Computer Speech & Language 60 (2020) 101023

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl

A question answering system in hadith using linguistic knowledge

Asad Abdi[a,*], Shafaatunnur Hasan[a], Mohammad Arshi[b], Siti Mariyam Shamsuddin[a], Norisma Idris[b]

[a] UTM Big Data Centre (BDC), Universiti Teknologi Malaysia, Johor, Malaysia
[b] Department of Artificial Intelligence Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia

ARTICLE INFO

Article History:
Received 28 March 2017
Revised 7 September 2019
Accepted 9 September 2019
Available online 12 September 2019

Keywords:
Question answering
Hadith
Text mining
Information extraction
Query expansion

ABSTRACT

Question answering system aims at retrieving precise information from a large collection of documents. This work presents a question answering method to apply on Hadith in order to provide an informative answer corresponding to the user's query. Hadith englobes stories and qualification of the prophet Muhammad (PBSL). It also includes the sayings of his companions and their disciples.

The problem with current methods is that they fail to capture the meaning when comparing a sentence and a user's query; hence there is often a conflict between the extracted sentences and user's requirements. However, our proposed method has successfully tackled this problem through: (1) avoiding extract a passage whose similarity with the query is high but whose meaning is different. (2) Computing the semantic and syntactic similarity of the sentence-to-sentence and sentence-to-query. (3) Expanding the words in both the query and sentences to tackle the fundamental problem of term mismatch between sentences and the user's query. Furthermore, in order to reduce redundant Hadith texts, the proposed method uses the greedy algorithm to impose diversity penalty on the sentences. The experimental results display that the proposed method is able to improve performance compared with the existing methods on Hadith datasets.

1. Introduction

In recent years, due to the huge amount of information, the users need search engines to obtain more and more information conveniently. However, for a specific question, a search engine collects a large amount of information, in which there may be massive redundant or irrelevant data. Therefore, users have to spend too much time in seeking for useful data. Users often have specific questions in mind, for which they hope to get answers. They would like the answers to be short and precise, and they always prefer to express the questions without any query formation rules, or even a specific knowledge domain.

Question Answering (QA) system can be an essential technology to tackle the aforementioned problems. It is a technology that locates, extracts, and represents a specific answer to a user's question expressed in natural language (Pavlić et al., 2015). It can provide answers to the user queries in a concise form (Lu et al., 2012). It combines information retrieval and information extraction techniques to present short answers to the user 's questions posed in a natural language (Peral et al., 2014).

* Corresponding author.
E-mail address: seyedasadollah-pd@utm.my (A. Abdi).

https://doi.org/10.1016/j.csl.2019.101023
0885-2308/

Natural Language Processing (NLP) in the Arabic language is still in its initial stage compared to the work in the English language, which has already benefited from the extensive research in this field. Despite this fact, many researchers around the world accomplished significant experiments upon Quranic Arabic in the past decade. They enriched the Quran text with annotations, capture its linguistic structure, and provide tools to perform automated, objective inference and querying (Atwell et al., 2011). However, another important Islamic source, Hadith, is almost overlooked by most academics in computer science.

Hadith derived from the Arabic word "Hadatha", "حدث" means news and story. Most scholars showed that there were strong agreements on what a "Hadith" is. Hadith is any speech, discussion, action, approval, and physical or moral description attributed to the prophet Muhammad, whether supposedly or truly (Batyrzhan et al., 2014). Hadith includes three main parts: (1) Matn (the central text), (2) Isnad (chain of narrators), and (3) Taraf, the beginning phase(s) of Hadith, which indicates the sayings, actions or characteristics of the prophet. Hadith has special importance in Islamic creed because it is the second source of legislation after the holy Quran. Therefore, the Hadith is available in a very large number over the web, Hadith corpus and digital library. Obviously, there should be available tools to be used in order to return the most likely answers to a given question. For this purpose, several methods have been proposed to extract the more question relevant Hadith from the Hadith corpus. There are three main problems with the current Hadith QA systems as follows:

(1) In-text relevance context, syntactic information, such as word-order, can provide useful information to distinguish the meaning of two sentences when two sentences share similar bag-of-words. For example, "*Teacher helps* student, *المعلم يساعد التلميذ*" and "*student helps teacher, التلميذ يساعد المعلم*" will be judged as similar sentences because they have the same surface text. However, these sentences convey different meanings. Most of the existing methods do not contribute syntactic information to calculate the similarity measure Query-to-Sentence (*Q2S*). They fail to capture the meaning in the comparison between the user's query and a sentence; hence sometimes the answers conflict with the information expressed in user's query. However, for the correct computing of similarity measure and to identify more relevant answers to the user's question, a method should take into account both semantic and syntactic information.
(2) The second problem with the current question answering system on Hadith set is that the submitted queries are combined with limited words or phrases. As a result, in many cases, these queries are too short to describe users' information needs clearly and completely. A possible solution to this problem lies in query expansion, a technique which has been applied successfully in various areas of information retrieval.
(3) The third problem with the current question answering system on Hadith set is redundant Hadith texts. Two different texts (two Hadith or two answers) may include similar content or convey similar information. Hence, the removal of such redundancies and keeping only one answer will give higher precision.

This paper presents a method to extract more relevant Hadith for a specific question. In order to determine an answer (Hadith) for user's question, we consider two types of features for each answer (Hadith): 1) the similarity measure between the current Hadith and the user's query using the combination of semantic and syntactic information; 2) the similarity measure between the current Hadith and other Hadiths from the Hadith set using the combination of semantic and syntactic information. Our method also expands the user's question to find more relevant Hadith from a Hadith set. Finally, the method removes redundant Hadith texts. The proposed method is called Question Answering System in Al-Hadith using Linguistic Knowledge (ASHLK).

The rest of this article is organized as follows. Section 2 presents a short overview of the previous methods that are used to extract Hadith. Section 3 presents the proposed method. Experiences and Evaluation results are reported in Section 4 and finally, Section 5 concludes this paper.

2. Literature review

In the recent past, the query-answering system has become one of the investigated subjects in NLP. The main goal of QA systems is to retrieve small pieces of texts that contain the answer to the question rather than the list of documents, traditionally returned by search engines. A QA system can be divided into factoid QA and non-factoid QA. A factoid QA asks about the facts and typically a single correct answer exists. In other words, questions match short answers, usually in the form of named or numerical entities (e.g., "*how many people live in Canada?*"; "*Where was Y born?*"). A non-factoid QA does not necessarily ask about facts. It can often be answered with opinions, descriptions and diverse sentences. In this paper, we focus on the non-factoid question-answering to retrieve sentences that are part of the answers. We now give a formal definition of our task. Given a non-factoid question, Q, and a document, D, our task is to find sentences that are part of answers.

Several studies have shown that computer can be used for generating answers based on the user's query. Therefore, several methods have already been proposed for the non-factoid QA and factoid QA. Since the Quran and Hadith are two important sources in the Islam world, recently, researchers proposed several methods to extract Hadith based on the user's query. In this section, we explain some of these methods as follows.

The halal term refers to any action or object that is permitted or allowed according to Islamic Law. Many scripts about halal products are available through resources such as web pages, e-books, and magazines. The end-user may inquire halal-related data via query phrases to fetch a list of relevant documents. To address this scenario, Hanum et al. (2014) scrutinized topic analysis techniques that are Latent Semantic Indexing (LSI) (Papadimitriou et al., 1998) and frequency-based inverted indexing. The experiment performed upon four Malay translated Hadith documents, which contain 436 words, and 36 other Malay language documents. In the dataset, 16 documents encompass various aspects of halal-related subjects. To obtain a vector of words, after tokenization and

eliminating stop words, all tokens are converted to their root form using a Malay language stemmer. The similarity between the query and the documents is measured by the cosine similarity technique (Tata and Patel, 2007). Five sets of queries, which contain words about halal products, are constructed. In order to evaluate the successfulness of the technique, the dataset is manually analyzed and a list of relevant judgment is compiled. The experiment proves that utilizing LSI gives better results than frequency analysis. However, LSI needs high computational resources to deal with large documents such as web pages.

Shatnawi et al. (2011) explained an experiment that contains two major steps: 1) retrieve Hadiths from web pages and 2) verify the correctness of the retrieved Hadiths. They used a database, which is provided by Sheikh Al-Albani's, that contains more than 17,000 Hadith texts along with their authentication degrees. Shatnawi et al. (2011) illustrate how to tokenize the database and remove stop words. Except for 28 Arabic alphabets, all characters are removed including Arabic vowel marks. Finally, a positional index is built, which contains more than 56,000 terms. In order to extract Hadith texts from web pages, a Java HTML cleaner eliminates HTML codes from the web pages. Then, four contiguous words from the web page are compared with the Hadith positional index to detect Hadith text. When all Hadith texts are fetched, each one is looked up in the database to determine their degree of authenticity.

Guirat et al. (2016) proposed a hybrid indexing system for Arabic document indexing. In the pre-processing stage, the basic linguistic functions: stemming, diacritic removal, tokenization, stop words removal and normalization are applied on both query and document. In the Insertion method, the system computes the frequency of each stem. It also used the structure (stem, root, pattern and frequency parameters) for retrieval process.

Abderrahim et al. (2013) proposed a system to retrieve the relevant documents based on the user's query. First, the stop words and segmentation processes are applied to both documents and the user's query. After eliminating the stop words, the concepts from the documents and query are extracted using Arabic WordNet (AWN). Finally, the system retrieves the senses of those concepts from the synsets of AWN in order to improve the retrieval results.

BEKHTI and AL-HARBI (2013) proposed AQuASys, an Arabic factoid QA system. The AQuASys includes three phases: (1) a question analysis phase; (2) a sentence filtering phase, and (3) an answer extraction phase. The question analysis phase processes the user's query to extract the key information such as noun, question's verb and question's keywords. The sentence filtering phase identifies the more relevant sentences to the user's question. The current phase used the information extracted in the previous phase. The relevant sentence is determined based on the presence of the user's query keywords in these sentences. The answer extraction phase presented the sentences that contain accurate answer.

Brini et al. (2009) also introduced a factoid QA system, QASAL, based on the use of the NooJ's platform (Mazauric and Rothiot, 2007). The system includes three steps. (1) Question analysis; (2) Passage retrieval; (3) Answer extraction. The Question analysis step takes, as input, any Arabic question. The main task of this step is to extract the important information from the question. The information can be expected an answer and the list of keywords. Passage retrieval step is the core of the system. It retrieves the passages which are estimated as relevant to contain the expected answer. The Answer extraction step extracts the accurate answer from the retrieved passage. This step considers the type of answer expected by the user.

Qarab (Hammo et al., 2002) is a QA system in the Arabic language to identify text passages that answer a natural language question. The system is based on Information Retrieval (IR) and NLP techniques. The IR step aims to identify the candidate documents that may include the answer; then the NLP step analyzes the top-ranked documents returned by the IR step to return sentences that may contain the answer.

ArabiQA (Benajiba et al., 2007a) integrates NER system (Named Entities Recogniser) (Friburger et al., 2002) and the Java Information Retrieval System (JIRS) (Benajiba et al., 2007b). ArabiQA includes the following steps: (1) question analysis identifies the question keywords and the named entities emerging in the question; (2) passage retrieval retrieves the documents which are guesstimated as relevant to include the answer; (3) answer extraction elicits a list of candidate answers from the relevant documents; (4) answers validation determines the most correct answer.

Arabic QAS (Kanaan et al., 2009) is a question answering system. In this system, the query and documents are processed by the question analyser step. The current step includes a set of basic linguistic functions such as tagger and tokenizing. Then a set of relevant documents that may include the answer are retrieved by the information retrieval step. Finally, the answer generation step extracts the passages of the relevant documents that include most of the words appearing in the user's query.

Summing up, in this section, different kinds of the method have been presented to produce answer based on a user's question.

The main problem with those systems is that they fail to capture the meaning when comparing a sentence-to-query; hence there is a conflict between extracted sentences and user's need. Considering the relationship between the words (syntactic composition) can help in identifying query relevant sentences. In this paper, we propose a method that is able to integrate both the semantic and syntactic information using a linear equation to capture the meaning of two sentences (*the query is also considered as a sentence)*, when a query and a sentence have same surface text (*the words are the same*) or they are a paraphrase of each other. However, the proposed method is able to avoid selecting the sentence whose similarity with the query is high, but its meaning is different.

3. Proposed method: ASHLK

In this section, we present the overall structure of the proposed method as shown in Fig. 1. It includes the following steps:

1. Performing pre-processing tasks on sentences and the user's query. It aims to prepare the sentences and the input query for the subsequent steps.

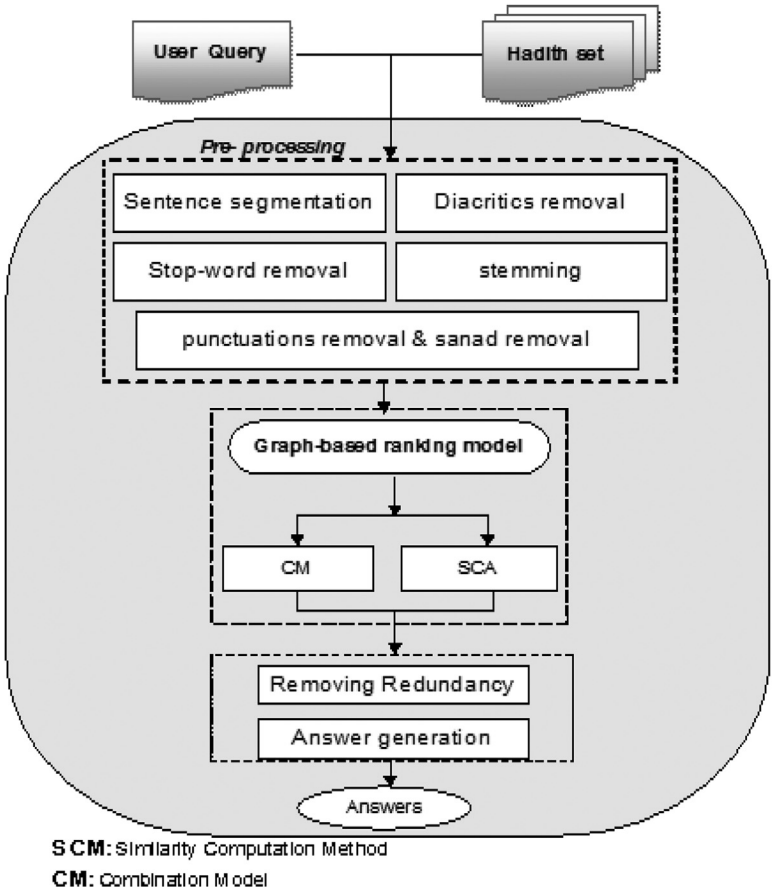

Fig. 1. The Architecture of the ASHLK.

2. Performing the graph-based ranking model. Firstly, it calculates the similarity measure between two nodes (e.g. S2S or Q2S) using the Similarity Computation Approach (SCA). Then, a final score is assigned to each sentence using the Combination method (CM). The idea of combination method is that the score of a sentence is determined as the sum of its similarity to the question and the similarities to the other sentences.
3. Performing the tasks of answer generation. It selects the sentences with the highest score until the length constraint is reached.

3.1. Pre-processing

In this stage, a pre-processing of the sentences and user's query is performed. This step includes linguistic techniques such as segmenting sentences, stop word removal, Diacritics removal, punctuations removal, sanad removal and stemming.

Text segmentation — the segmentation process consists of dividing the texts into meaningful units, in our method text are split into sentences. A sentence ends with a sentence delimiter (".", "?", "!") whereas a paragraph is ended by a new line. Therefore, a paragraph consists of a group of sentences.

Using stop word removal, the words that are very common within a text and are also considered as noisy terms are removed. Obviously, their removal can be effective before the accomplishment of a natural language processing task. Such removal is usually performed

Table 1
Example of stop words.

ان,يعد, ضد, يلي, الي, ي, في, من, حتي, هو, يكون, به, ليس, ذا, منذ, احد, علي, وكان, تلك, كذالك, تي, بين, فيها, لم, عليها, ان, لكن, عن, مساء, لماذا, ليس, ذا, ذي, اما, حين, لا, ليسب, وكانت, اي, ما, عنه, حول, دون, مع, لكنه, لكن, له, هذا, تي, فقط, ثم, هذه, انه, تكون, قد, بين, جدا, لن, نحو, كان, لهم, لان, اليوم, لم, هولاء, فان, فيه, ذالك, لو, عند, اللذين, كل, بد, لدي, وثي, ان, مع, فقد, بل, هو, عنها, منه, بها, في, فهو, تحت, لها, او, ما, لا, الي, مازال, لازال, كما, الا, سفير, بهذا, هي, عبد, لعل, بات

by word filtering with the aid of a list of stop words (Abdi et al., 2015a; Alajmi et al., 2012). In our proposed method a list of stop words has been made manually, it consists of Arabic pronouns, prepositions and conjunctions. Table 1 displays an example of these words:

Punctuation and diacritical signs– the punctuation signs such as (, ¡: ' ÷ × ° > < | \ ¿ '! @ # $% ^ & *) (_ - + = ~ ø ،"), and the diacritical signs such as ( ) are prevalent in Hadith text. These do not have any value in our question-answering system, therefore, we remove all punctuations and diacritics. Sanad removal; in this process, the chain of names of persons who have transmitted Hadith will be removed manually.

Stemming – is a computational procedure that reduces the words with the same root. There are various Arabic root-finding algorithms, e.g. Al-Kabi and Al-Mustafa (2006), Taghva et al. (2005), Al-Serhan et al. (2003), Ghawanmeh et al. (2005), Al-Serhan et al. (2003), Khoja and Garside (1999), Alshalabi (2005), Larkey et al. (2002), Ghawanmeh et al. (2005) and Sonbol et al. (2008). According to the results of research (Al-Kabi et al., 2011; Hadni et al., 2012), among all the various aforementioned methods, the stemmer of Ghwanmeh et al. (2009) achieved the best results. Therefore, based on this result, we used this algorithm to extract stems.

Query Expansion Phase (*QEP*)– is generally used to solve the fundamental problem of term mismatch between sentences and the user's query. Hence, the ASHLK employs the query expansion phase to expand the words in the query and the sentences. The content word expansion method is based on semantic word similarity. The semantic similarity between two words is determined using the following formula:

Dice measure (Vani and Gupta, 2014). The similarity measure between two words w_1 and w_2 based on their synonyms can be defined as follows:

$$\mathrm{Sim}_{\mathrm{Dice}}(w_1, w_2) = \begin{cases} \dfrac{2 \cdot |syns(w_1) \cap syns(w_2)|}{|syns(w_1)| + |syns(w_2)|} & if\ w_1 \neq w_2 \\ 1 & if\ w_1 = w_2 \end{cases} \quad (1)$$

Where *syns*(*w*) is the set of words (synonyms) annotated to *w* based on the lexical database Arabic dictionary, while |*syns*(*w*)| represents the cardinality of the set *syns*(*w*).

3.2. Graph-based ranking model

Given a user's query, the aim of the question answering system is to fetch the answers from a set of sentences. Let *q* denote the query and *S* denotes the set of sentences. A question answering system usually includes two main tasks: (1) the sentences in *S* are scored according to *q;* (2) N sentences with the highest scores are selected to make the answer. In this method, a graph model is created in which each node represents a sentence in *S*. Each edge calculates the similarity measure between the corresponding pair of sentences. We consider two factors when deciding whether sentence S_i is selected as an answer. First, S_i is relevant to the query *q*. Second, S_i is similar to other sentences which have high query-sentence similarity. We denote the sentence-to-sentence (*S2S*) similarity measure as *sim* (S_i, S_j), and the sentence-to- query (*S2q*) similarity measure as *sim* (S_i, *q*). The similarity measure between S2S and the similarity measure between S2Q is calculated using the Similarity Computation Approach (SCA). Then, the score of each sentence is calculated using the Combination Method (CM).

3.2.1. Similarity computation approach (SCA)

This approach calculates the similarity measure between two sentences (*when we compute the similarity between the query and a sentence, the query is also viewed as a sentence*), and then assigns it to the edge between the two current sentences. The overall process of applying the semantic and syntactic information to calculate the similarity measure follows:

1. It takes two sentences S_1 and S_2 as its input.
2. It computes the semantic similarity measure between two sentences (*refer to section "a) Semantic similarity between sentences"*).
3. It calculates the word-order similarity measure between two sentences (*refer to the section "b) Word-order similarity between sentences"*).
4. Finally, it calculates the similarity measure between two sentences (S_1 and S_2) using a linear equation that combines the obtained similarity measures from steps 2 and 3.
5. The final score obtained from the previous step is assigned to the edge between two sentences S_1 and S_2.

a) ***Semantic similarity between sentences***

To calculate the semantic similarity measure between two sentences, we follow the following steps:

1. To create a *Word-Set* (Li et al., 2006). Let $WS = \{W_1, W_2, \ldots, W_N\}$ denote word set, where *N* is the number of distinct words in the word set. However, given two queries *S1* and *S2*, a "word-set" is generated using the following main steps:

   1.1. Two queries are taken as input.

1.2. Using a loop for each word, W_i, from *S1*, main tasks are undertaken, which include:

i. Determining the root of the W_i (denoted by the *RW*).
ii. If the *RW* appears in the *WS*, go to step *1.2* and continue the loop using the next word from *S1*, otherwise, jumping to step iii);
iii. If the *RW* does not appear in the *WS*, then assign the *RW* to the *WS* and then go to the step *1.2* to continue the loop using the next word from *S1*.

We must conduct the same process for *S2*.

2. *To create the semantic-vector.*

The Semantic -Vector is created using the word-set and corresponding query. Each cell of the *SV* corresponds to a word in the word-set, therefore, the dimension of *SV* is equal to the number of words in the word-set.

3. *To weight each cell of the semantic-vector.*

Each cell of the *SV* is weighted using the following steps:

i. If the word, *W*, from the word-set, appears in the *S1*, the weight of the *W* in the *SV* is set to 1. Otherwise, go to the next step;
ii. If the *S1* does not contain the *W*, then compute the similarity score between the *W* and the words from *S1* using the *Dice measure* approach (*refer to Eq. (1)*).
iii. If similarity values exist, then the weight of the *W* in the *SV* is set to the highest similarity value. Otherwise, go to the next step;
iv. If there is no similarity value, then the weight of the *W* in the *SV* is set to 0.

The semantic similarity measure is computed based on the two semantic-vectors. The following equation (Jaccard 1912) is used to calculate the semantic similarity between sentences:

$$\mathrm{Sim}_{\mathrm{Jaccard}}\ (S_1, S_2) = \frac{\sum_{j=1}^{m} (w_{1j} \cdot w_{2j})}{\sum_{j=1}^{m} w_{1j}^2 + \sum_{j=1}^{m} w_{2j}^2 - \sum_{j=1}^{m} (w_{1j} \cdot w_{2j})} \tag{2}$$

Where $S_1 = (w_{11}, w_{12}, \ldots, w_{1m})$ and $S_2 = (w_{21}, w_{22}, \ldots, w_{2m})$ are the semantic vectors of two sentences; w_{1j} is the weight of the j^{th} word in vector S_1, m is the number of words.

b) ***Word-order similarity between sentences***

Since sentences containing the same words but in different orders may result in very different meanings, we use the syntactic-vector approach (Li et al., 2006) to measure the word-order similarity between sentences. The following tasks are performed to measure the word-order similarity between two sentences. For more details, please refer to Abdi et al. (2015c) and Abdi et al. (2017).

1. To create the syntactic-vector.

The syntactic-vector is created using the word-set and corresponding sentence. The dimension of a syntactic-vector is equal to the number of words in the word-set.

2. To weight each cell of the syntactic-vector.

Unlike the semantic-vector, each cell of the syntactic-vector is weighted using a unique index. The unique index can be the index position of the words that appear in the corresponding sentence. However, the weight of each cell in syntactic-vector is determined by the following steps:

2.1. For each word, W_i, from the word set. If the W_i appears in the sentence *S1* the cell in the syntactic-vector is set to the index position of the corresponding word, W_i, in the sentence *S1*. Otherwise, go to the next step;
2.2. If the word W_i does not appear in the sentence *S1*, then compute the similarity score between the W_i and the words from sentence *S1* (*refer to Eq. (1)*).
2.3. If similarity values exist, then the value of the cell is set to the index position of the word from the sentence *S1* with the highest similarity measure.
2.4. If there is not a similar value between the W_i and the words in the sentence *S1*, the weight of the cell in the syntactic-vector is set to 0.

For both sentences, the syntactic-vector is created. Then, the syntactic similarity measure is computed based on the two syntactic-vectors. The following equation is used to calculate the word-order similarity between sentences:

$$Sim_{\mathrm{wordorder}}(S_1, S_2) = 1 - \frac{||\boldsymbol{O}_1 - \boldsymbol{O}_2||}{||\boldsymbol{O}_1 + \boldsymbol{O}_2||} \tag{3}$$

Where $O_1=(d_{11},d_{12},\ldots,d_{1m})$ and $O_2=(d_{21},d_{22},\ldots,d_{2m})$ are the syntactic-vectors of sentences S_1 and S_2, respectively; d_{1j} is the weight of the j^{th} cell in vector O_1.

c) ***Sentence similarity measurement***

The similarity measure between two sentences is calculated using a linear equation that combines the semantic and word-order similarity. The similarity measure is computed as follows:

$$\mathrm{Sim}_{\mathrm{sentences}}(S_1,S_2)=\lambda\cdot\mathrm{sim}_{\mathrm{semantic}}(S_1,S_2)+\ (1-\lambda)\cdot\mathrm{sim}_{\mathrm{wordorder}}(S_1,S_2) \tag{4}$$

Where $0<\lambda<1$ is the weighting parameter, specifying the relative contributions to the overall similarity measure from the semantic and syntactic similarity measures. The larger the λ, the heavier the weight for the semantic similarity. If $\lambda=0.5$ the semantic and syntactic similarity measures are assumed to be equally important.

3.2.2. Combination model (CM)

The main aim of the query-answering system is to choose sentences which are more relevant to the input query. Hence, the sentences which are similar to the input query must obtain high scores. However, a sentence that is similar to the other high scoring sentences in the graph must also get a high score. For example, if a sentence that obtains a high score in measuring the similarity between a sentence and the query is likely to include an answer to the question, then a related sentence, which may not be similar to the input query itself, is also likely to include an answer. This idea is modelled by the following combination model (Losada, 2010; Miao et al., 2010; Zhao et al., 2009):

$$P(s|q)=\alpha\ \times\frac{Sim(s,\ q)}{\sum_{z\in C}Sim(z,\ q)}+(1-\alpha)\ \times\sum_{v\in C}\frac{Sim(s,\ v)}{\sum_{z\in C}Sim(z,\ v)}\times P(v|q) \tag{5}$$

Where, $P(s|q)$ denotes the score of a sentence s given a question q, which is determined as the sum of the similarity between the current sentence and the query, and the similarity between the current sentence and the other sentences in the document set. C contains all sentences.

$0\le\alpha\le1$ is the weighting parameter, it is used to specify the relative contribution of two similarities: the similarity of a sentence to the query and similarity to the other sentences. The bigger the α, the heavier the weight for the *Q-to-S* similarity. If $\alpha=0.5$ the *S2S* similarity measure and the *Q2S* similarity measure are assumed to be equally important. The denominators in both terms are for normalization. The similarity measure between two sentences, $sim(s,\ v)$, and the similarity measure between a sentence and the query, $sim(s,\ q)$ are calculated using the Eq. (4).

The matrix form of formula 1 can be written as

$$\begin{cases}P_{(k+1)}=D^TP_k\\ \\ D=\alpha\ U+(1-\alpha)M\end{cases} \tag{6}$$

Where M, U and D are square matrices. The matrix U indicates the similarity measure between sentences and the M indicates the similarity measure between sentences and the input query. Both matrices (U and M) are normalized to make the sum of each row equal to 1. K represents the k^{th} iteration. The vector $P=[p_1,\ldots,p_N]^{\mathrm{T}}$ is the vector of sentence ranking scores that we are looking for, which corresponds to the stationary distribution of the matrix D. The iteration is guaranteed to converge to a unique stationary distribution given that D is irreducible and aperiodic. For more details please refer to Erkan and Radev (2004) and Otterbacher et al. (2005).

The combination model based on Eq. (6) is performed by doing the following steps.

1. Given two sentences S_i and S_j, the similarity measure between two sentences is calculated using the SCA. Also, given a sentence and the query, the similarity measure between the sentence and the query is calculated using the SCA.
2. Create the square matrix U using $U_{i,j}=Sim(\mathrm{S_i},\ \mathrm{S_j})$. Also, create square matrix M using $M_{ij}=Sim(\mathrm{S_i},\ \mathrm{q})$. M and U should be normalized to make the sum of each row equal to 1.
3. Iterate $P_{(k+1)}=[\alpha\ U+(1-\alpha)M]^TP_k$ until convergence, where α is a parameter between [0,1] and vector p is initialized as the uniform distribution $\left[\frac{1}{N},\frac{1}{N},\ldots,\frac{1}{N}\right]^T$. Usually, the iteration is terminated when $P_{(k+1)}-P_k$ falls below a given threshold value, defined by the user.
4. Let P denote the converged result. Each sentence $S_i\ (1\le i\le N)$ obtains its ranking score p_i.

3.3. Answer generation

Once the sentences are ranked, the simple approach to create the answer is just to select the sentences (answers) with the highest score values until the required answer length is reached. In this case, since various sentences (answers) may include similar content or convey similar information, it is necessary to reduce redundancy. In our method, in order to tackle this problem, we employ two main levels of analysis: first, a scoring level, where every sentence of the hadith corpus is scored using the graph-based model and second, a comparison level, where, before adding the sentences (answers) to the final answer, the sentences assumed to be significant are compared to each other and only those that are not too similar to other answers are

contained in the final answer. To do this, we use the greedy algorithm (Abdi et al., 2015b; Wan et al., 2007; Zhang et al., 2005) to impose a diversity penalty on the sentences in order to remove redundancy. The algorithm includes the following steps.

1. Make two sets, $A_1 = \emptyset$ and $A_2 = \{S_i | i = 1, 2 \ldots, N\}$, and initialize the score of each sentence to its score calculated using Eq. (5).
2. Sort the sentences in A_2 based on their scores in descending order.
3. If S_i is a sentence with a high score in A_2, move S_i from A_2 to A_1, and re-compute the scores of the remaining sentences in A_2 by imposing a redundancy penalty as follows. For each sentence $S_j \in A_2$,

$$Score(s_j) = Score(s_j) - (Sim(s_i, s_j) \times P(s_i|q)) \tag{7}$$

   Where $Sim(s_i, s_j)$ is the similarity measure between two sentences defined in Eq. (3).
4. Go to step 2 and iterate until $A_2 = \emptyset$ or the answer length limitation is satisfied.

Finally, the sentences in the set A_1 are considered as answers for a specific question.

4. Experimental results

We conducted two experiments on the Hadith corpus provided by Sahih al-Bukhari.[1] We describe the details of experiment 1 and experiment 2 in the following sections.

4.1. *Data set (*Hadith corpus*)*

In this section, we explain the data used throughout our evaluation. For the assessment of the performance of the proposed method, we used the datasets provided in Sahih al-Bukhari. Sahih al-Bukhari is a collection of Hadith compiled by Imam Muhammad al-Bukhari. His-collection is recognized by the overwhelming majority of the Muslim world to be the most authentic collection of reports of the sunnah of the Prophet Muhammad (peace and blessings be upon him). It contains over 7500 Hadith in 97 books. In order to evaluate the performance of our method, we conducted two experiments using 4000 Hadiths and corresponding questions, as shown in Table 2. Our experiments were performed on 3825 queries. These users' queries were randomly divided into two separate datasets. In the first experiment, 2678 queries (*training dataset*) were used for parameter tuning (the λ and the α). In the second experiment, the performance of the proposed method is evaluated using the remaining queries (*testing dataset*).

Moreover, in order to evaluate the proposed method, we need a gold standard data, which is a set of all correct results. Based on this dataset, also known as judgment data, we can decide whether the output of the method is correct or not. For this purpose, two experts (having PhD and master's degree, with good reading skills and understanding ability in the Arabic language as well as experience in Islamic Hadith), were asked to collect relevant Hadiths (answers) for a corresponding question, as shown in Table 2. The outputs of our proposed method have been compared with the human reference to calculate the precision, recall and F_measure.

We used several users' queries and corresponding correct answers. We asked human experts to generate questions for the system. There are various types of question. The big sets of questions are those established by a "*wh-question*". The imperative query (e.g., "*show*", "*give*", "*tell*", etc.) are also treated as "*Wh-question*". The process of creating the users' query database contains three main steps: (1) we generate a set of questions according to the Hadith dataset. For this purpose, two human experts were asked to query our Hadith dataset. The Hadith dataset are shown to them in order to generate questions for any kind of Hadith (i.e., *Fear Prayer, Fasting, Dress, Friday Prayer*) they were interested in; (2) this step aims to identify and delete all same and duplicated questions; (3) eventually, the correct answer has been associated for any of the questions.

Inter-raters agreement – we used Cohen's Kappa (Cohen, 1968; Fleiss, 1971) as a measure of agreement between the two raters. The Kappa coefficient for measuring the inter-raters agreement was 0.63. This value indicated that our assessors had a good agreement (Landis and Koch, 1977) to produce gold-standard data.

4.2. *Evaluation metrics*

In order to evaluate and compare the performance of our proposed method, we used the standard measures. We used Precision (P), Recall(R) and F_ measure (Fazlı, 2011; Manning et al., 2008; Perry et al., 1955) to measure the performance of our proposed method. The precision and recall are defined in terms of a set of retrieved Hadith (the list of Hadith produced by ASHLK for a query) and a set of relevant Hadith (the list of all Hadith in human reference).

$$P = \frac{|\{\text{relevant Hadiths }\} \cap \{\text{retrieved Hadiths}\}|}{|\{\text{retrieved Hadiths}\}|} \tag{8}$$

[1] http://sunnah.com/bukhari

Table 2
Description of sample dataset.

Hadith (*Arabic*)		حَدَّثَنَا عَبْدُ اللَّهِ بْنُ يُوسُفَ، قَالَ أَخْبَرَنَا مَالِكٌ، عَنْ هِشَامِ بْنِ عُرْوَةَ، عَنْ أَبِيهِ، عَنْ عَائِشَةَ أُمِّ الْمُؤْمِنِينَ ـ رضى الله عنها ـ أَنَّ الْحَارِثَ بْنَ هِشَامٍ ـ رضى الله عنه ـ سَأَلَ رَسُولَ اللَّهِ صلى الله عليه وسلم فَقَالَ يَا رَسُولَ اللَّهِ كَيْفَ يَأْتِيكَ " أَحْيَانًا يَأْتِينِي مِثْلَ صَلْصَلَةِ الْجَرَسِ ـ وَهُوَ أَشَدُّهُ عَلَىَّ ـ فَيُفْصَمُ عَنِّي الْوَحْىُ فَقَالَ رَسُولُ اللَّهِ صلى الله عليه وسلم وَقَدْ وَعَيْتُ عَنْهُ مَا قَالَ، وَأَحْيَانًا يَتَمَثَّلُ لِيَ الْمَلَكُ رَجُلاً فَيُكَلِّمُنِي فَأَعِي مَا يَقُولُ ". قَالَتْ عَائِشَةُ رضى الله عنها وَلَقَدْ رَأَيْتُهُ يَنْزِلُ عَلَيْهِ الْوَحْىُ فِي الْيَوْمِ الشَّدِيدِ الْبَرْدِ، فَيَفْصِمُ عَنْهُ وَإِنَّ جَبِينَهُ لَيَتَفَصَّدُ عَرَقًا.
Hadith (*English*)		(the mother of the faithful believers) Al-Harith bin Hisham asked Allah's Messenger (ﷺ) "O Allah's Messenger (ﷺ)! How is the Divine Inspiration revealed to you?" Allah's Messenger (ﷺ) replied, "Sometimes it is (revealed) like the ringing of a bell, this form of Inspiration is the hardest of all and then this state passes off after I have grasped what is inspired. Sometimes the Angel comes in the form of a man and talks to me and I grasp whatever he says." 'Aisha added: Verily I saw the Prophet (ﷺ) being inspired divinely on a very cold day and noticed the sweat dropping from his forehead (as the Inspiration was over).
Question	*Arabic*	كَيْفَ يَأْتِيكَ الْوَحْىُ؟
	English	How is the Divine Inspiration revealed to you?

$$R = \frac{|\{\text{relevant Hadiths }\} \cap \{\text{retrieved Hadiths}\}|}{|\{\text{relevant Hadiths}\}|} \tag{9}$$

F_measure is a statistical measure that combines both precision and recall (Manning et al., 2008). It is also defined as the harmonic mean of precision and recall. F_measure is computed as follows:

$$F\_measure = \frac{2 \times P \times R}{P+R} \tag{10}$$

4.3. Experiment 1

This section evaluates the performance of the proposed method. We start with parameter setting followed by comparing the performance of ASHLK with the existing methods. Finally, we present a statistical significance test.

4.3.1. Parameter setting

The proposed method requires two parameters to be determined before use: (1) a weighting parameter (λ) for weighting the significance between semantic information and syntactic information and (2) a trade-off parameter (α), which is a trade-off between the similarity of a sentence to the query and to other sentences in document sets. Both parameters in the current experiment were found using training data, (70%) of Hadiths. We ran our proposed method on the current dataset. We use Eqs. (1)–(10). We evaluate our method for each peer (λ) between 0.1 to 0.9 with a step of 0.1 and (α) between 0 to 1 with a step of 0.1, (e.g. λ = 0.4, α = 0.7). Table 3 presents our experimental results achieved by using various the α and the λ values. We evaluate the results in terms of precision, recall and F_measure.

By analyzing the results, we find that the best performance is achieved by a $\lambda = 0.8$ and $\alpha = 0.7$. This λ and the α produced the scores for three metrics as follows: 0.8175 (precision), 0.6231 (recall), 0.7072 (F_measure). As a result, using the current dataset, we obtain the best result when we use 0.7 as the α value and 0.8 as the λ value. Therefore, we can recommend this the α and the λ values for use on the remaining Hadiths set.

4.3.2. Comparison with related methods

In this section, the performance of ASHLK is compared with other well-known or recently proposed methods. In particular, to evaluate our methods on dataset, we select the following methods: (1) AWNIR (Abderrahim et al., 2013), (2) HMADI (Guirat et al., 2016), (3) VHCIW (Shatnawi et al., 2011), (4) AQAS (Kanaan et al., 2009). These methods have been chosen for comparison because they have achieved the best results. We apply our method to the Hadith set only with the α value 0.7 and λ value 0.8. Table 4 presents the obtained results for the three metrics with the α of 0.7 and the λ of 0.8. Table 4 shows the ASHLK obtained the best result in comparison with the AQAS, which is the best existing approach and has a F_measure of (72.37%). However, due to the result, the ASHLK outperformed the other existing method.

4.3.3. Detailed comparison

We use the relative improvement: ((Our Method - Other methods)/ (Other methods)) $\times$ 100, for comparison between the ASHLK and other approaches. Table 5 displays the results. In Tables 5 "+" indicates that ASHLK method improves the existing approaches. Table 5 presents among the existing approaches the AQAS obtains the best results. However, in comparison with the approach AQAS, ASHLK improves the performance of the AQAS approach as follows: 8.39% (F_measure).

4.3.4. Statistical significance test

We compared the performance of our proposed method with other systems statistically. We used a non-parametric statistical significance test, called Wilcoxon's matched-pairs signed rank-based statistical test, to determine the significance of our results.

Table 3
Performance of the proposed method against various λ and the α values. (Due to space limitations of this paper, sample results are shown).

Trade-off (α)	Weighting (λ)	Precision	Recall	F_measure
α =(0 .. 0.6)	*0.1*	–	–	–
	–	–	–	–
	–	–	–	–
	0.9	–	–	–
α =0.7	0.1	0.4679	0.4293	0.4478
	0.2	0.5018	0.4537	0.4765
	0.3	0.5593	0.4858	0.5200
	0.4	0.6268	0.5154	0.5657
	0.5	0.6922	0.5583	0.6181
	0.6	0.7641	0.5806	0.6598
	0.7	0.7958	0.5990	0.6835
	0.8	0.8175	0.6231	0.7072
	0.9	0.7835	0.5794	0.6662
α =(0.8 .. 1)	*0.1*	–	–	–
	–	–	–	–
	–	–	–	–
	0.9	–	–	–

Table 4
Performance comparison between ASHLK and other methods on Hadith set.

Precision, Recall and F_measure values of the methods			
System	Precision	Recall	F_measure
ASHLK	0.8347	0.6387	0.7237
AWNIR	0.7315	0.5511	0.6286
VHCIW	0.6209	0.5357	0.5752
AQAS	0.7843	0.5812	0.6676
HMADI	0.7200	0.5207	0.6043

The statistical significance test for independent samples has been conducted at the 5% significance level. We created five groups, corresponding to the five systems: 1. ASHLK, 2. AWNIR, 3. VHCIW, 4. AQAS, 5. HMADI for data set. Two groups are compared at a time one corresponding to ASHLK system and the other corresponding to some other system considered in this paper. Each group consists of the precision and F_measure for the dataset produced by each corresponding system.

The median values and standard deviation (Stdv.) of precision and F_measure of each system for the data set are presented in Tables 6. As shown in Table 6 the median values of precision and F_measure for ASHLK system on data set are better than that for the other systems. To establish that this goodness is statistically significant, Table 7 reports the P-values produced by Wilcoxon's matched-pairs signed-rank test for comparison of two groups (one group corresponding to establish and another group corresponding to some other system) at a time. As a null hypothesis, it is assumed that there are no significant differences between the median values of the two groups. Whereas, the alternative hypothesis is that there is significant difference in the median values of the two groups. It is clear from Table 7 that *P*-values are much less than 0.05 (5% significance level). For example, the Wilcoxon's matched-pairs signed-rank test between the systems establishes and AQAS for data set provides a *P*-value of 0.028 (*F_measure*), which is very small. This is strong evidence against the null hypothesis, indicating that the better median values of the performance metrics produced by establishing ASHLK is statistically significant and has not occurred by chance. Similar results are obtained for all other systems compared to ASHLK system, establishing the significant superiority of the proposed system. From the statistical results, we observe that our ASHLK system significantly outperforms the other systems.

4.4. Experiment 2

This section aims to examine the effectiveness of N-gram measure (Ngm), Jaccard measure (Jm), cosine similarity measure (Csm), Overlap similarity coefficients (Osc) and Similarity word order (Swo) on ASHLK.

Table 5
Performance evaluation compared between the ASHLK and other methods.

ASHLK improvement (%)				
Metrics	AWNIR	VHCIW	AQAS	HMADI
F_measure	+ 15.12	+ 25.81	+ 8.39	+ 19.74

Table 6
Median values and standard deviation of systems on dataset.

System	Precision		F_measure	
	Median	Stdv.	Median	Stdv.
ASHLK	0.7262	4.52E-02	0.6660	5.06E-02
AQAS	0.7256	2.44E-01	0.6710	7.72E-02
AWNIR	0.6292	1.09E-01	0.5438	1.08E-01
HMADI	0.6293	1.20E-01	0.5427	1.14E-01
VHCIW	0.5812	1.07E-01	0.4978	1.06E-01

4.4.1. *Influence of Ngm, Jm, Csm, Osc and Swo on Answer generation*

Csm — the following equation (Alguliev et al., 2011) is used to calculate the semantic similarity between sentences:

$$Sim_{semantic}\ (S_1, S_2) = \frac{\sum_{j=1}^{m} (w_{1j} \times w_{2j})}{\sqrt{\sum_{j=1}^{m} w_{1j}^2} \times \sqrt{\sum_{j=1}^{m} w_{2j}^2}} \tag{11}$$

Where $S_1 = (w_{11}, w_{12}, \ldots, w_{1m})$ and $S_2 = (w_{21}, w_{22}, \ldots, w_{2m})$ are the semantic vectors of sentences S_1 and S_2, respectively; w_{pj} is the weight of the j^{th} word in vector S_p, m is the number of words.

Ngm — we use the N-gram (Lin, 2004) to measure the word-order similarity between sentences. Ngm takes word order information into account when computing sentence similarity. A Ngm is a subsequence of n words from a given sentence. We split sentences into n-grams ($1 \le n \le 3$) and computed the number of co-occurrence n-grams in S_i and S_j.

$$\mathrm{sim}_{\mathrm{wordorder}}(\mathrm{S_i}, \mathrm{S_j}) = \sum_{n\text{-}gram} \frac{2 \times |S_I \cap S_j|}{|S_i| + |S_j|} \tag{12}$$

Where, $|S_i|$ and $|S_j|$ are the length of sentence S_i and sentence S_j, respectively.

Osc — it considers the set of words occurring in both sentences. Given two sentences, let $S_1 = \{W_1, W_2, \cdots \mathrm{W_N}\}$ be a sentence of summary text, where N is the number of words in the sentence $\mathrm{S_1}$, $S_2 = \{W_1, W_2, \cdots \mathrm{W_M}\}$ is a sentence of the original text, where M is the number of words in the sentence $\mathrm{S_2}$. However, for each word from sentence $\mathrm{S_1}$, the same word or the synonym word must be restated in sentence $\mathrm{S_2}$. Hence, the following statement can be to calculate the similarity between two sentences.

$$Sim_{overlap}(S_1,\ S_2) = \frac{|syns(S_1) \cap\ syns(S_2)|}{min(|syns(S_1)|, |syns(S_2)|)} \tag{13}$$

Where $syns(d)$ is the set of words and their synonyms in the sentence d.

Swo and ***Jm*** — we used Eqs. (2) and (3) to calculate Jaccard measure (Jm) and Similarity word order (Swo), respectively.

In this experiment, our aim is to examine the efficiency of the *Ngm, Jm, Csm, Osc* and *Swo* on our proposed method. We set the α value 0.7 (parameter in Eq. (5)) and λ value 0.8 (parameter in Eq. (4)). The results of our experiment are reported in Table 8 where various types of sentence similarity are tested.

As seen from Table 8 the combined measure (Swo + Jm) out-performs the (Swo + Osc) measure, (Swo + Csm) measure, (Ngm + Csm) measure, (Ngm + Osc) measure and (Ngm + Jm) measure. From this table, it can be seen that the performance of the (Swo + Jm) measure is better than measures in terms of the results of F_measure. Due to the results, we used the combined measure (Swo + Jm) to calculate the similarity measure between two sentences in our proposed method.

4.5. *Discussion*

From Tables 5–7, we obtained the following observations. The ASHLK outperforms other methods and obtained good performance. This is due to the facts that, *1)* It is able to identify the synonymous words among all sentences using the QEP approach. *2)* Given two sentences (i.e., S_1: *Father likes Son, الأب يحب الابن; S_2: Son likes Father, الابن يحب الأب*), unlike another method, our method is able to distinguish the meaning of two sentences by using the combination of semantic and syntactic information. It integrates the semantic and syntactic information to compute (*S to S*) and (*Q to S*) similarity measures.

Table 7
P-values produced by Wilcoxon's matched-pairs signed-rank test by comparing ASHLK with other systems.

Systems	AQAS	AWNIR	HMADI	VHCIW
Data set	*Comparing medians of Precision metric of ASHLK with other systems*			
	0.033	0.001	0.005	0.040
	Comparing medians of F_measure metric of ASHLK with other systems			
	0.028	0.016	0.004	0.005

Table 8
Performance of the ASHLK against various tests (*Ngm, Swo, Jm, Csm, Osc*).

Semantic similarity between sentences			
Word order similarity	Csm	Osc	Jm
Swo	$F = 0.6654$	$F = 0.5844$	$F = 0.7237$
	$P = 0.7807$	$P = 0.6847$	$P = 0.8347$
	$R = 0.5797$	$R = 0.5097$	$R = 0.6387$
Ngm	$F = 0.4677$	$F = 0.4225$	$F = 0.5088$
	$P = 0.5047$	$P = 0.4827$	$P = 0.5537$
	$R = 0.4357$	$R = 0.3757$	$R = 0.4707$

As a result, the results show that the combination of the semantic word similarity, syntactic and semantic, information can improve the performance.

The system also fails to answer some of the queries. We explain the common errors we encountered during our experiment: (1) Arabic language has more complex morphological, grammatical and semantic structures that make existing NLP techniques inadequate for the Arabic text. (2) The lack of tools and software development environments that process the Arabic script. (3) The lack of lexicons, and machine-readable dictionaries, which are essential to advance research. (4) The limitations of the lexical database that can be a lack of information, concepts and some semantic relations between synsets.

5. Conclusion and future work

In this paper, we present a Question-Answering system for Arabic Language using Hadith datasets. Our proposed method in this work not only combines the semantic and syntactic information to capture the meaning in the comparison between two sentences or query-to-sentence but also considers query expansion to extract the more query relevant sentences from the Hadith datasets. However, this particular method can improve the performance because it is able to avoid extracting a sentence whose similarity with the query is high but whose meaning is different. Furthermore, to reduce redundancy in answer text, this method uses the greedy algorithm to impose diversity penalty on the sentences. In addition, the proposed method expands the words in both the query and the sentences to tackle the problem of information limit. It bridges the lexical gaps for semantically similar contexts that are expressed using different wording.

The evaluation of ASHLK is conducted over Hadith datasets. The proposed method is very easy to follow and requires minimal text processing cost. Initially, parameters of ASHLK are optimized over the training dataset. Later we used the remaining datasets (testing dataset) to assess the performance of ASHLK using the recall, precision and F_measure. The ASHLK is compared with the well-known existing methods. The experimental results display that the performance of the proposed method is very competitive when compared with other methods.

This paper presents the following suggestions for future work. The method uses a lexical database as the main semantic knowledge base to calculate the semantic similarity between words. The comprehensiveness of the lexical database is determined by the proportion of words in the text that are covered by its knowledge base. However, the main criticism of the lexical database concerns its limited word coverage to calculate the semantic similarity between words. Obviously, this disadvantage has a negative effect on the performance of our proposed method. To tackle this problem, in addition to the lexical database, other knowledge resources, such as large corpus can be used. Furthermore, the experiment revealed that there are some common failures that we encountered during our evaluation. We would like to consider them in order to improve the performance of our proposed method.

Acknowledgements

This work is supported by Universiti Teknologi Malaysia (UTM) under research university grant no. 17H62, 03G91, and 04G48. The authors would like to express their deepest gratitude to ASEAN-India Collaborative R&D Program, Cyber Physical System research group, as well as School of Computing, Faculty of Engineering for their continuous support in making this research a success.

References

Abderrahim, M.A., Abderrahim, M.E.A., Chikh, M.A., 2013. Using Arabic wordnet for semantic indexation in information retrieval system. arXiv:1306.2499.
Abdi, A., Idris, N., Alguliev, R.M., Aliguliyev, R.M., 2015. Automatic summarization assessment through a combination of semantic and syntactic information for intelligent educational systems. Inf. Process. Manag. 51 (4), 340–358.
Abdi, A., Idris, N., Alguliyev, R., Aliguliyev, R., 2015. Query-based multi-documents summarization using linguistic knowledge and content word expansion. Soft. Comput. 21 (7), 1–17.
Abdi, A., Idris, N., Alguliyev, R.M., Aliguliyev, R.M., 2015. PDLK: Plagiarism detection using linguistic knowledge. Expert Syst. Appl. 42 (22), 8936–8946.
Abdi, A., Shamsuddin, S.M., Idris, N., Alguliyev, R.M., Aliguliyev, R.M., 2017. A linguistic treatment for automatic external plagiarism detection. Knowl. Based Syst. 135, 135–146.
Al-Kabi, M., Al-Mustafa, R., 2006. Arabic root based stemmer. In: Proceedings of the International Arab Conference on Information Technology, Jordan.

Al-Kabi, M.N., Al-Radaideh, Q.A., Akkawi, K.W., 2011. Benchmarking and assessing the performance of Arabic stemmers. J. Inf. Sci. 37 (2), 111–119.
Al-Serhan, H.M., Al Shalabi, R., & Kannan, G. (2003). New approach for extracting Arabic roots.
Alajmi, A., Saad, E. M., Darwish, R.R., 2012. Toward an ARABIC stop-words list generation. International Journal of Computer Applications. 46, 8–13.
Alguliev, R.M., Aliguliyev, R.M., Mehdiyev, C.A., 2011. Sentence selection for generic document summarization using an adaptive differential evolution algorithm. Swarm Evol. Comput. 1 (4), 213–222.
Alshalabi, R., 2005. Pattern-based stemmer for finding Arabic roots. Inf. Technol. J. 4 (1), 38–43.
Atwell, E., Brierley, C., Dukes, K., Sawalha, M., Sharaf, A.-B., 2011. An Artificial Intelligence approach to Arabic and Islamic content on the internet. In Proceedings of NITS 3rd National Information Technology Symposium, pp. 1–8. Leeds.
Batyrzhan, M., Kulzhanova, B., Abzhalov, S., Mukhitdinov, R., 2014. Significance of the Hadith of the Prophet Muhammad in Kazakh Proverbs and Sayings. Procedia-Soc. Behav. Sci. 116, 4899–4904.
BEKHTI, S., AL-HARBI, M., 2013. AQuASys: a question-answering system for arabic, wseas international conference. In: Proceedings of the Recent Advances in Computer Engineering Series. WSEAS.
Benajiba, Y., Rosso, P., Lyhyaoui, A., 2007. Implementation of the Arabiqa question answering system's components. In: Proceedings of the Workshop on Arabic Natural Language Processing, 2nd Information Communication Technologies Int. Symposium, ICTIS-2007, Fez, Morroco, April, pp. 3–5.
Benajiba, Y., Rosso, P., Soriano, J.M.G., 2007. Adapting the JIRS passage retrieval system to the Arabic language. In: Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics. Springer, pp. 530–541.
Brini, W., Ellouze, M., Mesfar, S., Belguith, L.H., 2009. An Arabic question-answering system for factoid questions. In: Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering. IEEE, pp. 1–7. NLP-KE 2009.
Cohen, J., 1968. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. Psychol. Bull. 70 (4), 213.
Erkan, G., Radev, D.R., 2004. Lexrank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. 22, 457–479.
Fazlı, C., 2011. Text summarization using latent semantic analysis. Middle East Technical University.
Fleiss, J.L., 1971. Measuring nominal scale agreement among many raters. Psychol. Bull. 76 (5), 378.
Friburger, N., Maurel, D., Giacometti, A., 2002. Textual similarity based on proper names. In: Proceedings of the Workshop Mathematical/Formal Methods in Information Retrieval, pp. 155–167.
Ghawanmeh, S., Al-Shalabi, R., Kanaan, G., Khanfar, K., Rabab'ah, S., 2005. An algorithm for extracting the root for the Arabic language. In: Proceedings of the 5th International Business Information Management Association Conference, Cairo, Egypt.
Ghwanmeh, S., Kanaan, G., Al-Shalabi, R., Rabab'ah, S., 2009. Enhanced algorithm for extracting the root of Arabic words. In: Proceedings of the Sixth International Conference on Computer Graphics, Imaging and Visualization. IEEE, pp. 388–391.
Guirat, S.B., Bounhas, I., Slimani, Y., 2016. A hybrid model for Arabic document indexing. In: Proceedings of the 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). IEEE, pp. 109–114.
Hadni, M., Lachkar, A., Ouatik, S.A., 2012. A new and efficient stemming technique for Arabic Text Categorization. In: Proceedings of the International Conference on Multimedia Computing and Systems. IEEE, pp. 791–796.
Hammo, B., Abu-Salem, H., Lytinen, S., 2002. QARAB: a question answering system to support the arabic language. In: Proceedings of the ACL-02 Workshop on Computational Approaches to Semitic Languages. Association for Computational Linguistics, pp. 1–11.
Hanum, H.M., Bakar, Z.A., Rahman, N.A., Rosli, M.M., Musa, N., 2014. Using topic analysis for Querying Halal Information on Malay documents. Procedia-Soc. Behav. Sci. 121, 214–222.
Jaccard, P., 1912. The distribution of the flora in the alpine zone. New phytologist 11, 37–50.
Kanaan, G., Hammouri, A., Al-Shalabi, R., Swalha, M., 2009. A new question answering system for the Arabic language. Am. J. Appl. Sci. 6 (4), 797.
Khoja, S., Garside, R., 1999. Stemming Arabic Text. Computing Department, Lancaster University, Lancaster, UK.
Landis, J.R., Koch, G.G., 1977. The measurement of observer agreement for categorical data. Biometrics 33 (1), 159–174.
Larkey, L.S., Ballesteros, L., Connell, M.E., 2002. Improving stemming for Arabic information retrieval: light stemming and co-occurrence analysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 275–282.
Li, Y., McLean, D., Bandar, Z.A., O'shea, J.D., Crockett, K., 2006. Sentence similarity based on semantic nets and corpus statistics. IEEE Trans. Knowl. Data Eng. 18 (8), 1138–1150.
Lin, 2004. Rouge: a package for automatic evaluation of summaries. In: Proceedings of the ACL-04 Workshop on Text Summarization Branches Out, pp. 74–81.
Losada, D.E., 2010. Statistical query expansion for sentence retrieval and its effects on weak and strong queries. Inf. Retr. Boston 13 (5), 485–506.
Lu, W., Cheng, J., Yang, Q., 2012. Question answering system based on web. In: Proceedings of the 2012 Fifth International Conference on Intelligent Computation Technology and Automation. IEEE Computer Society, pp. 573–576.
Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to Information Retrieval. Cambridge University Press, Cambridge.
Mazauric, C., Rothiot, J.-P., 2007. Frontières Et éspaces frontaliers Du Leman à la Meuse: Recompositions et Echanges De 1789 à 1814. Presses Universitaires de Nancy.
Miao, Y., Su, X., Li, C., 2010. Improving question answering based on query expansion with Wikipedia. In: Proceedings of the 22nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI). IEEE, pp. 233–240.
Otterbacher, J., Erkan, G., Radev, D.R., 2005. Using random walks for question-focused sentence retrieval. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 915–922.
Papadimitriou, C.H., Tamaki, H., Raghavan, P., Vempala, S., 1998. Latent semantic indexing: a probabilistic analysis. In: Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. ACM, pp. 159–168.
Pavlić, M., Han, Z.D., Jakupović, A., 2015. Question answering with a conceptual framework for knowledge-based system development “Node of Knowledge”. Expert Syst. Appl. 42 (12), 5264–5286.
Peral, J., Ferrández, A., De Gregorio, E., Trujillo, J., Maté, A., Ferrández, L.J., 2014. Enrichment of the phenotypic and genotypic data warehouse analysis using question answering systems to facilitate the decision making process in cereal breeding programs. Ecol. Inform. 26, 203–216.
Perry, J.W., Kent, A., Berry, M.M., 1955. Machine literature searching x. machine language; factors underlying its design and development. Am. Doc. 6 (4), 242–254.
Shatnawi, M.Q., Abuein, Q.Q., Darwish, O., 2011. Verification hadith correctness in islamic web pages using information retrieval techniques. Inf. Commun. Syst. 164.
Sonbol, R., Ghneim, N., Desouki, M.S., 2008. Arabic morphological analysis: a new approach. In: Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications. IEEE, pp. 1–6.
Taghva, K., Elkhoury, R., Coombs, J., 2005. Arabic stemming without a root dictionary. In: Proceedings of the International Conference on Information Technology: Coding and Computing. IEEE, pp. 152–157.
Tata, S., Patel, J.M., 2007. Estimating the selectivity of tf-idf based cosine similarity predicates. ACM SIGMOD Rec. 36 (2), 7–12.
Vani, K., Gupta, D., 2014. Using K-means cluster based techniques in external plagiarism detection. In: Proceedings of the International Conference on Contemporary Computing and Informatics (IC3I). IEEE, pp. 1268–1273.
Wan, X., Yang, J., Xiao, J., 2007. Manifold-Ranking based topic-focused multi-document summarization. In: Proceedings of the IJCAI, pp. 2903–2908.
Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., Chen, Z., Ma, W.-Y., 2005. Improving web search results using affinity graph. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp. 504–511.
Zhao, L., Wu, L., Huang, X., 2009. Using query expansion in graph-based approach for query-focused multi-document summarization. Inf Process Manag. 45 (1), 35–41.