British Journal of Mathematics & Computer Science
22(1): 1-21, 2017; Article no.BJMCS.32541
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Feature-based Model for Extraction and Classification of High Quality Questions in Online Forum

Bolanle Ojokoh[1*], Tobore Igbe[1] and Ayobami Araoye[1]

[1] *Department of Computer Science, Federal University of Technology, P.M.B. 704, Akure, Nigeria.*

Authors' contributions

This work was carried out in collaboration between all authors. Author BO initiated and designed the study. Author TI managed the implementation and experimental study and wrote the first draft of the manuscript. Author AA managed the literature searches and part of the experimental study. All the authors read and approved the final manuscript.

Article Information

DOI: 10.9734/BJMCS/2017/32541
Editor(s):
(1) Dariusz Jacek Jakóbczak, Chair of Computer Science and Management in this department, Technical University of Koszalin, Poland.
Reviewers:
(1) Anuj Goel, Maharishi Markandeshwar University, India.
(2) S. K. Srivatsa, Anna University, Chennai, India.
(3) H. Khalil, University of Poonch Rawlakot, Pakistan.
(4) S. Sridhar, R V College of Engineering, Bangalore, India.
(5) Su Su Htay, University of Information Technology, Yangon, Myanmar.
Complete Peer review History: http://www.sciencedomain.org/review-history/18868

Original Research Article

Received: 1st March 2017
Accepted: 6th April 2017
Published: 1st May 2017

Abstract

Aims: To design and implement a classification-based model using specific features for identification and extraction of high quality questions in a thread.
Study Design: The study design is divided into three modules: preprocessing, configuration, and question classification
Place and Duration of Study: Department of Computer Science of the Federal University of Technology Akure, between June 2016 and December 2016
Methodology: This research proposes a way of identifying, extracting and classifying questions in order to enhance high quality answers in an online forum. One of the major issues in question extraction and classification in forum is the restriction on the number of categories considered such as Who, What, Where, Where, Which, Why and How which are not sufficient to capture all possible questions. In this work, a number of parameters were proposed and aggregated using fuzzy logic for context based spam detection and removal in order to enhance question identification and classification. Part of speech (POS) tagging was applied to analyse the structure of each extracted sentence based on the presence and position

*Corresponding author: E-mail: bolanleojokoh@yahoo.com;

of predefined question tags; with this, issues like case sensitivity, grammatical construction and synonyms are addressed. Question classification is carried out with Naïve Bayes and identifying semantic relationship between extracted questions is achieved with cosine similarity model. Experiments were performed on dataset constructed from Research Gate website.

Results: We presented questions extracted from researchgate website into the system. The output consists of the corresponding POS tags and the category the question is classified into. The number of questions extracted from the website is dependent on the number of questions available in a forum. We were able to achieve a successful result of 3015 correctly extracted and classified questions at 80% POS tag occurrence.

Conclusion: Our approach to question identification and classification was effective and covers more question categories. This can be applied to any question answering system.

Keywords: Question; online forum; ResearchGate; Naïve Bayes; spam filtering.

1 Introduction

Online forums contain enormous amount of valuable user generated content, such as text and pictures in addition to links to other resources. An online discussion forum is a collection of threads; each thread consisting of the first post with subsequent reply post(s) [1]. These contents also contain questions and discussions (answers) about the questions and sometimes more related questions to the initial question which result in more discussion within the same discussion forum usually about a subject. A system capable of extracting and classifying questions becomes imperative to be able to find appropriate answers without reading all the content of a forum therefore building a question and answer (QA) system. Moreover, QA has been a tool in solving problems in specific domains (such as open and closed domain). Open domain question answering deals with questions about nearly anything and can only rely on general ontologies and world knowledge. Closed domain question answering deals with questions under a specific domain such as medicine or aeromechanic maintenance and can be seen as an easier task because natural language processing (NLP) systems can exploit domain specific knowledge frequently formalized in ontologies.

QA systems typically include a question classifier module that determines the type of question and the type of answer. After the question is analysed, the system typically uses several modules that apply increasingly complex NLP techniques on a gradually reduced amount of text; thus, a document retrieval module uses search engines to identify the documents or paragraphs in the document set that are likely to contain the answer, and a filter pre-selects small text fragments that contain strings of the same type as the expected answer. For example, if the question is "Who invented computer?", the filter returns text that contains names of people. Finally, an answer extraction module looks for further clues in the text to determine if the candidate answer can indeed answer the question. Identification and classification of question on QA sites have been crucial because the entire answer extraction process relies on finding the correct question type and hence the correct answer type [2].

Earlier question classification work includes [3] and [4] in which language model and Rapier rule learning was employed respectively. [5] proposed a machine learning approach, which uses the Sparse Network of Winnows (SNoW) learning architecture. [6] used linear support vector machines (SVMs) with question word bigrams and error-correcting output. Other works on question classification include: A function-based question classification technique proposed by [7] was built on Markov logic network (MLN), and tailored to general question answering. [8] worked on QA classification and sentential level ranking. Most successful research focuses on just five category of questions: Who, What, Where, Where, Which, Why and How (5W1H) which do not cover all the possible category for asking questions such as could, may, do and so on. Spam detection is omitted in most QA research and relationship between extracted question is not established to help improve answer retrieval. This research work is motivated by these outlined research works and some of their identified limitations.

Questions are extracted from post and threads obtained from an online forum (ResearchGate website). Similar questions for a category usually follow the same pattern, for example, “What is a dictionary?” or “What are the people in Nigeria called?”. We exploited these patterns using part of speech tags (POS) and based on the level of occurrence of these tag patterns in a sentence, it is either identified as a question or an ordinary sentence. The benefit of using POS is to accommodate the use of different words with the same meaning and also to handle the position of words in a post which could probably be as a result of the format in the construction of the sentence, error in typing or poorly formed sentence. To correctly classify the extracted question, we applied Naïve Bayes classifier to determine the category. Naïve Bayes classifier is a supervised machine learning probabilistic classifier that uses the label or attribute of a new instance to estimate the probability of each class or category [9]. It has been used in text classification research in different domains. Some applications of Naïve Bayes classifier can be found in [10] for heart disease prediction, [11] for intrusion detection and [12] for online randomised learning methods.

The content of this paper is structured as follows: In section two, we make a review of works focusing on question identification and extraction from web logs (blogs) and text. Our proposed method for spam detection, question identification and classification is described in section three. Section four discusses the experimental results and dataset construction. Evaluation for the performance of our model is discussed in section five, while conclusion and proposed further works are presented in section six.

1.1 Related works

Many QA systems used manually constructed sets of rules to map a question to a type, which is not efficient to manage. With the increasing popularity of statistical approaches, machine learning plays a more important role in this task. An advantage of the machine learning approach is that one can focus on designing insightful features, and rely on learning process to efficiently and effectively cope with the features. In addition, a learned classifier is more flexible to reconstruct than a manually constructed system because it can be trained on a new taxonomy when there are new changes.

Correct classification of question with respect to the expected answer type is prerequisite for question answering system. [13] proposed a novel architecture for 5W1H question classification and answer searching based on index scheme. Their proposed system performed analysis on crawled web document to extract question and applied constructed function called Indexer for classification. The indexer accepts TermSet (keywords), generated by a preprocessor as its input and generates the index by using an adapted pseudo code. The index is based on the type of answer expected with respect to the question. The system showed promising results than the existing systems based on question classification.

Lu et al. [9] discussed classification using Head Words and their Hypernyms where two models of classifier were used namely; Support Vector Machine (SVM) and Maximum Entropy (ME) model. Support Vector Machine is a useful technique for data classification. It uses kernel function for problem solution. Five feature sets (question wh-word, head word, WordNet semantic features for head word, word grams, and word shape feature) were used separately by the classifiers (SVM and ME) to determine their individual contribution. The experimental result was designed in two ways to test the accuracy of the classifiers. The first experiment evaluates the individual contribution of the classifiers for different feature types of question classification accuracy while the feature set was incrementally fed to both SVM and ME in the second experiment. The best accuracy achieved for 50 classes is 89.2% for SVM and 89.0% for ME.

Hong and Davison [14] explored the problem of extracting question answering content from discussion boards. In their research, they addressed both question detection and answer extraction. They focused on classification methods for question detection such as Question mark, 5WIH words, total number of posts within one thread, authorship, N-gram and answer detection using natural language techniques like position of the answer post, authorship, N-gram, stop words and query likelihood model score. They used Library for Support Vector Machine (LIBSVM2.88) as their classifier. The result of their research shows that; the use of N-grams and the combination of several non-content features can improve the performance of detecting question-related threads in discussion boards. The limitation of their research was the scope of question category considered (only 5WIH). They failed to address the questions available in later posts and did not consider the number of questions in the question posts.

Pal et al. [15] proposed a Minimally Supervised Question Classification and Answering based on WordNet and Wikipedia (Wikisense). This method was used for classifying questions into semantic categories in the lexical database like Word Net. In the database, a set of 25 Word Net lexicographer's file was taken from the titles of Wikipedia entry. They implemented and evaluated the proposed methods using a simple redundancy based QA system. In their experiment, they first run their QA system without any question classification as their baseline. They also run the system on the same evaluation dataset using two different question classifiers; one trained by WordNet and the other trained on WordNet plus WikiSense. At threshold of 2.25, the Mean Reciprocal Rank (MRR) was higher than the baseline by 0.061, and precision was higher by 9%.

Mishra et al. [16] presented their research work on question classification using machine learning approach. In order to train the learning model, they designed a rich set of features such as lexical, syntactic, and semantic that are predictive of question categories. The task of question classification was carried out as predicting the entity type of the answer of a natural language question. They tested their proposed approaches on the well-known University of Illinois at Urbana-Champaign (UIUC) dataset and succeeded to achieve a new record on the accuracy of 96.2% and 91.1% for coarse and fine grained classification on this dataset which outperforms every other state of the art result in their reviewed papers.

Ding et al. [17] addressed the issue of detecting spammers on community question answering (CQA) sites. They discovered that spammers are usually connected to other spammers via the best-answer relation, a pattern which cannot be easily detected for lack of identifiable textual patterns. Their proposed model incorporated the link-based information by adding regularization constraints to textual predictor. To evaluate their proposed approach, they crawled and constructed dataset from a CQA portal. Experimental results demonstrated that their method is more effective for spammer detection compared to other state of the art methods.

Ligozat [18] proposed Question Classification Transfer. Question answering systems have been developed for many languages, but most resources were created for English, which can be a problem when developing a system in another language such as French. In particular, for question classification, no labelled question corpus is available for French, so their paper studied the possibility of using existing English corpora and transferring classification by translating the question and their labels. By translating the training corpus, they obtained results close to a monolingual setting. This paper presented a comparison between two transfer modes to adapt question classification from English to French. Results showed that translating the training corpus gives better results than translating the test corpus. Part-of-speech information only was used, but since [17] showed that best results are obtained with parse trees and tree kernels, it could be interesting to test this additional information; yet, parsing translated questions may prove unreliable.

Ojokoh and Ayokunle [19] presented an online question and answer processing system. The user is allowed to generate the questions in an input field provided in an application interface. Their experiments were carried out using question subjects from various categories such as Facebook, google, laptops & notebooks, Wikipedia, YouTube and so on into which the dataset was classified. The question subjects from each category were supplied from the user's query. The experiment was also repeated using Levenshtein distance algorithm as well as a modified version of the algorithm proposed in the work. The results and evaluation from computing-related datasets demonstrated the effectiveness of their proposed technique.

Fong et al. [20] considered another approach to classifying forum questions using feature selection method called principal component analysis (PCA). Features from forum questions are extracted and then data mining techniques was applied to identify the relevant features that will help predict the quality of questions. Their classification model is used for testing new question posted to the forum to estimate the chance of being answered. This is done by comparing the features of the new questions to those that have been learned by the model from the previous records of questions from the forum, both that have received replies successfully and otherwise. They divided the quality of question into two classes: good and bad questions.

Their first task was to select attributes that define the quality of the questions. Secondly, they use selected features in the classification models by applying principal component analysis for providing in classifying between good and bad questions. The result showed success in question classification, and also provides guidelines on how to post questions that are likely to be answered.

In [21], the preliminary part of this work consisting of the design of an extensible question identification, extraction and classification model from weblog was presented. In the paper, a systematic approach to identification and classification of questions was proposed. A further and more detailed review of related works, a more comprehensive overview of the design with the incorporation of spam filtering and extensive experiments with several files from researchgate are presented here.

The above mentioned research recorded success in their results, however, [17,9,16] and [19] considered a small category of questions (Who, What, Where, Where, Which, Why and How) which did not cover many possible category of question that can be found in the web. [20] considered only two type of classification for question, which will not be sufficient to provide answers to questions. The achievements recorded by [21] and [17] did not take into consideration the possibility of the presence of spam from web content in question extraction and identification, which could save processing time, computing resources and enhance the possibility of quick identification of quality question from the forum. In addition, previous researches did not consider relationship between questions extracted from document to help provide better answer and information about the similarity in the structure of questions. This research introduces a context-based spam detector to eliminate spams in online forum and proposes Naïve Bayes classifier to classify the question using part of speech tags obtained from question template into thirteen different categories after the identification of questions has been achieved. Semantic relationship between extracted questions is constructed by computing cosine distance score between questions.

2 Methodology

Extracting quality questions from online blogs requires detailed processing and analyses of the text content to determine the existence and the category of questions in the blog.

2.1 System architecture

2.1.1 Pre-processing module

The pre-processing module, accepts web pages (or dataset files) as input, scans through the web pages in search for users' comments. In this module, unwanted content such as web tags and text formatting used in the creation of blogs are eliminated. The extracted blogs are scanned for spams to eliminate blogs that do not have meaningful contribution, and contains features used by spammers for generating content in order to make web pages appear active. The extracted useful blogs consist of a post and subsequent replies called threads (this post and thread form a group). Therefore, there exists many groups from the extracted blogs. Each group is treated as an entity and the comments in the group are broken down into sentences for further processing.

2.1.2 Extensible question configuration module

The question configuration module is a setup module for the system. It is responsible for specifying the category of questions to be extracted from blogs. It is extensible because it is possible for the system to be adjusted to detect more categories of questions. How precisely and effectively the system is able to extract questions depends on this module. The question configuration module accepts as input a question category or class and one or more sentence instance of the category; each of the instances is annotated with part of speech (POS) tag in English Language. The annotations for the instances are extracted to represent the sentences. The extracted annotations are mapped to the question category and stored for use in the question classification module. This module maintains a question category with one or more unique annotation instances.

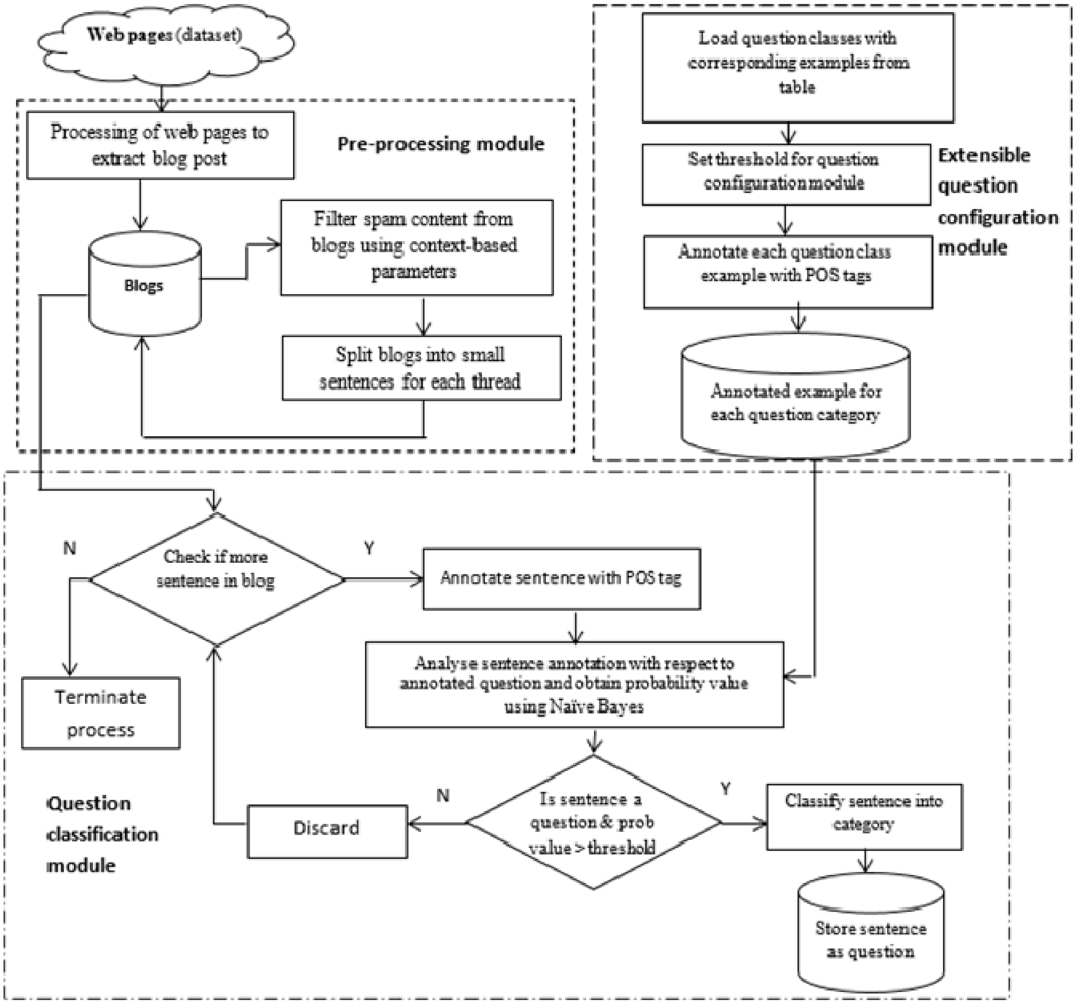
Web pages (dataset)
Processing of web pages to extract blog post
Pre-processing module
Blogs
Filter spam content from blogs using context-based parameters
Split blogs into small sentences for each thread
Load question classes with corresponding examples from table
Set threshold for question configuration module
Annotate each question class example with POS tags
Extensible question configuration module
Annotated example for each question category
N
Check if more sentence in blog
Y
Annotate sentence with POS tag
Analyse sentence annotation with respect to annotated question and obtain probability value using Naïve Bayes
Terminate process
Question classification module
Discard
N
Is sentence a question & prob value > threshold
Y
Classify sentence into category
Store sentence as question

Fig. 1. Proposed system architecture

2.1.3 Question classification module

The detection and classification of questions are done in the question classification module. The extracted blogs from the pre-processing module and the annotations for each question category from question configuration module are supplied into this module as input parameters. The classification module iterates through the groups extracted from the pre-processing module; for each group, it annotates all sentences with POS tag in English Language. The tags in each of the sentences are obtained and analysed with the question classification tags generated from the question configuration module. A relative probability value is assigned to each sentence based on the occurrence of question tags exploiting the probability computation of Naïve Bayes. The relative probability approach takes into consideration the position and occurrence of tags. Relative probability value for a tagged sentence that exceeds the threshold is regarded as a question and it is assigned to the category with the highest value, while tagged sentences below the threshold is discarded.

2.2 Context-based spam detection in blog

Some web pages containing blogs tend to employ techniques that give search engines a wrong impression about its content in order to have a higher rating which could result into higher monetary gain, increase in traffic, and personal benefits. Some of these techniques include stuffing blog with keywords to increase relevance and hyperlink to increase reference index. A blog is said to be a spam if the purpose is to increase the status of a blog or its related content without meaningful improvement to the viewer, which is achieved by constructing blogs using different techniques [22]. Several techniques have been considered to detect spam, which include feature analysis available on web content, such as content duplication, language model, compressibility and so on [22-25]. Another approach is the relationship between webpages to determine the presence of spam on the webpage [26-29]. In the following section, we present features adapted from [22] to detect spam from web content.

2.2.1 Average length of words

One major characteristic of spam is the use of composite word. This method takes normal or regular word, concatenating the words to form long composite words. Examples of such words are “computersystem”, “humanbone”, “phonecamera”, “bottlewater” and so on.The purpose of such words in blogs is to handle every category of misspelled words in search query by users using search engines when users omit space between words. This work adopts the method of [22], that investigated the average length (in characters) and the likelihood of spam, and discovered that many pages with average length of words between 8 and 10 are spam. Average length of words is computed as follows:

$$W^D = f_p(\{w \rightarrow BT\}_{j-1} \geq S_{(p)}) \tag{1}$$

$$\widehat{D_1} = \frac{W^D}{\lambda_n(\{w \rightarrow BT\}_{j-1})} \tag{2}$$

where f_p is a word length function that counts (→) the length of words (in character) w in $\{BT\}$. $S_{(p)}$is a predefined integer value (set at 10 in this research), which is used as a factor by $f_p$to determine if w should be considered, that is if it is equal to or exceeds the $S_{(p)}$ value. W^D is the total number of words w in $\{BT\}_{j-1}$, with length the same as or greater than the value of $S_{(p)}$ and j is 1,2,3, …,n the size of the extracted groups $\{BT\}$ in D. λ_n is a word-count function that counts all the words in a group $\{w \rightarrow BT\}_{j-1}$.$\widehat{D_1}$ is the mean occurrence of words greater than or equal to $S_{(p)}$.

2.2.2 Amount of anchor text

Another approach used by spam on the web is to stuff a page with anchor to other pages. The concept used by search engines is to determine the list of pages for a particular search query. For example, if there exists a word, “hospital” on page X that points to page Y, the search engine returns page Y if “hospital” is found in the search query, even if the keyword “hospital” is not found on page Y.

The average occurrence of anchor text in a blog can indicate if the blog is a spam. The average occurrence of anchor words ranges between 0.0 and 1.0, with reference to the result from [22], that used this parameter, it can be observed that higher fraction of anchor text may imply higher prevalence of spam. The results showed that pages with higher occurrence of anchor text became stable below the value of 0.75. Hence, in this work, occurrence of anchor text was found using:

$$\widehat{D_2} = \frac{\sum_j^H (f_a(\{w \rightarrow BT\}_{j-1}) = w_{(a)})}{\lambda_n(\{w \rightarrow BT\}_{j-1})} \tag{3}$$

where $\widehat{D_2}$ is the mean occurrence of anchor text in a blog group $\{BT\}$, $w_{(a)}$ is the anchor text, f_a is anchor-text function that gets the word w in each group $\{w \rightarrow BT\}$ that are anchor text and H is the size of blog group. λ_n is a word-count function that counts all the words in a group $\{w \rightarrow BT\}_{j-1}$.

2.2.3 Compressibility

Compressing the page to create compression ratio can reveal if such a page is a spam or not. The compression ratio measures the redundancy of the web page; it is computed by dividing the size of the uncompressed page by the size of compressed page. [22] used a fast and efficient gzip compression algorithm to compress pages to determine the compression ratio. The result from their research shows that a compression ratio with value of 4.0 and above was judged to be spam. However, in this research, hash table in place of gzip is used in the compression ratio from blog. The hash table contains distinct words from the blogs as key and the number of occurrence as the value. The compression ratio is measured by dividing size (number of words) in the blog using word-scanner by the size of the hash table. Word-scanner is an independent and sometimes part of an application capable of reading and counting the number of words in a text file or string. Therefore, we obtain the compression ratio as:

$$W^H = f_m(w\,,\ \sum\{w \rightarrow BT\}) \tag{4}$$

and

$$\widehat{D_3} = \frac{\lambda_n(\{w \rightarrow BT\}_{j-1})}{\lambda_h(W^H)} \tag{5}$$

where f_m is a word mapping function that creates a hash map W^H, that consists of words w in $\{BT\}$ and assigns (←) the number of times it occurred. $\widehat{D_3}$is the compression ratio and λ_h is a weight function that gets the size of W^H.

2.2.4 Fraction of globally popular words

The measure of N most popular words on a page can reveal the content of the page. N is obtained by searching all available blog groups for N most occurring words, where the size of N can be 200,300 or 500. Spam pages are usually stuffed with common words used by users of search engines, and which form part of search query. It is easy for spammers to create or fill pages with random selection of words from dictionary of any discipline. This metric was used in [22] and it was observed that based on a fraction of 500 most popular words, the prevalence of spam is modest throughout the distribution, with a dramatic spike for those few pages in which 75% or more of the popular words appear indicating spam. Equations 6 and 7 handle the fraction of globally popular words in a blog group.

$$W^N = f_N\big(\big[\{w \rightarrow BT\}_0\,, \{w \rightarrow BT\}_1\,, \dots, \{w \rightarrow BT\}_{j-1}\big] == N\big) \tag{6}$$

and

$$\widehat{D_4} = \frac{\lambda_n(W^N \xRightarrow{k=0,j-1} \{w \rightarrow BT\}_k)}{N} \tag{7}$$

where f_N is a word-scanning function that scan words from all the groups$\{w \rightarrow BT\}$, saving N most occurring words in W^N. λ_n is word-sum function that determine the size of W^N and λ_n is a word-sum function that gets the number of words in W^N in each group $\{w \rightarrow BT\}_k$.

2.2.5 Independent N-gram likelihood

N-gram of n consecutive words (where n could either be 3 or 4) is constructed from the extracted collection of blog groups. The n-gram is regarded as a probabilistic approach for predicting the occurrence of a sequence in a document [30]; it is a contiguous sequence of *n* items from a given sequence of text or speech. The probability of n-gram for a blog,$p(H_1, \ldots, H_j)$ with j n-grams is defined as:

$$p(H_1, \ldots, H_j) = \frac{\text{number of occurrences of n} - \text{gram}}{\text{total number of n} - \text{grams}} \tag{8}$$

$$\widehat{D_5} \ = -\frac{1}{j}\sum_{i=1}^{j} \log p(H_1, \ldots, H_j) \tag{9}$$

where $\widehat{D_5}$ is the independent n-gram likelihood.[22], set the value of n to be 3 and was able to show that documents composed of frequently occurring 3-grams words on a page for detecting spam. To determine the value of $\widehat{D_5}$, n is given a value of 3 and from the result of [22] values above 12.50 will be considered to be spam.

2.3 Heuristics combination with Fuzzy logic algorithm

Using individual heuristics discussed in section 3.2, will not be adequate to flag a blog as a spam. In this section, we explain how these heuristics are combined using an adapted fuzzy logic algorithm to detect spam. Fuzzy logic application cuts across several fields from artificial intelligence to control theory and achieved invaluable gain in performance and expected output [31]. The application of fuzzy logic algorithm for the detection of spam requires the combination of the heuristics parameters described in Table 1.

Table 1. Feature set used for spam detection

Parameter	$\widehat{D_i}$	**Threshold mark (T_i)**	**≥Threshold**	**Below threshold**
Average length of words	$\widehat{D_1}$	10	1	0
Amount of anchor text	$\widehat{D_2}$	0.75	1	0
Compressibility	$\widehat{D_3}$	4.0	1	0
Fraction of globally popular words	$\widehat{D_4}$	75%	1	0
Independent n-gram likelihoods	$\widehat{D_5}$	12.50	1	0

The threshold values described in the table is obtained from [22]; these values indicate evidence of spam in a document. The linguistic variables for spam detection is the "parameter" column and each value is referred to as linguistic value, S[t]= {compressibility, average length, amount of anchor text, fraction of globally popular words, independent n-gram likelihood}. Therefore, S[t] is the linguistic variable and compressibility, average word length is linguistic value, which are not fuzzy linguistic terms. A membership function λ_f, which converts the linguistic variables to fuzzy linguistic terms represented as $\widehat{D_1}$, $\widehat{D_2}$, and so on is achieved by using equation (2), (3), (5), (7) and (9).

$$\widehat{D_i} = \lambda_f(\mathrm{S}[\mathrm{t}_i]) \tag{10}$$

The fuzzy rulef_{rule} is created with the fuzzy linguistic term and the threshold mark. Table 2 shows the generated fuzzy rules, the rule comprises of IF-THEN-ELSE statements which consist of a condition and a conclusion. The condition compares the fuzzy logic term and the threshold mark. If the condition in the IF-THEN statement is true the value 1 is returned as the outcome and if it is false, 0 is returned which is the ELSE outcome. The outcome of the fuzzy rule is either 1 indicating spam or a 0 indicating a normal text.

Table 2. Fuzzy rule for spam detection

RULE 1	IF ($\widehat{D_1} \geq T_1$) THEN {return 1} ELSE {return 0}
RULE 2	IF ($\widehat{D_2} \geq T_2$) THEN {return 1} ELSE {return 0}
RULE 3	IF ($\widehat{D_3} \geq T_3$) THEN {return 1} ELSE {return 0}
RULE 4	IF ($\widehat{D_4} \geq T_4$) THEN {return 1} ELSE {return 0}
RULE 5	IF ($\widehat{D_5} \geq T_5$) THEN {return 1} ELSE {return 0}

Defuzzification is carried out on the aggregation of the returned values, F_{value} from the fuzzy rule. A threshold value of 2.0 is used to know if a text is a spam, the purpose of using 2.0 as a mark for the threshold is based on the fact that if any two conditions in the rule is satisfied then the text is a spam. A document that is spam will definitely satisfy more than one of the rules. Therefore, if the value of F_{value} is greater than or equal to 2.0 then it's a spam otherwise it's a blog. The computation of the defuzzification value is carried out in equation (11)

$$F_{value} = \sum_{i}^{k} f_{rule}(\widehat{D_\iota} > T_i) \tag{11}$$

where T_i is the threshold mark for rule i and k is the total number of rules.f_{rule} is the rule function created with the fuzzy linguistic term and the threshold mark and $\widehat{D_\iota}$ is the fuzzy linguistic terms for a document.

2.4 Question detection and extraction with POS

Every word in English Language can be classified into a particular part of speech such as noun, pronoun, verb, and so on. Tagging words in a sentence will help to perform the analysis on the sentence based on the position of the subject and the object in the sentence. Using tag for analysis eliminates and deals with issues like case sensitivity, grammatical construction and synonyms.

$$D_a = \delta_a(A \longrightarrow D) \tag{12}$$

$D_a = \{BT_0 \Rightarrow (d_a[0], d_a[1], \ldots, d_a[\mathrm{m}]), \ldots, BT_n \Rightarrow (d_a[0], d_a[1], \ldots, d_a[\mathrm{m}])\}$ is an array of labelled sentences with POS tag, A is a collection of part of speech tags in English Language (such as noun, verb, adjective, and so on). $\delta_a$is a tagging function that assigns ($\longrightarrow$) tags in A to Dusing Stanford parser that employs the principle of maximum entropy. Stanford parser takes a sentence as input and produces a labelled output of each word in the sentence with part of speech in English language.

Given category of question classes C = {c_1, c_2,…,c_j } each class having one or more question instance sentence e, which forms the group g allotted to a class, g = {e_1, e_2,…, e_k}. Therefore C={(g, c_1),(g,c_2)....,(g,c_j)}. Each document g in each question category is tagged with POS in English Language.

$$C_a = \delta_a(A \longrightarrow C) \tag{13}$$

where $C_a = \{(g_a, c_1), (g_a, c_2), \ldots, (g_a, c_j)\}$ is the category of question with POS tag for each group in each question class.

2.5 Question classification with Naïve Bayes

Naïve Bayes classifier is a machine learning-based text classification. It requires an initial set of data which is used in the learning process. It generates a set of rules that forms the decision criteria for classification. Its application can be found in authorship identification, age/gender identification, language identification, document-subject classification, sentiment analysis, medical diagnosis [32] among others. The training record for this system is obtained from $C_a = \{(g_a, c_1), (g_a, c_2), \ldots, (g_a, c_j)\}$, which is the output from tagging question instance in each group. The quality of the classifier is enhanced by increasing the content of the training record. The learned Naïve Bayes classifier assigns tagged document d, to its corresponding class c, and this is achieved by searching for the occurrence of g tags in D_a,and assigning it to a class c in C_a. The procedure/process is described as follows:

$$W_c = \underset{c \in C_a\{1,,j\}}{\operatorname{argmax}} p(C_k) \prod_{k=1}^{m} p(D_a|C_k) \tag{14}$$

$$p(d_a[i]|C_k) = \frac{count(a \rightarrow d_a[i], \mathrm{c}) + 1}{(\sum_{i=1}^{R} count(a_i, c)) + |V|} \tag{15}$$

where W_c={ $(Q[1], c)$, $(Q[2], c)$,…, $(Q[t], c)$} is the output of the Naïve Bayes classifier for document in D_a that contains questions $Q[1], Q[2], \dots, Q[t]$. Every document in D_a is searched for the occurrence of questions. The search is performed based on the presence of question tag in d_a for each class C_a. Questions found in document D_a is assigned a probability value based on the occurrence of tags found in d_a. Q is assigned a class in C based on the highest probability value of$c$.

$p(C_k) = \frac{N_e}{N}$ where$N_e$is the number of question instance sentences e for a question group c, and N is the number of question instance sentences in all the question groups. Equation (15) is Laplace (add-1) smoothing for Naïve Bayes which solves the common problem of maximum likelihood to avoid the occurrence of 0 probability. $|V|$ refers to the total number of unique tags in C_a. $count(a \rightarrow d_a[i], \mathrm{c})$ refers to the total number of unique tags a that belongs (→) to document $d_a[i]$ for class c. $d_a[i\ ]$is said to be a question Q for class c if the maximum probability value exceeds the question threshold mark Q_{TM}, the value is adjustable to regulate the rate or efficiency at which the system is a able to detect question in documents [33].

2.6 Semantic relationship between extracted questions

Considering a Question Q, extracted from document d_a belonging to a particular blog BT, the relationship between questions in the same thread can be obtained from W_c ={ $BT_0 => (Q[1], c)$, $BT_1 => (Q[2], c)$,…, $BT_n => (Q[t], c)$ }. Constructing semantic relationship between the questions extracted from the same thread can help to improve answer detection and play an important role in information retrieval [34], because those questions in the same thread may share the same answer paragraph.If there is more than one question extracted for BT, the semantic relationship can be achieved by computing the cosine distance or dot-product between two questions [35].

$$S(A, B) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_i^n (A_i)^2} \times \sqrt{\sum_i^n (B_i)^2}} \tag{16}$$

where A and B are lexical semantic vectors obtained from two questions to be compared and $S(A, B)$ is the similarity between two questions from which the vectors were obtained. The result ranges from zero (0) meaning exactly opposite, to positive one (1) meaning exactly the same and values in-between indicating intermediate dissimilarity or similarity [35]. In order to create a balanced level of semantical similarity between questions, a value greaterthan or equal to 0.5 will be considered semantically similar, this will show the closeness or similarity of questions in a particular group (blog).

Let $Q[1]$ and $Q[2]$ be questions from a group BT, and A and B be vectors of $Q[1]$ and $Q[2]$ respectively. A joint word set,$Q_{set} = \{w_1, w_2, \dots, w_n\}$ is constructed containing all the distinct words in $Q[1]$ and $Q[2]$.

$$Q_{set} = Q[1] \cup Q[2] \tag{17}$$

Each vector A and B is created by comparing each word in Q_{set} with $Q[1]$ and $Q[2]$ respectively. If w_i is present in$Q[1]$, the entry for A is set to 1. If w_i is not present, a similarity score(w_i, t_i), is computed between w_i and each word in $Q[1]$, the highest similarity score greater than the threshold is entered for A, otherwise 0 is set as the value for w_i. The reason for using threshold is as a result of the maximum similarity score could be very low indicating they are very dissimilar and could introduce noise to the vector if added [36].

The similarity score between w_i in Q_{set} and t_i in $Q[1]$ or $Q[2]$, is calculated by considering both the path length and depth in the hierarchical semantic WordNet.

$$s(w_i, t_i) = f_1(l) \times f_2(h) \times I(w_i) \times I(t_i) \tag{18}$$

$f_1$and $f_2$are transfer functions of path length(l), and depth(h) from the Semantic WordNet respectively.

length(l) refers to the shortest path between $w_i\ and\ t_i$ in the WordNet and depth(h) refers to the length of the path to t_i from the global root entity (node). Path$l(w_i)$ and $l(t_i)$ is information gain in Q_{set} and $Q[1]$ respectively, which is the probability of occurrence of w_i and t_i in Q_{set} and $Q[1]$ respectively, which can be computed as follows:

$$I(x) = 1 - \frac{\log(x+1)}{\log(\mathrm{N_{set}}+1)} \tag{19}$$

$I(x)$ can either be $l(w_i)$ or $l(t_i)$ and x is the total number of occurrence of w_i in Q_{set} or t_i in $Q[1]$ respectively and $\mathrm{N_{set}}$ is the total number of words in Q_{set}.

Semantic WordNet is a large database of English words, systematically arranged to form a tree-relationship between words. There are different techniques used to find semantic relationship between words these include corpus-based measure of semantic word similarity and a normalised common subsequence string matching algorithm [36].

3 Results and Discussion

3.1 Dataset construction

The dataset used for this research is made up one thousand (1000) web files obtained from ResearchGate website, a social networking website for scientists and researchers to collaborate and exchange ideas, ask and give answers to questions, share papers and often search for jobs. The website provides the platform to create a public and semi-public personal profile and to search for collaborators or people in similar area of interest [37]. The website started in 2008 with few features and over time it grew rapidly based on the contribution from users, scientist and researchers, and has more than seven million users as at 2015 [38] and participants cut across several fields, such as agriculture, medicine, computer science, engineering and so on. ResearchGate provides a feature for creating discussion board (blog) among users. A comment or question could be posted by a user and there will be different responses from other users. These responses could be a reply or answers to the initial post or question to spur further discussion. In our work, the created blog pages from the website were crawled using HTTRACK software, a free and open source downloadable application, for processing and question extraction. This software is capable of downloading webpages from a site on the internet to a local computer and still maintaining the site relative link structure so that the pages can be browsed on the local system [39,40]. Some of the advantages of the software include ability to resume interrupted download, configurable options to filter downloads and ability to follow links that are generated with JavaScript, flash and applets.

We implemented the Question Classification Module, Extensible Question Configuration Module, Pre-processing Module and the working inter-relationship between these modules with Java programming language and MySQL relational database management system as the database server. The tags that are used for extraction of text from ResearchGate blog is represented in Table 3. This system uses three extraction parameters to search for blogs on the web page dataset. These parameters were obtained by observing the mark-up structure of the web page.

Table 3. Extraction parameters

ID	Start Tag	End Tag
1	<h1 class=”topic-post-title”>	</h1>
2	<p>	</p>
3	<a class=”js-question-title topics-post-feed-item-title”>	</a>

The categories of questions that are considered in this research are specified in Table 4. The instances are used to determine if a sentence obtained from a blog is a question and Naïve Bayes is applied to determine the category the question.

Table 4 indicates categories of questions and the number of instances used in our research. Questions extracted are placed into one of these categories and the instances represent formats for the question category. The system uses these instances to identify a sentence as a question. Our system is flexible and allows for more categories to be added as well as instances for any category

Table 4. Question classification category

Question category	Number of instances	Question category	Number of instances
Any	16	Not	11
Ask	13	Perhaps	6
Can	15	Please	13
Could	14	Suppose	1
Do	16	Were	19
Excuse me	5	What	20
Have	18	When	12
How	13	Where	21
Is	9	Who	10
Let	2	Why	17
May	15	Would	17
Might	18		

3.2 Discussion of results

Table 5 shows the results from question identification and classification. The table shows the questions extracted from the blog; the corresponding POS tags and the category the question is classified into. The number of questions extracted from a group varies and it is dependent on the number of questions available. The result displayed is the question extracted from group 1. Results are obtained from the 1000 dataset files crawled from ResearchGate website. Each file is a forum (or discussion) thread which represents a group and it comprises of one or more sentences. A total of 24,911 sentences were obtained and the number of questions extracted is dependent on the value of the percentage of question tag occurrence.

Table 6 shows a summary of the number of questions extracted and classified at 50% questions tag occurrence. There are five columns in each of the tables which indicate how the questions for a particular category are classified. The first column represents the identification (ID) number for each question category and the second column represents the question categories used in this research, which are assigned to a question extracted from the blog. The third column refers to the total number of questions identified for that category, while the fourth and fifth column represents the number of questions that are accurately classified for that category and the number that are incorrectly classified for that question category respectively. The sixth column signifies questions that are extracted and classified for that question category but are not really question, which could be as a result of the extracted text meeting the classification structure and requirements.

Table 5. All the question extracted from group 1

Extracted question	POS Tag	Question category
I want to ask you if you would like to provide me a benchmark: a table + a work load (for large databases).	PRP VBP TO VB PRP IN PRP MD VB TO VB PRP DT NN DT NN CC FW FW JJ NN	Would
I would like to have access to a public data base of proteins to compare expression of proteins in *Pinus radiata*	PRP MD VB TO VB NN NN JJ NNS NN IN NNS RB VBP NN IN NNS IN NN NN	Have
I want to learn how these ingredients work	PRP VBP TO VB WRB DT NNS VBP	How
Many people find supplements but how do you know they are produced good.	JJ NNS VBP NNS CC WRB VBP PRP VB PRP VBP VBN NN	How
Can anybody tell me what possibilities are available to improve the algorithm of existing paper (base paper)	JJ NN VB PRP WP NNS VBP JJ TO VB DT NN IN VBG NN FW FW	What
In TGCA analysis, what defines the base level of genes in the samples	JJ NN NN WP VBZ DT NN NN IN NNS IN DT NNS	What

The summary for different percentages of tag occurrence for identification and classification of questions is represented in Table 7. The different percentage value is selected for checking the occurrence of question based on the sequential changes of the experimental results. The higher the percentage value, the higher the strictness of the system to check for the tag occurrence to determine if the sentence is a question. At 50% tag occurrence, a total of 3081 questions were extracted and 3048 were correctly classified while 3 were wrongly classified and 30 that were classified are not questions. 3055 questions were obtained from 60% tag occurrence and 3048 questions was rightly classified, while a total of 1 and 6 are the values for questions wrongly classified and questions that were classified but are not question respectively. Tag occurrence of 70% and 80% have no values for neither questions wrongly classified nor classified questions that are not questions. There is a significant difference between the questions that are rightly classified at 70% and 80% with 33 rightly classified questions difference and this is as a result of the value of the tag occurrence, for 90% and 100% the value is expected to reduce even further based on the result obtained for 85% tag occurrence. 70% tag occurrence value produced the required results for the system because all the questions that were extracted were correctly classified.

Fig. 2a to 2d are graphical representation of the summary of the question extracted for different percentage value of question tag occurrence. The questions extracted, rightly classified questions, wrongly classified and extracted but not questions are plotted against the tag occurrence values: 50%, 60%, 70% and 80% respectively. Fig. 2a shows the questions extracted for the different percentage values and it shows that the lower the percentage value the higher the number of questions discovered and correspondingly there will be more sentence which are not questions but detected and extracted as questions as it can be seen in Fig. 2c. Fig. 2b shows questions correctly classified, where the extracted sentences are correctly classified as questions. The higher the percentage occurrence value the lower the questions extracted as seen between the 70% and 80% mark. Fig. 2c shows sentences that are extracted as questions but given a wrong classification identity. Also, the figure shows that the mark of 70% and above do not have wrongly classified questions.

Table 6. Question category classification summary at 50% question tag occurence

Question category	Number of questions extracted for category	Number of questions rightly classified	Number of questions wrongly classified	Number of content extracted but not questions
Any	19	19	0	0
Ask	23	23	0	0
Can	88	87	0	1
Could	31	31	0	0
Do	45	42	0	3
Excuse me	4	3	0	1
Have	329	326	0	3
How	354	353	0	1
Is	9	6	2	1
Let	5	5	0	0
May	53	48	0	5
Might	22	22	0	0
Not	6	4	0	2
Perhaps	7	7	0	0
Please	18	18	0	0
Suppose	16	11	1	4
Were	66	62	0	4
What	367	367	0	0
When	326	324	0	2
Where	295	295	0	0
Who	362	361	0	1
Why	386	386	0	0
Would	250	248	0	2

Table 7. Summary of question extracted and classified for different question tag occurence

Tag occurrence	Content extracted as question for category	Question rightly classified	Question wrongly classified	Extracted but not question
50%	3081	3048	3	30
60%	3055	3048	1	6
70%	3048	3048	0	0
80%	3015	3015	0	0
85%	3000	3000	0	0

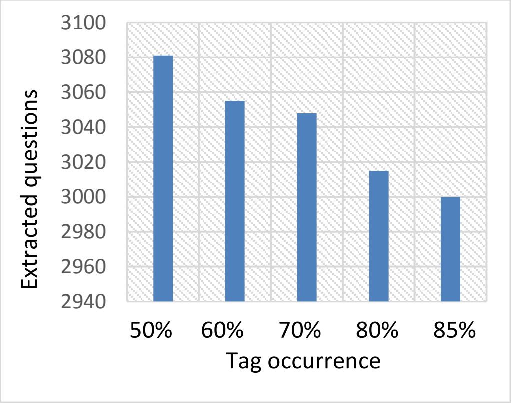

Fig. 2a. Content extracted as question

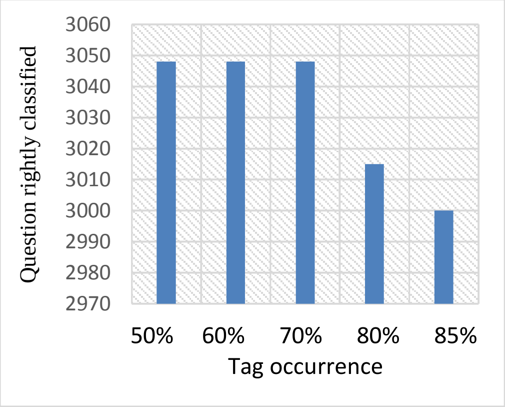

Fig. 2b. Question correctly classified

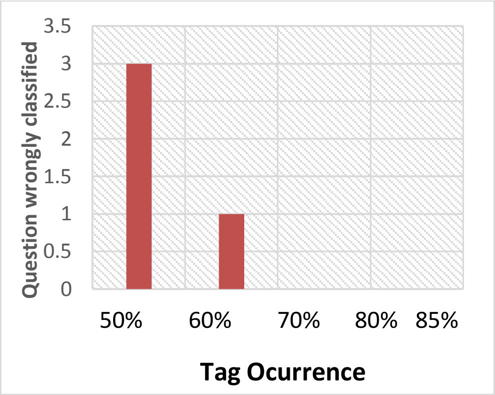

Fig. 2c. Question wrongly classified

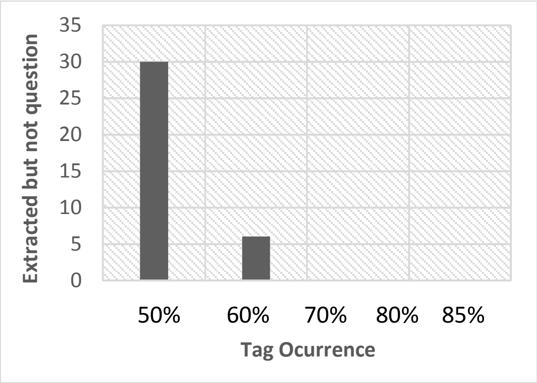

Fig. 2d. Extracted but not question

3.3 Evaluation

The results of the question classification and identification of the system were evaluated using precision, recall, accuracy and F-measure values as defined as follows:

$$\text{Precision}=\frac{A}{A+C}\text{ , Recall }=\frac{A}{A+B}\text{ , Accuracy }=\frac{A+D}{A+B+C+D}\text{ , F-measure }=\frac{2*precision*Recall}{precision+Recall}$$

A= Number of correctly classified questions for a category
B= Number of questions found (existing) but not classified as question for a category
C= Number of questions that are wrongly classified
D= Number of question that are not found (existing) but not classified as question for a category

The evaluation results of question extraction based on tag occurrence is presented in Fig. 3a and 3b, which shows the accuracy and f-measure respectively. The accuracy and f-measures are plotted against tag occurrence. It can be seen from the graph that at the tag occurrence between 70% and 85% the question identification is high and significantly different from the tag occurrence 50% and 60%. The model is optimal at 70% tag occurrence mark.

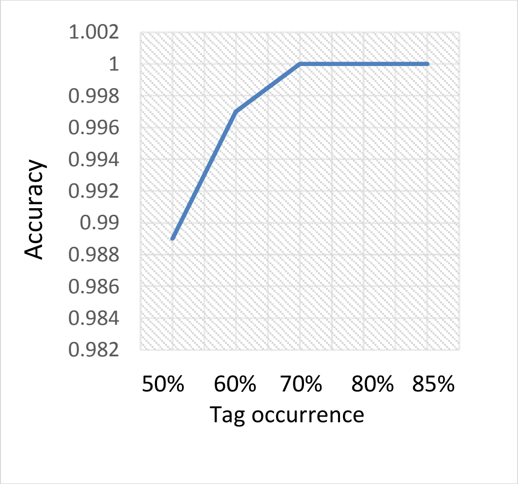

Fig. 3a. Overall question identification accuracy

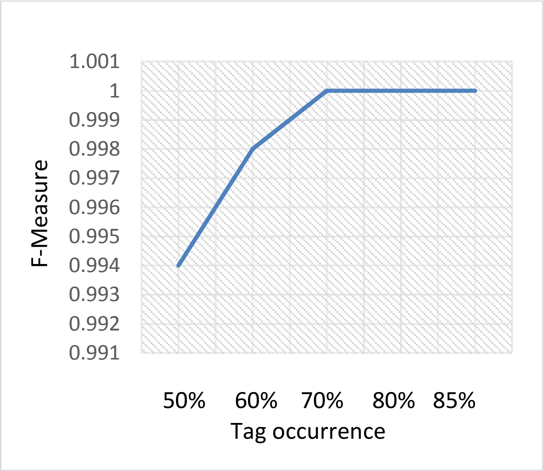

Fig. 3b. Overall question identification F-measure

Table 8 shows the abbreviations of features used in question extraction and classification models obtained from [41]. Table 9 shows the comparative result analysis of the Naïve Bayes classifier used against other classifiers. It can be seen that combination of AUTH+QM+5W+LEN achieved a good performance but our proposed model obtained a better result using the evaluation metrics of precision, recall, accuracy and f-measure.

Table 8. Features and their abbreviations from [41]

Features	Abbreviations
Question Mark	QM
5W1H Words	5W
Total # Posts	LEN
Sequential Patterns	SPM
N-grams	NG
Authorship	AUTH
Position	POSI
Query Likelihood Model	LM
Stop Words	SW
Graph+Query Likelihood Model	GQL
Graph+KL-divergence Model	GKL

Table 9. Evaluation result from comparison with other models

Features	Precision	Accuracy	Recall	F-measure
QM+5W	0.614	0.648	0.764	0.681
5W+LEN	0.627	0.650	0.709	0.666
SPM	0.642	0.661	0.702	0.671
QM+LEN	0.656	0.687	0.764	0.706
QM+5W+LEN	0.672	0.698	0.755	0.711
NG	0.752	0.772	0.799	0.775
AUTH+LEN	0.813	0.839	0.874	0.843
AUTH+QM+5W+LEN	0.863	0.876	0.889	0.876
NB	**0.999**	**0.996**	**0.997**	**0.998**

Table 10 describes a cross section of comparative characteristics in question extraction models. The comparison is carried out on [41,17] and our model. The characteristics depict features obtainable in the models which describes the quality of the model. The more the features present in a model the better the proficiency. The table shows that our model has all the characteristics taken into consideration, which makes it a better and comprehensive model.

Table 10. Characteristics features in question extraction models

Characteristic features	NBM	[41] Model	[17] Model
Spam detection (context based spam detection)	Yes	No	Yes
Removing spams (heuristic measure)	Yes	No	No
Checking all the questions in the blog	Yes	No	Yes
Part of speech tagging	Yes	No	No
Question detection and classification	Yes	Yes	No
Semantic relationship between extracted questions	Yes	No	No

4 Conclusion

To obtain quality answers from a community question answering system accurate question identification and extraction becomes imperative. In this paper, we presented an instance based technique using part of speech tag to identify and extract questions and Naïve Bayes classifier for question classification. We also employed context based features such as average length of words, amount of anchor text, compressibility, fraction of globally popular words and independent n-gram likelihoods to scan for spam in text posted by users or in blogs in order to expunge irrelevant content and enhance extraction of high quality questions. In our research, the model extracting and identifying quality question was defined and the result of our model showed impressive performance with respect to other models. However, there are areas for further research for our work to improve the quality of questions identified, extracted and classified. To improve the question extraction and classification result of our research we intend to create a hybrid model that combines our approach with another successful technique such as SVM which was used in [9] to observe how it will improve question extraction. In addition, the assigned threshold value for our model worked well with the dataset from Research Gate. We intend to apply similar values to other dataset to know what value works best for different datasets.

ACKNOWLEDGEMENTS

This work was partially supported by the COMSTECH-TWAS Joint Research Grant (14-014 RG/ITC/AF/AC_C-UNESCO FR: 3240283404).

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Hong L, Davison BD. Empirical study of topic modeling in twitter. In Proceedings of the first workshop on social media analytics. ACM. 2010;80-88.

[2] Allam AM, Haggag MH. The question answering systems: A survey. International Journal of Research and Reviews in Information Sciences (IJRRIS). 2012;2(3).

[3] Schaufeli WB, Martinez IM, Pinto AM, Salanova M, Bakker AB. Burnout and engagement in university students a cross-national study. Journal of Cross-cultural Psychology. 2002;33(5):464-81.

[4] Radev D, Fan W, Qi H, Wu H, Grewal A. Probabilistic question answering on the web. Journal of the American Society for Information Science and Technology. 2005;56(6):571-83.

[5] Li X, Roth D. Learning question classifiers. In proceedings of the 19th international conference on Computational linguistics. Association for Computational Linguistics. 2002;1(1-7).

[6] Hacioglu K, Ward W. Question classification with support vector machines and error correcting codes. In proceedings of the 2003 conference of the north American chapter of the association for computational linguistics on human language technology: Companion volume of the proceedings of HLT-NAACL 2003--short papers. Association for Computational Linguistics. 2003;2(28-30).

[7] Bu F, Zhu X, Hao Y, Zhu X. Function-based question classification for general QA. In Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics. 2010;1119-1128.

[8] Gupta S, Malhotra S. Question answering system based on question classification and sentential level ranking. International Journal of Computer Applications. 2014;93(15).

[9] Lu SH, Chiang DA, Keh HC, Huang HH. Chinese text classification by the Naïve Bayes classifier and the associative classifier with multiple confidence threshold values. Knowledge-based Systems. 2010;23(6):598-604.

[10] Subbalakshmi G, Ramesh K, Rao MC. Decision support in heart disease prediction system using naive bayes. Indian Journal of Computer Science and Engineering (IJCSE). 2011;2(2):170-6.

[11] Mukherjee S, Sharma N. Intrusion detection using naive Bayes classifier with feature reduction. Procedia Technology. 2012;4:119-28.

[12] Godec M, Leistner C, Saffari A, Bischof H. On-line random naive bayes for tracking. In Pattern Recognition (ICPR), IEEE. 2010 20th International Conference on 2010;3545-3548.

[13] Mudgal R, Madaan R, Sharma AK, Dixit A. A novel architecture for question classification based indexing scheme for efficient question answering. arXiv preprint arXiv:1307.6937; 2013.

[14] Hong L, Davison BD. A classification-based approach to question answering in discussion boards. In proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval. ACM. 2009;171-178.

[15] Pal A, Chang S, Konstan JA. Evolution of experts in question answering communities. In ICWSM; 2012.

[16] Mishra M, Mishra VK, Sharma HR. Question classification using semantic, syntactic and lexical features. International Journal of Web & Semantic Technology. 2013;4(3):39.

[17] Ding Z, Gong Y, Zhou Y, Zhang Q, Huang X. Detecting spammers in community question answering. In IJCNLP. 2013;118-126.

[18] Ligozat AL. Question classification transfer. In ACL. 2013;(2):429-433.

[19] Ojokoh B, Ayokunle P. Online question answering system. International Journal of Computer Science. 2013;3(03):02-9.

[20] Fong S, Zhuang Y, Liu K, Zhou S. Classifying forum questions using PCA and machine learning for improving online CQA. In International Conference on Soft Computing in Data Science. Springer Singapore. 2015;13-22.

[21] Ojokoh B, Igbe T, Araoye A, Ameh F. Question identification and classification on an academic question answering site. In Digital Libraries (JCDL), 2016 IEEE/ACM Joint Conference. IEEE. 2016; 223-224.

[22] Ntoulas A, Najork M, Manasse M, Fetterly D. Detecting spam web pages through content analysis. In Proceedings of the 15th International Conference on World Wide Web. ACM. 2006;83-92.

[23] Mishne G, Carmel D, Lempel R. Blocking blog spam with language model disagreement. In AIR Web. 2005;5(1-6).

[24] Svore KM, Wu Q, Burges CJ, Raman A. Improving web spam classification using rank-time features. In Proceedings of the 3rd international workshop on Adversarial information retrieval on the web. ACM. 2007;9-16.

[25] Fetterly D, Manasse M, Najork M. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS. ACM. 2004;1-6.

[26] Castillo C, Donato D, Becchetti L, Boldi P, Leonardi S, Santini M, Vigna S. A reference collection for web spam. In ACM Sigir Forum. ACM. 2006;40(2):11-24.

[27] Geng GG, Li Q, Zhang X. Link based small sample learning for web spam detection. In Proceedings of the 18th international conference on world wide web. ACM. 2009;1185-1186.

[28] Zhou D, Burges CJ, Tao T. Transductive link spam detection. In Proceedings of the 3rd international workshop on Adversarial information retrieval on the web. ACM. 2007;21-28.

[29] Drost I, Scheffer T. Thwarting the nigritude ultramarine: Learning to identify link spam. In European Conference on Machine Learning. Springer Berlin Heidelberg. 2005;96-107.

[30] Jurafsky D, Manning C. Natural language processing. Instructor. 2012;212(998):3482.

[31] Pelletier FJ. Hájek Petr. Metamathematics of fuzzy logic. Trends in logic, vol. 4. Kluwer Academic Publishers, Dordrecht, Boston, and London, 1998, viii+ 297 pp. Bulletin of Symbolic Logic. 2000;6(03):342-6.

[32] Rish I. An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence. IBM New York. 2001;3(22):41-46.

[33] Reschke K, Vogel A, Jurafsky D. Generating recommendation dialogs by extracting information from user reviews. In ACL. 2013;(2):499-504.

[34] Rodríguez MA, Egenhofer MJ. Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering. 2003;15(2):442-56.

[35] Jimenez S, Duenas G, Baquero J, Gelbukh A, Bátiz AJ, Mendizábal A. UNAL-NLP: Combining soft cardinality features for semantic textual similarity, relatedness and entailment. In Proceedings of the 8th International Workshop on Semantic Evaluation. 2014;732-742.

[36] Li Y, Bandar ZA, McLean D. An approach for measuring semantic similarity between words using multiple information sources. IEEE Transactions on knowledge and data engineering. 2003;15(4):871-82.

[37] Lin T. Cracking open the scientific process. New York Times. 2012;16;16:D1.

[38] ResearchGate. 8 out of 8 million. ResearchGate; 2015.
Available:https://www.researchgate.net/blog/post/8-out-of-8-million
(Accessed 21 October 2015)

[39] Engebretson P. The basics of hacking and penetration testing: Ethical hacking and penetration testing made easy. Elsevier; 2013.

[40] Beaver K. Hacking for dummies. John Wiley & Sons. 2012;278,280–281.
ISBN: 9781118380963

[41] Hong L, Davison BD. A classification-based approach to question answering in discussion boards. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. ACM. 2009;171-178.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://sciencedomain.org/review-history/18868