained in $answer\left(Sq_i\right)$.

Table 4 shows the entropy percentage score; the table is constructed based on the minimum and maximum possible entropy value and it is converted to an equivalent percentage value, where 0.0 represents 100% and 1.0 represents 0% and the other values fall in between respectively. The purpose of the table is to assign a higher percentage value to lower entropy scores. Moreover, the cumulative score for the text's quality of an answer will be positively influenced by lower entropy value.

3.8.2 Syntax and Semantic Complexity

The next feature that is considered is the semantic and syntactic complexity of the text. The readability and comprehension of a text is an indication of the quality of the content. Therefore, the higher the text readability index the better the understanding, readership, memorization, and reading speed. However, lower readability index can create misunderstanding, disinterest, and even deception (Karmakar, 2011). In this paper, a readability metric defined in Karmakar (2011) was adopted that uses a single average number that classifies a via syntax complexity and semantic complexity. The syntax analysis of an answer is achieved by the average number of characters per word in the sentence, the number of syllables per word, and the vocabulary-based method. The number of vowels (a, e, i, o, u) heard in a word is the number of syllables in the word.. The size of the vowels is counted, and then subtract the number of silent vowels (e.g., the silent 'e' at the end of a word) and diphthongs (e.g., oi, oy, ou, ow, au, aw, oo, and so on). The vocabulary score is obtained by dividing the words in the answer not found in the WordNet by the totals number of words. The vocabulary score is 0 if all the words are found in the WordNet, which indicates familiar words. Two or more clauses and phrases, usually connected by conjunctions make up a complex sentence. Semantic complexity is measured by the number of conjunctions in the answer. A Stanford parser is used to analyze the sentence to assign annotation such as subject, the verb to the answer, and further parsing of the annotated text reveals the number of clauses and phrases.

$$SYN(X) = \frac{\sum Xc}{\sum Xw} + L(X) + V(X)\,(16)$$

$$SEM(X) = \sum f_p(X)\,(17)$$

$$R(X) = SYN(X).\,SEM(X)\,(18)$$

where $R(X)$, $SYN(X)$ and $SEM(X)$ is the readability, syntactic score and semantic score for sentence X respectively. $\sum Xc$ and $\sum Xw$ is the summation of the character and word in X respectively. $L(X)$ = total syllable in X, $V(X)$ is the vocabulary score for X and f_p is a function the generates a parse tree using Stanford parse for X and $\sum f_p$ counts the number of clauses in the parse tree.

Table 4
Entropy percentage score

Entropy value	Percentage Score
0.0	100.0%
0.1	90.0%
0.2	80.0%
0.3	70.0%
0.4	60.0%
0.5	50.0%
0.6	40.0%
0.7	30.0%
0.8	20.0%
0.9	10.0%
1.00	0.0%

3.8.3 Grammaticality

Another feature considered is the grammaticality of a sentence, which describes the linguistic competence of the sentence. Formal frequency measurement of this feature is used in Heylighen and Dewaele (2002), and part-of-speech (POS) tag is used for the sentence annotation. Two factors are considered for measuring the grammatical quality of the text, the deictic, D (words that need the context to understand the meaning) and non-deictic < ND (words that do not require the context to understand the meaning) in the expression of the text. The formality frequency increases with more deictic category words and decreases with the non-deictic.. Eqs. 19 and 20 reflect these respectively. The category of conjunctions has no a priori correlation within the context as shown in Eq, 21.

$$D_{freq} = Noun_{freq(X)} + Adj_{freq(X)} + Prep_{freq(X)} + Article_{feq(X)} \quad (19)$$

$$ND_{freq} = Pronoun_{freq(X)} - verb_{freq(X)} - Adverb_{freq(X)} - Interjection_{freq(X)} \quad (20)$$

$$F(X) = \frac{D_{freq} - ND_{freq} + 100}{2} \quad (21)$$

where $D_{freq}$and ND_{freq}refer to deictic and non-deictic categories respectively, $F(X)$ is the measure of grammaticality.

3.8.4 Readability Index

The last feature that is considered to describe the quality of answers is the readability index. The index estimates the smallest set of words required to understand a text. The idea behind readability index is that well written text will be understandable, well written, understandable and free from unnecessary complexity. Flesch reading ease is adopted to compute this feature for an answer (Dalip et al. 2009), which is shown in Table 5. The purpose of the score index is to provide a scale to rate written text for the different age groups. This feature takes into consideration the number of word per sentence and syllable per sentence, and a high score indicates that the answer is easier to read and a low score suggest that the answer is difficult to read. The readability index $Readability(X)$ is defined as:

$$Readability(X) = 206.835 - 1.015\left(\frac{totalwords(X)}{totalsentences(X)}\right) - 84.6\left(\frac{totalsyllables(X)}{totalwords(X)}\right) \quad (22)$$

Table 5
Readability Score index for a text from Dalip et al. 2009

Score	Note
90.0–100.0	Very easy to read
80.0–90.0	Easy to read in English conversation
70.0–80.0	Fairly easy to read
60.0–70.0	Plain English
50.0–60.0	Fairly difficult to read
30.0–50.0	Difficult to read
0.0–30.0	Very difficult to read, Best understood by a college graduate

The result for the answer to a question, $Fanswer\left(Sq_i\right)$, is a text quality score information that is assigned to the answer to give a numerical value and condition of the answers. The combination of all the text quality features is computed and appended to each answer. The purpose of text quality score is to rate answers where there is more than one answer to a question, the higher the score value the more appropriate the answer is to the question.

$$Fanswer\left(Sq_i\right) = H\left(answer\left(Sq_i\right)\right).R\left(answer\left(Sq_i\right)\right).F\left(answer\left(Sq_i\right)\right).Readability\left(answer\left(Sq_i\right)\right)\quad(23)$$

3.9 Utilised Dataset

This paper is a continuation of our previous published work on question extraction from the ResearchGate website (Ojokoh et al. 2016); therefore, we performed answer extraction for the extracted questions. The search for answers was performed from the same forum thread the questions was extracted. The website is a platform for profile management, discussion, find update on new development in different fields and collaboration among scientist (Lin, 2012; ResearchGate, 2015). Proliferation of the application among users across different countries, with of its users in Europe and America engaging in different discussion, which makes it suitable to for this research to identify QA pairs.

The content from the website was crawled with HTTRACK application, which copies the pages of a website on a local computer (Engebretson (2011); Beaver (2012)). Therefore, we extracted the required post in the discussion forum from the local copy of the webpages for analysis.

4. Result And Discussion

The discussion of the results and performance for the extraction of answers to questions are discussed in this section. Our results are represented in tables and graphs for comprehension and explanation. A sample of the patterns and keywords that are used for the extraction of answers from group1 is represented in Table 6. This system does not only consider the use of patterns or keywords for searching for answers but also applies the combination of the patterns and keywords to search for answers in the text obtained from web pages in the dataset. The blog group analyzed to extract questions is the same group examined for answers as well as other groups that share similarities with the question under consideration. The text group, questions, and similarity between questions was adopted from Ojokoh et al. (2016).

4.1 Performance of QBAE and PBAE Methods

The number of patterns and keywords generated is varied to observe the performance of the system. These parameters were obtained dynamically in the application module for the analysis presented in the methodology. The system implementation is configurable and adjustable to create the varying size of patterns and keywords for answer detection. The conversation text content that forms the blog group is processed to obtain answers. However, there could be some HTML tags within the text; the system inspects the text's content and uses an equivalent regular expression to detect and remove these tags before processing to search for answers.

Table 6
Patterns and keywords generated from group1

Extracted question	**Sample Pattern**	**Keywords for Query String**
I am considering using Java for some research works and I would like to know if someone has used ROS with Java	<java > < like > < ans>, <java > < ans>, <ans > < java>, <ans > < java > < like>, <java > < use > < ans>, <ans > < java > < use>, <java > < research > < ans>, <java > < ans > < research>, <work > < java > < ans > < research>	java,like,know,someone,would,used,works,using,some,research
Facebook posts, likes, and shares, nature of posts being investigated, do we have any software available for that	<facebook > < post > < ans>, <facebook > < likes > < ans>, <investigate > < facebook > < ans>, <investigate > < share > < ans>, <software > < facebook > < ans>	posts,have,software,available,investigated,nature,likes,shares,facebook,being
How does the SAX works, I have the code for the same in java but how to decide the sliding window size for a time series	<SAX > < ans>, <work > < ans>, <sax > < work > < ans>, <work > < sax > < ans>, <code > < ans > < sax>, <sliding > < ans > < SAX>, <window > < work > < ans>	window,SAX,sliding,size,time,series,decide,java,works,have,code
What circumstances I could apply the VFDT	<vfdt > < ans>, <ans > < vdt>, <vfdt > < apply > < ans>, <ans > < apply > < vfdt>, <could > < apply > < ans>, <apply > < ans>	vfdt,apply,could,circumstances

Table 7 shows the answer obtained for each question extracted in each group. The “Extracted question” column is the question extracted from the group and, the extracted answers display the output for the different answer extraction for the different methods in the subsequent columns. The answer extraction method with ‘N/A’ values refers to questions without answers.

Table 7
Answers extraction for questions extracted from group29

Extracted question	QBAE	PBAE	QBAE + PBAE
I would like to use a different way to draw maps	QGis	QGis	ArcGis
Does anyone know of a global Digital Terrain Model of the seabed	I confirm that GEBCO data base is the mostly used source of data for your purpose.	I confirm that GEBCO data base is the mostly used source of data for your purpose.	I would recomend the ETOPO
Can we elucidate structure by doing GCMS of fractions or purified fractions.	I think you can do GCMS for analyzing your extracts	I think you can do GCMS for analyzing your extracts	GCMS or LCMS and NMR study is required for complete structure elucidation
Is there any method to extract the data (means numerical values) from the graphical representation	The response is... YES	There is software "dcsDigitiser Graph Digitizer" that I have developed worthy of trying	I've found Plot Digitizer to be the friendliest software for converting graphs into data
How can I deal with missing data in MATLAB	If I am getting it proper, the missing number can be found by interpolation.	Why don't you use the loop and find NaN and replace that with 0 within the loop.	Why don't you use the loop and find NaN and replace that with 0 within the loop.
Where can I find (estimated) monthly irradiation data	you need to get weather data set eith in TMY or EPW format	Data can be viewed free and shall be copied from site	Data can be viewed free and shall be copied from site

Tables 8, 9, and 10 show the text quality information for QBAE, PBAE and QBAE + PBAE extracted answer technique for the same question obtained from Table 7. There are seven columns in the table, the first column is for the question under consideration and the second column is for the answers returned for the applied method. The “visual quality”, “syntax and semantic complexity”, “grammaticality”, “Readability Index” columns hold the score value for text quality information for the answers to the question. The “average” column represents the average score of all the text quality information. This score is necessary if there is more than one answer assign to the question.

Tables 11 to 15 show the performance for the answers extracted by different methods, considering the varying percentage of patterns and keywords that were generated. The “question category” field holds the different question category that was considered from Ojokoh et al. (2016). “Question Extracted” field refers to the number of questions available for that category. The answer extraction method fields “QBAE”, “PBAE” and “QBAE + PBAE” are further divided into two columns, which are “Correctly Answered” and “Wrongly Answered” which represents answers that correctly address the question for that category. A Question is said to be correctly answered if the answer type in Table 1 assigned to a question category in Table 2 is found in the answer text. Otherwise, the question is said to be wrongly answered. The number of questions without answers are recorded in the “No Answer” column. The results show that the combination of both methods (PBAE + QBAE) performs better when all the patterns and keywords generated are used, the same performance is achieved when 90% of the patterns and keywords were applied. To measure the system's performance, the total number of patterns and queries generated for answer identification and extraction varies based on percentage. The lowest limit of the pattern and query strings used for the processing is set at 60% to achieve meaningful results. The result shows that at 90%, all the required keywords and patterns are adequate to detect and extract answers to a question.

Table 8
Answers provided by QBAE with text quality information

Question	Answer	Visual Quality	Syntax and Semantic Complexity	Grammaticality	Readability Index	Average
How can I deal with missing data in MATLAB	If I am getting it proper, the missing number can be found by interpolation	60	70	75	85	72.5
	the missing number can be found by interpolation	50	60	50	70	57.5
	If I am getting it proper	50	60	40	65	53.75

Table 9
Answers provided by QB with text quality information

Question	Answer	Visual Quality	Syntax and Semantic Complexity	Grammaticality	Readability Index	Average
How can I deal with missing data in MATLAB	Why don't you use the loop and find NaN and replace that with 0 within the loop	80	80	85	85	82.5
	Why don't you use the loop and find NaN	60	70	60	80	67.5
	Why don't you use the loop	60	65	50	65	60.0

Table 10
Answers provided by QB + PB with text quality information

Question	Answer	Visual Quality	Syntax and Semantic Complexity	Grammaticality	Readability Index	Average
How can I deal with missing data in MATLAB	Why don't you use the loop and find NaN and replace that with 0 within the loop	80	80	85	85	82.5
	Why don't you use the loop and find NaN	60	70	60	80	67.5
	Why don't you use the loop	60	65	50	65	60.0

Table 11
Answers provided by different methods at 60% generated patterns and keywords

Question Category	Question Extracted	QB method		PB method		QB + PB method		No Answer
		Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	
Any	15	5	5	3	7	5	5	5
Ask	23	10	9	10	9	10	9	4
Can	80	70	9	70	9	70	9	
Could	30	20	4	20	4	20	4	6
Do	42	32	10	32	10	32	10	
Excuse me	3	0	1	0	1	0	1	2
Have	326	304	10	304	10	304	10	12
How	352	320	12	320	12	320	12	20
Is	6	2	4	0	6	2	4	
Let	5	1	3	0	4	1	3	1
May	48	32	9	32	9	32	9	2
Might	22	13	6	13	6	13	6	4
Not	3	0	3	0	3	0	3	
Perhaps	6	0	6	0	6	0	6	
Please	18	5	10	5	10	5	10	3
Suppose	5	0	5	0	5	0	5	
Were	62	45	10	45	10	45	10	7
What	362	336	3	336	3	336	3	23
When	324	304	2	304	2	304	2	18
Where	295	271		271		271		24
Who	361	322	11	322	11	322	11	33
Why	380	348	8	346	10	346	10	34
Would	247	212	9	211	10	212	9	31

Table 12
Answers provided by different methods at 70% generated patterns and keywords

Question Category	Question Extracted	QB method		PB method		QB + PB method		No Answer
		Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	
Any	15	8	2	8	2	10		5
Ask	23	17	2	15	4	19		4
Can	80	77	3	77	3	70	9	
Could	30	23	1	23	1	21	3	6
Do	42	39	3	36	6	42		
Excuse me	3	0	1	0	1	0	1	2
Have	326	312	2	309	5	314		12
How	352	330	2	328	4	332		20
Is	6	1	5	1	5	6		
Let	5	0	4	0	4	4		1
May	48	45	2	42	5	35	6	2
Might	22	15	3	12	6	11	7	4
Not	3	3		3		3		
Perhaps	6	6		6		6		
Please	18	15		15		15		3
Suppose	5	0	5	0	5	2	3	
Were	62	55		55		55		7
What	362	334	5	336	3	336	3	23
When	324	302	4	302	4	304	2	18
Where	295	268	3	267	4	271		24
Who	361	326	2	323	5	328	5	33
Why	380	344	2	341	5	342	4	34
Would	247	211	4	211	4	216	5	31

Table 16 presents the performance for the answers extracted for each answer type for QBAE method, PBAE method and the combination of both methods at different percentage value. PBAE and QBAE methods have similar result except for the "purpose" answer type. QBAE + PBAE answer type have a better result as more answer was obtained from "purpose" and "description" answer type. The performance of the system increased as the percentage of patterns and keywords considered increased, this proves that the more the patterns and keywords considered the better the performance. Also, the performance is better when both methods are combined.

Table 13
Answers provided by different methods at 80% generated patterns and keywords

Question Category	Question Extracted	QB method		PB method		QB + PB method		No Answer
		Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	
Any	15	10		9	1	10		5
Ask	23	19		19		19		4
Can	80	76	4	77	3	79	1	
Could	30	24		23	1	24		6
Do	42	42		42		42		
Excuse me	3	1		1		1		2
Have	326	314		314		314		12
How	352	332		332		332		20
Is	6	6		6		6		
Let	5	3	1	3	1	4		1
May	48	38	8	38	8	44	2	2
Might	22	17	1	17	1	17	1	4
Not	3	3		3		3		
Perhaps	6	6		6		6		
Please	18	15		15		15		3
Suppose	5	5		5		5		
Were	62	55		55		55		7
What	362	336	3	336	3	339		23
When	324	306		306		306		18
Where	295	271		271		271		24
Who	361	327	1	327	1	328		33
Why	380	346		346		346		34
Would	247	213	3	215	1	216		31

Figures 4 and 5 show a summary of the result displayed in Table 16. The result from the figures shows the overall performance of the different answer extraction methods. QBAE and PBAE methods share close similarities concerning the number of answers that were correctly given to questions. The combination of both methods shows a better performance with 4 wrong answers at 80% and no wrong answer at 90% and 100%. The table also shows that there is a total of 229 questions without answers. At 60% QBAE method performed slightly better than other methods with 2637 questions correctly extracted over the PBAE method and QBAE + PBAE method with 2639 and 2635 correctly answered questions respectively. At 70% and above, PBAE + QBAE outperformed other methods with a decline in the number of wrong answers to questions outcome. The reason for this could be the availability of enough keywords to detect more answers above other methods. The overall optimal performance of the QBAE + PBAE method over the particular method is due to the combined score value for answers to a question which improves the index and ranking for the answers.

Table 14
Answers provided by different methods at 90% generated patterns and keywords

Question Category	Question Extracted	QB method		PB method		QB + PB method		No Answer
		Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	
Any	15	10		10		10		5
Ask	23	19		19		19		4
Can	80	79	1	79	1	80		
Could	30	24		24		24		6
Do	42	42		42		42		
Excuse me	3	1		1		1		2
Have	326	314		314		314		12
How	352	332		332		332		20
Is	6	6		6		6		
Let	5	3	1	4		4		1
May	48	45	1	45	1	46		2
Might	22	17	1	18		18		4
Not	3	3		3		3		
Perhaps	6	6		6		6		
Please	18	15		15		15		3
Suppose	5	5		5		5		
Were	62	55		55		55		7
What	362	339		339		339		23
When	324	306		306		306		18
Where	295	271		271		271		24
Who	361	328		328		328		33
Why	380	346		346		346		34
Would	247	216		216		216		31

Table 15 Answers provided by different methods at 100% generated patterns and keywords

Question Category	Question Extracted	QB method		PB method		QB + PB method		No Answer
		Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	Correctly Answered	Wrongly Answered	
Any	15	10		10		10		5
Ask	23	19		19		19		4
Can	80	79	1	79	1	80		
Could	30	24		24		24		6
Do	42	42		42		42		
Excuse me	3	1		1		1		2
Have	326	314		314		314		12
How	352	332		332		332		20
Is	6	6		6		6		
Let	5	3	1	4		4		1
May	48	45	1	45	1	46		2
Might	22	17	1	18		18		4
Not	3	3		3		3		
Perhaps	6	6		6		6		
Please	18	15		15		15		3
Suppose	5	5		5		5		
Were	62	55		55		55		7
What	362	339		339		339		23
When	324	306		306		306		18
Where	295	271		271		271		24
Who	361	328		328		328		33
Why	380	346		346		346		34
Would	247	216		216		216		31

4.2 System Evaluation

The results of the answer extraction for questions were evaluated with the following parameters precision, recall, accuracy, and F-measure. These parameters are defined as:

Precision= $\frac{A}{A+C}$

Recall = $\frac{A}{A+B}$

Accuracy = $\frac{A+D}{A+B+C+D}$

F-measure = $\frac{2 \times precision \times Recall}{precision + Recall}$

A = Total correct answers to questions of a category

B = Total answers found (existing) but not classified as an answer for a category

C = Total wrong answers to questions of a category

D = Total answers that are not found (not existing) but not classified as an answer for a category

Tables 17, 18, and 19 show the record for evaluating the system's performance using precision, accuracy, and f-measure. The result from the tables shows that the average value in Table 19, which represents the result of the QBAE + PBAE. It method provides the best result with an average value of 1.000 for accuracy and f-measure.

Table 16
Questions Assigned to Answer Type for different percentages

Answer type	QBAE				PBAE				QBAE + PBAE			
	60%	70%	80%	90%	60%	70%	80%	90%	60%	70%	80%	90%
Person	172	184	186	188	172	180	188	188	172	186	188	188
Organisation	141	152	152	156	143	152	152	156	141	152	156	156
Length	122	122	122	125	122	122	122	125	122	122	122	122
Nominal												
Place	372	394	394	394	372	394	394	394	372	394	394	394
Description	528	549	555	555	528	542	555	555	528	549	557	557
Money	16	17	17	17	16	17	17	17	16	17	17	17
Others	341	351	351	358	341	349	350	358	341	351	359	361
Date	4	4	4	4	4	4	4	4	4	4	4	4
Abbreviation												
Reason	428	428	432	432	428	428	432	432	428	432	432	432
Number	38	38	40	40	38	38	40	40	38	38	40	40
Known for	1	1	1	1	1	1	1	1	1	1	1	1
Duration	42	42	45	45	42	42	45	45	42	45	45	45
Definition	23	38	47	48	23	30	47	50	23	36	48	48
Rate	3	5	5	5	3	5	5	5	3	5	5	5
Purpose	406	406	414	414	406	406	414	414	404	406	414	416

Table 20 shows the summary of the result for the evaluation of the performance of different models at 90%. The result shows that the average accuracy for all methods is 1.000, which depicts the system's correctness. The average F-measure score shows that QBAE + PBAE has an excellent result, and the PBAE method has a better average performance above QBAE.

Table 17
QBAE method answer evaluation at 90% generated patterns and keywords

Groups	QBAE method			
	Precision	Accuracy	Recall	F-measure
Any	1.000	1.000	1.000	1.000
Ask	1.000	1.000	1.000	1.000
Can	0.988	1.000	0.988	0.994
Could	1.000	1.000	1.000	1.000
Do	1.000	1.000	1.000	1.000
Excuse me	1.000	1.000	1.000	1.000
Have	1.000	1.000	1.000	1.000
How	1.000	1.000	1.000	1.000
Is	1.000	1.000	1.000	1.000
Let	0.750	1.000	0.800	0.857
May	0.978	1.000	0.979	0.989
Might	0.944	1.000	0.955	0.971
Not	1.000	1.000	1.000	1.000
Perhaps	1.000	1.000	1.000	1.000
Please	1.000	1.000	1.000	1.000
Suppose	1.000	1.000	1.000	1.000
Were	1.000	1.000	1.000	1.000
What	1.000	1.000	1.000	1.000
When	1.000	1.000	1.000	1.000
Where	1.000	1.000	1.000	1.000
Who	1.000	1.000	1.000	1.000
Why	1.000	1.000	1.000	1.000
would	1.000	1.000	1.000	1.000
Average	0.985	1.000	0.988	0.992

4.3 Comparative Analysis with Previous Models

Table 22 shows the comparative analysis of our proposed system result against other models described in Hong and Davison (2009). The abbreviation description of the methods is given in Table 21. In our proposed model, precision, accuracy, recall and f-measure are 1.000 each respectively, against other features compared. The performance of POSI + AUTH performed well with the values of precision, accuracy, recall and f-measure, which are 0.958, 0.993, 0.975, and 0.975 respectively. However, based on the results, our proposed method outperforms every other model in answer detection and extraction in CQAs.

Table 18
PBAE method answer evaluation at 90% generated patterns and keywords

Groups	PBAE method			
	Precision	Accuracy	Recall	F-measure
Any	1.000	1.000	1.000	1.000
Ask	1.000	1.000	1.000	1.000
Can	0.988	1.000	0.988	0.994
Could	1.000	1.000	1.000	1.000
Do	1.000	1.000	1.000	1.000
Excuse me	1.000	1.000	1.000	1.000
Have	1.000	1.000	1.000	1.000
How	1.000	1.000	1.000	1.000
Is	1.000	1.000	1.000	1.000
Let	1.000	1.000	1.000	1.000
May	0.978	1.000	0.979	0.989
Might	1.000	1.000	1.000	1.000
Not	1.000	1.000	1.000	1.000
Perhaps	1.000	1.000	1.000	1.000
Please	1.000	1.000	1.000	1.000
Suppose	1.000	1.000	1.000	1.000
Were	1.000	1.000	1.000	1.000
What	1.000	1.000	1.000	1.000
When	1.000	1.000	1.000	1.000
Where	1.000	1.000	1.000	1.000
Who	1.000	1.000	1.000	1.000
Why	1.000	1.000	1.000	1.000
Would	1.000	1.000	1.000	1.000
Average	0.999	1.000	0.999	0.999

Table 19
QBAE + PBAE method answer evaluation at 90% generated patterns and keywords

Groups	QBAE + PBAE method			
	Precision	Accuracy	Recall	F-measure
Any	1.000	1.000	1.000	1.000
Ask	1.000	1.000	1.000	1.000
Can	1.000	1.000	1.000	1.000
Could	1.000	1.000	1.000	1.000
Do	1.000	1.000	1.000	1.000
Excuse me	1.000	1.000	1.000	1.000
Have	1.000	1.000	1.000	1.000
How	1.000	1.000	1.000	1.000
Is	1.000	1.000	1.000	1.000
Let	1.000	1.000	1.000	1.000
May	1.000	1.000	1.000	1.000
Might	1.000	1.000	1.000	1.000
Not	1.000	1.000	1.000	1.000
Perhaps	1.000	1.000	1.000	1.000
Please	1.000	1.000	1.000	1.000
Suppose	1.000	1.000	1.000	1.000
Were	1.000	1.000	1.000	1.000
What	1.000	1.000	1.000	1.000
When	1.000	1.000	1.000	1.000
Where	1.000	1.000	1.000	1.000
Who	1.000	1.000	1.000	1.000
Why	1.000	1.000	1.000	1.000
Would	1.000	1.000	1.000	1.000
Average	1.000	1.000	1.000	1.000

Table 20
Summary for answer evaluation method for 90% Generated Patterns and keywords

Answer extraction Methods	**Average value**			
	Precision	Accuracy	Recall	F-measure
QBAE	0.985	1.000	0.988	0.992
PBAE	0.999	1.000	0.999	0.999
QBAE + PBAE	1.000	1.000	1.000	1.000

Table 21
The features and their abbreviations adapted from Hong and Davison (2009)

Features	**Abbreviations**
Authorship	AUTH
Graph-Based	GB
Graph + Query Likelihood	GQL
Query Likelihood Model	LM
N-grams	NG
Pattern-Based	PB
Position	POSI
Stop Words	SW

Table 22
Proposed system against other methods compared in Hong and Davison (2009)

Method	**Precision**	**Accuracy**	**Recall**	**F-measure**
LM + GQL	0.735	0.688	0.594	0.657
LM + AUTH	0.700	0.719	0.771	0.734
SW + NG	0.737	0.720	0.688	0.712
LM + SW	0.765	0.747	0.717	0.740
LM + POSI	0.780	0.815	0.879	0.827
LM + POSI + SW	0.846	0.867	0.899	0.872
POSI + SW	0.846	0.868	0.901	0.873
LM + POSI + AUTH	0.951	0.970	0.991	0.970
POSI + AUTH	0.958	0.975	0.993	0.975
QBAE + PBAE	1.000	1.000	1.000	1.000

5. Conclusion

The multiple ways and patterns of asking question create a challenge for retrieving answers; it becomes important to create an automatic system capable of handling these challenges. In our previous work, we have extracted questions from an online web forum. In this paper, we extract answers to questions that we have identified. Our model is a hybrid method for answer extraction based on pattern learning and query-based technique. The experimental result shows that using a hybrid algorithm for answer extraction will improve the QA system's performance. We presented a combined model for detecting answers to questions, and the model further identifies questions without answers, where there is no response available to questions. A question query string was constructed that was used for detecting possible answers from a web forum. The query string consists of keywords drawn from the question. Keywords in a question refer to significant words in the question that defines the question. A question-driven approach was considered for dynamically generating patterns for detecting answers for different types of question categories. The pattern used for finding answers to a question is dependent on the text or words found in the question. A combined pattern and query string were applied for answer extraction. Related answers to questions were considered based on question similarity, and where there is no answer to a question, such question is labeled as having no answers. Where there is more than one answer to a question, the rating of answers is done based on the text quality features of the answers of the question. The dataset for this research was crawled from ResearchGate with Httrack web copier.. Answers were identified for questions that were extracted from the crawled online forum. The result shows that the proposed model can identify and extract answers from CQA. The evaluation of the system and the performance shows that the system identifies and extracts answers to questions obtained in the forum. However, this result is obtained from considering only the dataset obtained from ResearchGate. The further direction in this research will be to consider how this model will perform for another type of dataset generated from other blogs and social media websites. Another area to further consider in this research is to improve the efficiency in finding answers to a question. It is essential to improve the model so that it returns results faster with minimal computing resources. In this work, answers were obtained from the same dataset and blog group where the questions were extracted, most times similar or the same question are discussed on a different platform. In future research, parallel answer extraction from multiple datasets for comparison to eliminate questions without answers, where a particular dataset does not have the answers to the question.

Declarations

Funding The research was partly funded by the COMSTEC-TWAS Joint Research Grants Programme 14-014 RG/TC/ITC/AF/AC_C- UNESCO FR: 3240283404

Conflicts of interest/Competing interests The authors do not have any conflicts of interest in this submission

Availability of data and material Not applicable

References

1. Ahmed, W., & Anto, D. B. (2016). Answer Extraction Technique for Question Answering Systems. International Journal of Innovative Research in Computer and Communication Engineering (An ISO 3297: 2007 Certified Organization), 4(11)
2. Albaham, A. T., & Salim, N. (2012). Quality-biased retrieval in online forums. Faculty of Computer Science and Information System, UTM, Johor, Malaysia-81310 Journal of Theoretical and Applied Information Technology 15th, 38 (pp. 1992–8645).
3. Allahbakhsh, M., Benatallah, B., Ignjatovic, A., Motahari-Nezhad, H. R., Bertino, E., & Dustdar, S. (2013). Quality control in crowdsourcing systems: Issues and directions, Internet Computing, IEEE 17 (pp. 76–81).
4. Beaver, K. (2012). Hacking for Dummies. John Wiley & Sons, ISBN 9781118380963 (pp. 278,280–281).
5. Bouguessa, M., Dumoulin, B., & Wang, S. (2008). Identifying authoritative actors in question-answering forums: The case of Yahoo! answers. In Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and

Data Mining, New York, NY(pp. 866–874).

6. Cong, G., Wang, L., Lin, C. Y., Song, Y. I., & Sun, Y. (2008). Finding question-answer pairs from online forums. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY (pp. 467–474).
7. Cooper, R. J. (2000). High precision information retrieval. M.Sc thesis, Imperial College of Science, Technology and Medicine, 180 Queen's way London, England.
8. Cooper, R. J., & Ruger, S. M. (2001). A simple question answering system. M.Sc work at the Department of Computing, Imperial College of Science, Technology and Medicine, 180 Queen's way London, SW7 2BZ England.
9. Dali, L., Rusu, D., Fortuna, B., Mladenić, D., & Grobelnik, M. (2009). Question answering based on semantic graphs. Journal of Jožef Stefan Institute Jamova cesta 39 1000 Ljubljana, Slovenia.
10. Dalip, D. H., Goncalves, M. A., Cristo, M., & Calado, P. (2009). Automatic quality assessment of content created collaboratively by web communities: a case study of Wikipedia. In Proceedings of the Joint International Conference on Digital libraries. Austin, TX, USA, (pp. 295–304).
11. Dan, S., Geert-Jan M. K., & Dietrich, K. (2005). Exploring syntactic relation patterns for question answering. In Robert Dale, Kam-Fai Wong, Jian Su, & Oi Yee Kwong, editors, Natural Language Processing IJCNLP 2005: Second International Joint Conference, Jeju Island, Korea, October 11–13, Proceedings. Springer-Verlag.
12. Davis, C. I., & Moldovan, D. (2010). Feasibility of automatically bootstrapping a persian WordNet. Proceeding of the 7th Language Resources and Evaluation Conference, (LREC) 2010, May 17–23, Valletta, Malta.
13. Engebretson, P. (2011). The basics of hacking and penetration testing. Elsevier. ISBN 9781597496568 (pp. 19–22).
14. Fox, C. (1992). Information retrieval data structures and algorithms. Lexical Analysis and Stop lists, (pp. 102–130).
15. G´omez-Adorno, H., Sidorov, G., Vilarino, D., & Pinto, D. (2015). Graph-Based Approach for Answer Selection in Community Question Answering Task. In the Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 18–22, Denver, Colorado, June 4–5, 2015. Association for Computational Linguistics.
16. Green, B., Wolf, A., Chomsky, C., Laughery, K. (1961). BASEBALL: An automatic question answerer. In: proceedings of the Western Joint Computer Conference.
17. Gupta, P. & Gupta, V. (2012). A survey of text question answering techniques. International Journal of Computer Applications, 53 (pp. 1–8).
18. Han, J., Yu, J., Jin, X. & Luo, J. (2011). Collection-based sparse label propagation and its application on social group suggestion from photos. ACM Trans. Intell. Syst. Technol. 2, 12 (pp. 21).
19. Heylighen, F., & Dewaele, J. M. (2002). Variation in the contextuality of language: An empirical measure. Context in Context. Special issue Foundations of Science, 7 (pp. 293–340).
20. Hong, L., & Davison, B. D. (2009). A classification-based approach to question answering in discussion boards. Proceeding of the SIGIR'09, Boston, Massachusetts, USA., CM 978-1-60558-483-6/09/0
21. Hong, R., Wang, M., Li, G., Nie, L., Zha, V, X Chua, Z.-J. (2012). Multimedia question answering. IEEE Multimedia 19 (pp. 72–78).
22. Jurafsky, D., & Manning, C. (2015). Natural Language Processing. Instructor, 212(998), 3482.
23. Jurczyk, P., & Agichtein, E. (2007). Discovering authorities in question answer communities by using link analysis. In Proceedings of the sixteenth ACM Conference on Information and Knowledge Management (CIKM), New York (pp. 919–922).
24. Kamp, H. (1984). A theory of truth and semantic representation. In: J. Groenendijk, T. M. Janssen, M. Stokhof (Eds.), Truth, Interpretation and Information: Selected Papers from the 3rd Amsterdam Colloquium, Dordrecht – Holland/ Cinnaminson – USA Foris, (pp. 1–41).
25. Karmakar, S. (2011). Syntactic and semantic analysis and visualization of unstructured English texts. Scholar work at Georgia University, Computer Science dissertations.

26. Kaur, J., & Gupta, V. (2010). Effective approaches for extraction of keywords. International Journal of Computer Science Issues (IJCSI), 7(6).
27. Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. Journal of the ACM 46 (pp. 604–632).
28. Lafferty, J., & Zhai, C. (2001). A study of smoothing methods for language models applied to Ad Hoc information retrieval. In proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval. 2001, ACM: New Orleans, Louisiana, United States (pp. 334–342).
29. Lin, T., (2012). Cracking open the scientific process. The New York Times. Retrieved 2014-06-26.
30. Liu, Q., & Agichtein, E. (2011). Modelling answerer behaviour in collaborative question answering systems. In: Proceedings of ECIR.
31. Liu, X., Croft, W. B., & Koll, M. (2005). Finding experts in community-based question-answering services. In ACM Conference on Information and Knowledge Management (CIKM).
32. Liu, Y., Li S., Cao Y., Lin, C.Y., Han D., & Yu. Y. (2009). Understanding and summarizing answers in community-based question answering services. In Proc. of COLING 2009.
33. Martinez-Romo, J., & Araujo, L. (2009). Web spam identification through language model analysis. Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web (pp. 21–28).
34. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Five papers on wordnet. International Journal of Lexicography.
35. Molla, D. (2006). Learning of graph-based question answering rules. Department of Computing Macquarie University Sydney 2109, Australia.
36. Mudgal, R., Madaan, R., Sharma, A. K., & Dixit, A. (2013). A Novel architecture for question classification based indexing scheme for efficient question answering. arXiv preprint arXiv:1307.6937.
37. Nakov, P., Marquez, L., Magdy, W., Moschitti, A., Glass, J. & Randeree, B. (2015). Answer selection in community question answering. In Proceedings of the 9th International Workshop on Semantic Evaluation, Denver, Colorado, June. Association for Computational Linguistics (pp. 269–281).
38. Ojokoh, B., Igbe, T., Araoye, A., &Ameh, F. (2016). Question identification and classification on an academic question answering site. In Digital Libraries (JCDL), 2016 IEEE/ACM Joint Conference on (pp. 223–224).
39. Ospanova, B. R. (2013). Calculating information entropy of language texts. World Applied Sciences Journal, ISSN 1818–4952 (pp. 41–45).
40. Peñas, A., Forner, P., Rodrigo, A., Sutcliffe, R., Forascu, C., & Mota, C. (2010). Overview of ResPubliQA 2010: Question Answering Evaluation over European Legislation. Working Notes CLEF Labs.
41. Radev, D., Fan, W., Qi, H., Wu, H., & Grewal, A. (2005). Probabilistic question answering on the Web. Journal of the American Society for Information Science and Technology (pp. 571–583).
42. Ramprasath, M., & Hariharan, S. (2016). A survey on question answering system. International Journal of Research and Reviews in Information Sciences (IJRRIS), United Kingdom (pg. 171–179).
43. Salloum, W. (2015). A Question Answering System based on Conceptual Graph Formalism. In the Faculty of Applied Sciences, University of Kalamoon, UOK, Damascus, Syria.
44. Toba, H., Ming, Z. Y., Adriani, M., & Chua, T. (2013). Discovering high quality answers in community question answering archives using a hierarchy of classifiers. Information Sciences journal homepage: Contents lists available at Science Direct.
45. Tomokiyo, T., & Hurst, M. (2003). A language model approach to keyphrase extraction. MWE '03 Proceedings of the ACL 2003 workshop on Multiword expressions: analysis, acquisition and treatment (pp. 33–40).
46. Voorhees, E. M. (2001). The TREC question answering track, Natural Language Engineering 7(4): 361–378.

47. Voorhees, E. M. (2004). Overview of the TREC 2004 question answering track. In Proceedings of the Thirteenth Text Retrieval Conference.
48. Wang, X. J., Tu, X., Feng, D., & Zhang, L. (2009). Ranking community answers by modeling question–answer relationships via analogical reasoning. In Proceedings of SIGIR.
49. Yen, S., Wu, Y., Yang, J., Lee, Y., Lee, C., Liu, J. (2013). A support vector machine-based context-ranking model for question answering. Information Sciences, Science Direct, pp. 77–87.
50. Zoidi, O., Fotiadou, E., Nikolaidis, N., & Pitas, I. (2015). Graph-based label propagation in digital media: A Review ACM Comput. Surv. V, N, Article A (January), 35.

Figures

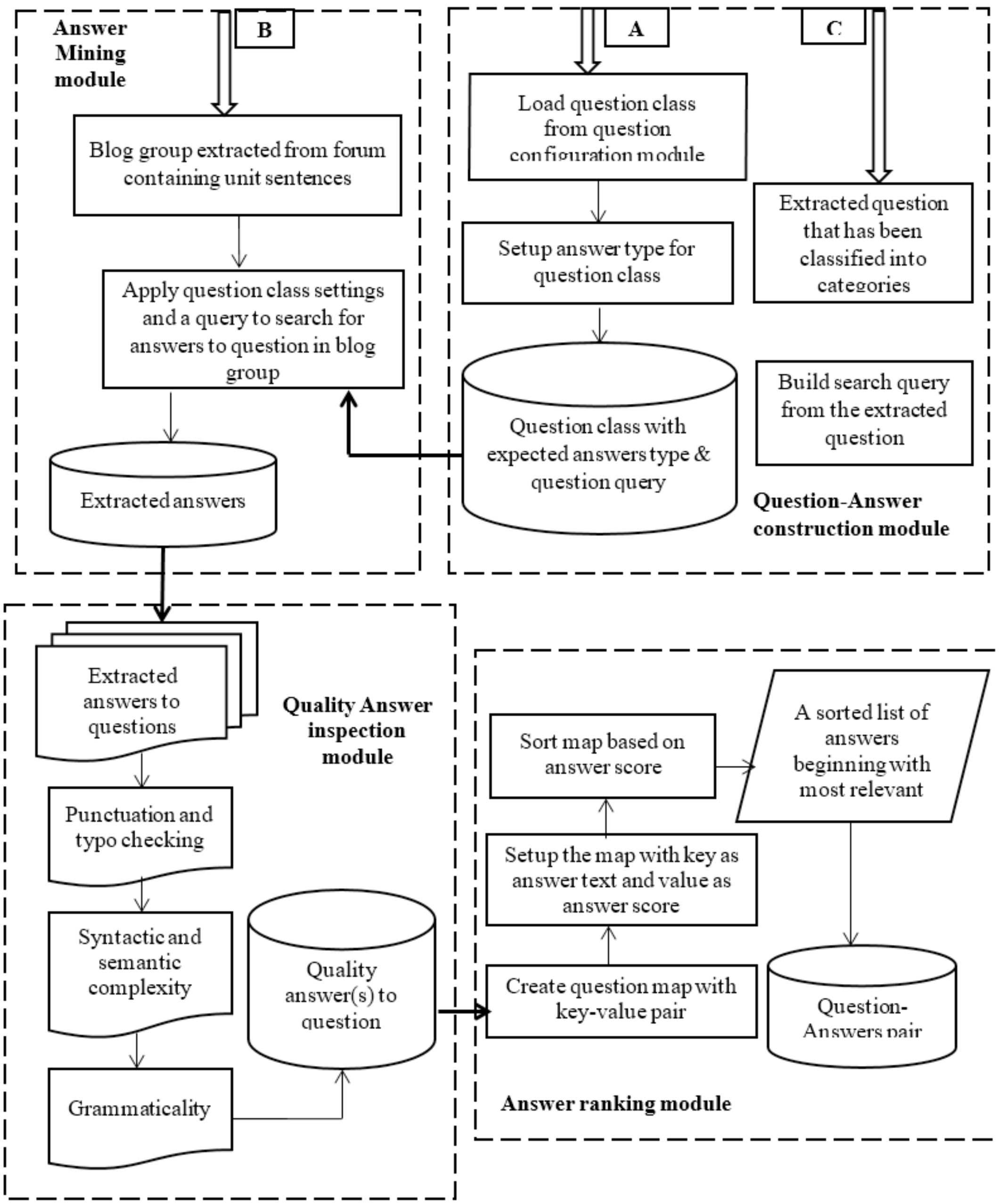

Figure 1

Quality answer extraction architecture.

A: Extensible Question Configuration Module; B: Pre-processing Module; C: Question Classification Module

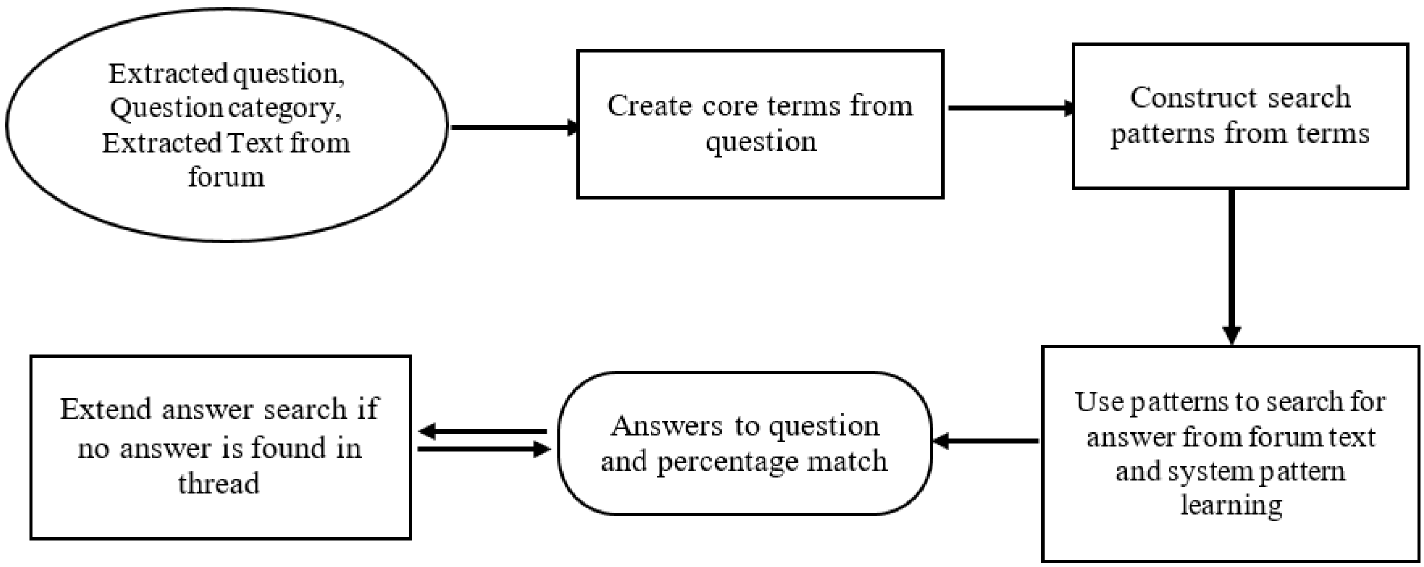

Figure 2

Answer detection using constructed pattern

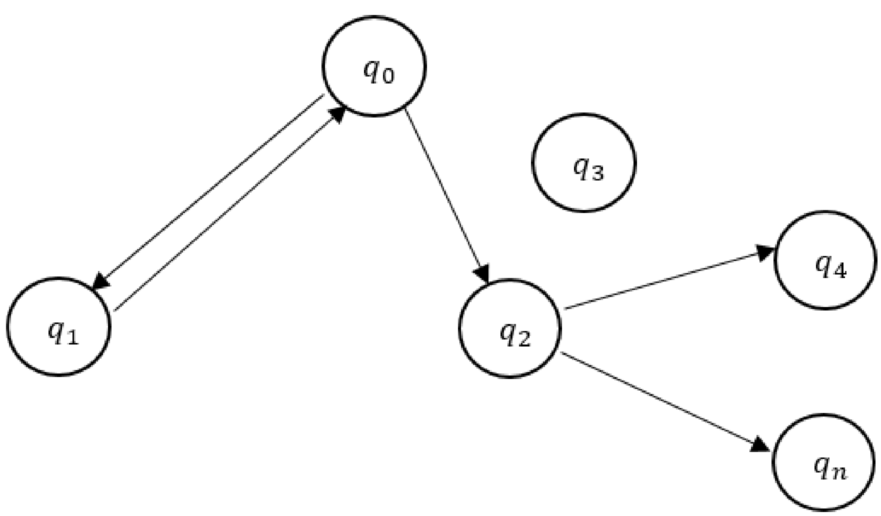

Figure 3

Graphical relationship between question for answer extraction

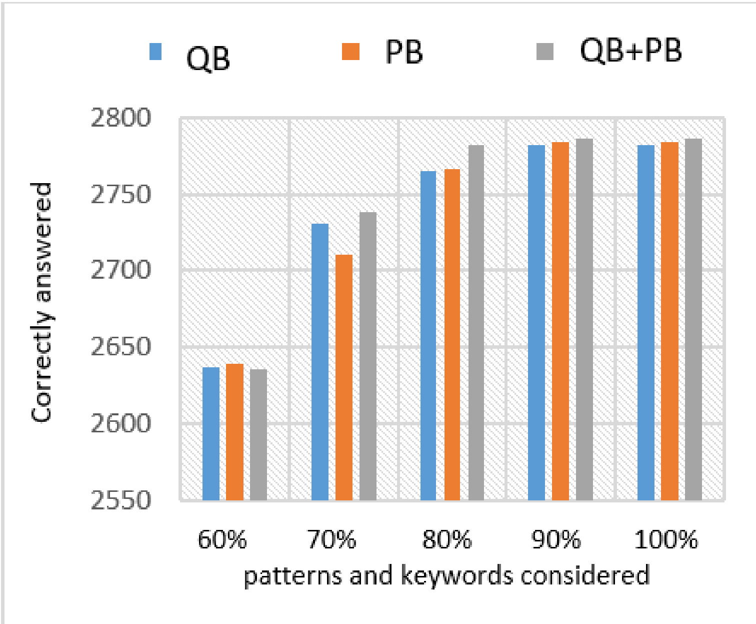

Figure 4

Summary of correctly answered questions

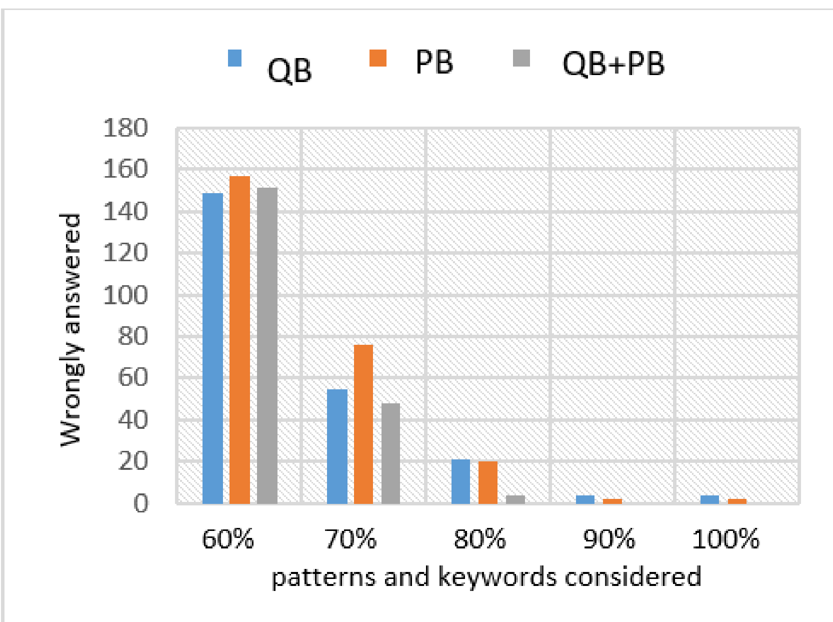

Figure 5

Summary of wrongly answered questions