Soft Comput (2017) 21:1785–1801
DOI 10.1007/s00500-015-1881-4

METHODOLOGIES AND APPLICATION

Query-based multi-documents summarization using linguistic knowledge and content word expansion

Asad Abdi[1] · **Norisma Idris**[1] · **Rasim M. Alguliyev**[2] · **Ramiz M. Aliguliyev**[2]

Published online: 23 September 2015

Abstract In this paper, a query-based summarization method, which uses a combination of semantic relations between words and their syntactic composition, to extract meaningful sentences from document sets is introduced. The problem with current statistical methods is that they fail to capture the meaning when comparing a sentence and a user query; hence there is often a conflict between the extracted sentences and users' requirements. However, this particular method can improve the quality of document summaries because it is able to avoid extracting a sentence whose similarity with the query is high but whose meaning is different. The method is executed by computing the semantic and syntactic similarity of the sentence-to-sentence and sentence-to-query. To reduce redundancy in summary, 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 experimental results display that the proposed method is able to improve performance compared with the participating systems in DUC 2006. The experimental results also showed that the proposed method demonstrates better performance as compared to other existing techniques on DUC 2005 and DUC 2006 datasets.

Keywords Query-based multi-document summarization · Graph-based sentence ranking · Query expansion · Extractive summarization

Communicated by V. Loia.

✉ Asad Abdi
asadabdi55@gmail.com

Norisma Idris
norisma@um.edu.my

Rasim M. Alguliyev
r.alguliev@gmail.com

Ramiz M. Aliguliyev
r.aliguliyev@gmail.com

[1] Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia

[2] Institute of Information Technology, Azerbaijan National Academy of Sciences, 9, B. Vahabzade Street, AZ 1141 Baku, Azerbaijan

1 Introduction

Due to the huge amount of information available, new technology that can process this information is required by users. Document summarization can be an essential technology for tackling this problem. Document summarization aims to produce a short version of a source text that provides informative information for users (Abdi et al. 2015; Aliguliyev 2009; Idris et al. 2009; Saggion and Poibeau 2013). It can be based on a single document or multiple documents (Lee et al. 2009; Mendoza et al. 2014). In multi-document summarization, several documents on a single topic are employed to produce a summary text. However, in single-document summarization, only one document is employed to generate a summary text. Summaries can be either generic summaries or query-based summaries (Abdi and Idris 2014; Lee et al. 2009; Sarker et al. 2013). In generic text summarization, the summary is made about the whole document whereas in query-based text summarization, the generated summary is about the query asked. Query-based summarization is a specific kind of document summarization. Given a user query, the task is to produce from the document a summary which can provide informative information corresponding to the

user's information needs (Badrinath et al. 2011; Mendoza et al. 2014).

With the rapid advances in computer technology, researchers have developed several methods and tools for query-based summarization (Kanejiya et al. 2003; Pérez et al. 2005; Wiemer-Hastings and Wiemer 2000; Wiemer-Hastings and Zipitria 2001). Due to the progress in other areas, such as e-learning (EL), natural language processing (NLP), automatic evaluation of summaries, document categorization, automatic question answering (Q/A) and information extraction (IE), automatic summarization based on user query has been made possible.

In the context of text relevance, linguistic knowledge, such as semantic relations between words and their syntactic composition, plays a key role in sentence understanding. This is particularly important in comparing two sentences where a single word token is used as the basic lexical unit for comparison.

Syntactic information, such as word order, can provide useful information to distinguish the meaning of two sentences, when two sentences share the similar bag of words. For example, "Alex calls John" and "John calls Alex" will be judged as similar sentences because they have the same surface text. However, these sentences convey different meanings.

Most of the existing query-based summarization methods do not include syntactic information in calculating the similarity measure between sentence-to-sentence (S2S) and query-to-sentence (Q2S). They fail to capture the meaning when comparing two sentences or the query and a sentence, hence sometimes the sentences in a summary text conflict with the information expressed in a user's query. However, for the correct computing of similarity measure and to identify sentences more relevant to the user's query, methods should take into account both semantic and syntactic information (Kanejiya et al. 2003; Pérez et al. 2005; Wiemer-Hastings and Wiemer 2000; Wiemer-Hastings and Zipitria 2001).

On other hand, when comparing two sentences (when we compare the query and a sentence, the query is also viewed as a sentence), two sentences are considered to be similar or relevant if most of the words are the same or if they are a paraphrase of each other. However, it is not always the case that sentences with a similar meaning share many similar words. Hence, semantic information such as semantic similarity between words and their synonyms can provide useful information when two sentences have a similar meaning, but used different words. This is because people can express the same meaning using various sentences in terms of word content. However, the more relevant sentence may be represented by similar words, rather than the original words expressed in the user's query. Furthermore, a query contains very few words. Thus, the main problem is to use this little information to find salient sentences which answer the question in the user's query. Therefore, due to the information limit that is expressed by a query (Badrinath et al. 2011; Lloret et al. 2011; Lu et al. 2012; Zhao et al. 2009), and to enhance the possibility of finding related sentences in the document, we perform content word expansion as part of our proposed method using the semantic similarity between words and their synonym.

The present work introduces a method to produce user-focused summaries without any complicated processing steps. The aim is to generate summaries that provide useful information for user needs.

To determine whether a sentence must be contained correctly in the summary text, we obtain two types of features for each sentence: the strength of connection with the user's query and the strength of general connection, The first is done by computing the similarity measure between the current sentence and the user's query using a combination of semantic and syntactic information and the second by calculating the similarity measure between the current sentence and other sentences in the text document using a combination of semantic and syntactic information. This method also expands the user's question and sentences to find more matching sentences from a text document. Moreover, to more effectively remove redundancy and increase the information diversity in the summary, we use a greedy algorithm presented in the last step (4.) of proposed method. The proposed method is called query-based summarization using linguistic knowledge (QSLK), since the summaries are produced using semantic information obtained from a lexical database and syntactic information is provided by analyzing the structure of the sentence.

The structure of this paper is as follows. Section 2 provides a short overview of the previous methods that have been used to produce summaries. Section 3 introduces the proposed method. Section 4 presents the summary generation. Section 5 discusses the performance analysis and presents the results of the analysis. Finally, in Sect. 6, we summarize the works discussed and the progress of the project.

2 Brief review of literature

In recent years, query-based summarization has become subject to be investigated in NLP. Several studies have shown that the computer can be used for generating summaries based on users' queries. However, computer models of the methods employed by users to provide salient information according to their queries. To implement these models is more difficult, since they have to identify important information which is related to a particular query in a text document (i.e., sentences/paragraph). Despite the difficulty in implementing these models, researchers have recently proposed several

methods for query-based summarization. In this section, we explain some of these methods.

Pandit and Potey (2013) proposed a graph-based method to summarize documents based on user's query. The proposed method includes two stages, the offline and online stages. In the offline stage, pre-processing tasks are performed. Give a document set which needs to be summarized, first, all stop words are removed from the sentences. Then, the documents are decomposed into a set of paragraphs. Each node of the graph represents a paragraph. An edge is added between two nodes if they are semantically related. If two nodes share common words, they are related. The similarity score between two nodes is calculated using the *TF-IDF* method. The similarity score is considered as the weight of the edge between two nodes. Finally, the nodes of the graph are clustered using the AHA approach (Davidson and Ravi 2005) and nearest neighbor algorithm (Shekhar and Xiong 2008) to reduce the processing time during the online stage. At the online level, first, the similarity measure between each cluster and query is calculated using the okapi equation which is based on *TF-IDF* (Varadarajan and Hristidis 2006). Second, minimal clusters are identified. Minimal clusters are the clusters which are related to the input query and the weight of the edge between a cluster and the input query is non-zero. These minimal clusters are shown in the result. For this purpose, the top-n clusters with the highest weight in relation to the input query are displayed.

He et al. (2012) proposed a method to generate summaries from multiple documents based on a user's question. The method considers various factors to determine the score for each sentence. These factors comprise the segmentation results weight (SRW), characteristics of sentence structure (CSS), length of sentence (LS) and the mutual information (MI) of the user's query. The SRW factor considers part of speech and term frequency to calculate the sentence weight. It assumes that sentences containing verbs and nouns are more important than sentences that contain adverbs and others parts of speech.

The CSS factor includes the sentence location and the kind of sentence (e.g., declarative sentence, interrogative sentence and exclamatory sentence). The location of a sentence indicates that a sentence is able to represent the main idea of the document, if it appears in the first paragraph or in the end paragraph of the document. Thus, if a sentence is from the first paragraph its position weight is equal to 1; the others $0 <$ sentence weight < 1. Moreover, this factor also considers the kind of sentence. A declarative sentence includes important information from the document in comparison with others such as the interrogative sentence and exclamatory sentence. Thus, the coefficient of the declarative sentence is equal to 1; the others $0 <$ coefficient of the sentence < 1. The LS factor takes the average length of sentence into account for sentence weight. Usually, a long sentence includes redundant information and a short sentence cannot represent the main idea of the document. Hence, the average length of the sentence is used to compute the sentence weight. The MI factor is used to calculate similarity between the query and a sentence. The similarity is determined based on the number of matching words they share. Finally, the results for each factor are summed up and then assigned to the sentence as sentence weight. The rough summarization step in the proposed method orders the sentences based on their weight from high to low. Afterwards, some of the sentences are selected according to the specified compression ratio.

Ouyang et al. (2011) proposed a method based on regression models to produce a single summary from a document set based on user's query. The proposed method assigns a score to each sentence using sentence features and a composite function. It uses seven features for sentence scoring. These features contain three query-dependent features and four query-independent features. Query-dependent features include the word matching feature, semantic matching feature and named entity matching feature. The query-independent features include the word TF-IDF feature, stop-word penalty feature, and sentence position feature.

The word matching feature indicates the number of words that the user's query and a document sentence share. The semantic matching feature uses a lexical dictionary to expand the query. The named entity matching feature contains the number of common named entities in a document sentence and the user's query. The word TF-IDF feature uses the TF-IDF approach to determine important words in a text document. The sentence position feature assumes the sentence at the beginning of the document or the paragraph presents the main idea of the document. To rank each sentence, a composite function based on the SVR approach (Basak et al. 2007) is used to calculate the sentence score using this feature set.

Finally, the method uses the maximum marginal relevance (MMR) approach to select sentences. First, all sentences are ordered according to their scores from high to low. Then, the sentences of the summary text are chosen as follows. The method compares the current sentence with other sentences before selection. If the similarity between the current sentence and the other sentence does not exceed the N value (N defined by user), the sentence is considered a summary sentence.

Hu et al. (2010) proposed a method for query-based summarization. The method considers two important characteristics in the scoring process. The characteristics include the query-dependent characteristic, and the query-independent characteristic. In the query-dependent characteristic, the similarity score between the query and each sentence is computed using the normalized cosine relevance value between the query and the sentence. In the query-independent characteristic, the similarity measure between two sentences is calculated.

The method first uses two base rankers to identify the important sentences. Then, a score is assigned to each sentence using the rank aggregation algorithm which merges these two ranking results. Finally, the highest ranked sentences are selected. The MMR algorithm (Carbonell and Goldstein 1998) is used to prevent the redundant sentence appearing in the summary text.

Summing up, in this section, different kinds of method have been presented to produce summaries from multiple documents based on a user's question. Text summarization systems usually provide the user with a generic summary that highlights the most important information in a text whereas in query-based text summarization, the generated summary contains more information related to the query. In other words, query-based multi-document summarization produces a single coherent summary of a set of related source documents, which answers the need for information expressed in a given query. As compared to single-document summarization, the challenge for query-specific multi-document summarization is that the created summary is not only expected to contain the important information in the whole document set, but also make sure the summary is biased to the given query as much as possible.

The main problem with the most existing system usually is that they fail to capture the meaning when comparing a sentence-to-sentence and sentence-to-query; hence there is a conflict between extracted sentences and user's need. For instance, the current systems make no difference between two sentences, 'tiger attacked someone' and 'someone attacked tiger'. Although, it is quite clear that the two sentences have quite different meaning. Considering relationship between the words (syntactic composition) can help in identifying query relevant sentences. In this paper, we propose a method that is able to capture the meaning in comparison between S2S or Q2S, when two sentences or a query and a sentence have same surface text (the words are the same) or they are a paraphrase of each other. The proposed method is able to avoid selecting the sentence whose similarity with query is high but its meaning is different. It is executed by computing the semantic and syntactic similarity of the sentence-to-sentence and query-to-sentence.

3 Proposed method: QSLK

In this section, a method for query-based multi-document summarization is presented. The overview of our proposed method is shown in Fig. 1. The method includes the following steps:

1. Perform pre-processing tasks on the document set and the input query. This aims to prepare the documents and the input query for the subsequent steps.
2. Apply the graph-based ranking model. First, the similarity measure between two nodes (e.g., S2S or Q2S) is calculated using the statistical similarity computation method (SSCM). Then, a final score is assigned to each sentence using the combination model (CM). The idea of the combination model is that the score of a sentence is determined as the sum of its similarity to the question and its similarity to the other sentences in the document.
3. Perform the tasks of summary generation. This involves selecting the number of sentences with a high score, until the length constraint is reached (a 250-word summary). At the same time, the greedy algorithm (Badrinath et al. 2011; Wan et al. 2007) is used to produce a summary with high coverage and less redundancy.

We describe each of the aforementioned steps in the subsequent sections.

3.1 Pre-processing

The main task of this step is to prepare the query and the document sentences for further processing. The step consists of several functions, such as sentence segmentation, tokenization and stop word removal. This step splits the document text into individual sentences. Then, it removes all stop words from both sentences and the input query.

3.2 Graph-based ranking model

This section describes how the graph-based ranking model is applied. This model calculates the score of each sentence. For this purpose, first, two similarity measures are computed: the similarity measure of a sentence to other sentences and the similarity measure of a sentence to the input query. Then, a score is assigned to each sentence by the CM.

The proposed method, QSLK, relies on the two-dimension graph model. It is created as follows. The input query and the document sentences are considered as nodes on the graph. For each node, two kinds of edge are used: (1) the sentence-to-sentence similarity measure: an edge between two sentences; (2) the sentence-to-query similarity measure: an edge between a sentence and the query. A weight is assigned to each edge in the graph to assess the correlation between the two nodes connected by the current edge. The weight with an edge is the similarity measure between two nodes. The score of a node, sentence, on the graph is determined by the relevance of the current sentence to both the input query and other sentences in the graph. Figure 2 shows a sample of a document graph that was constructed using the query and the document sentences.

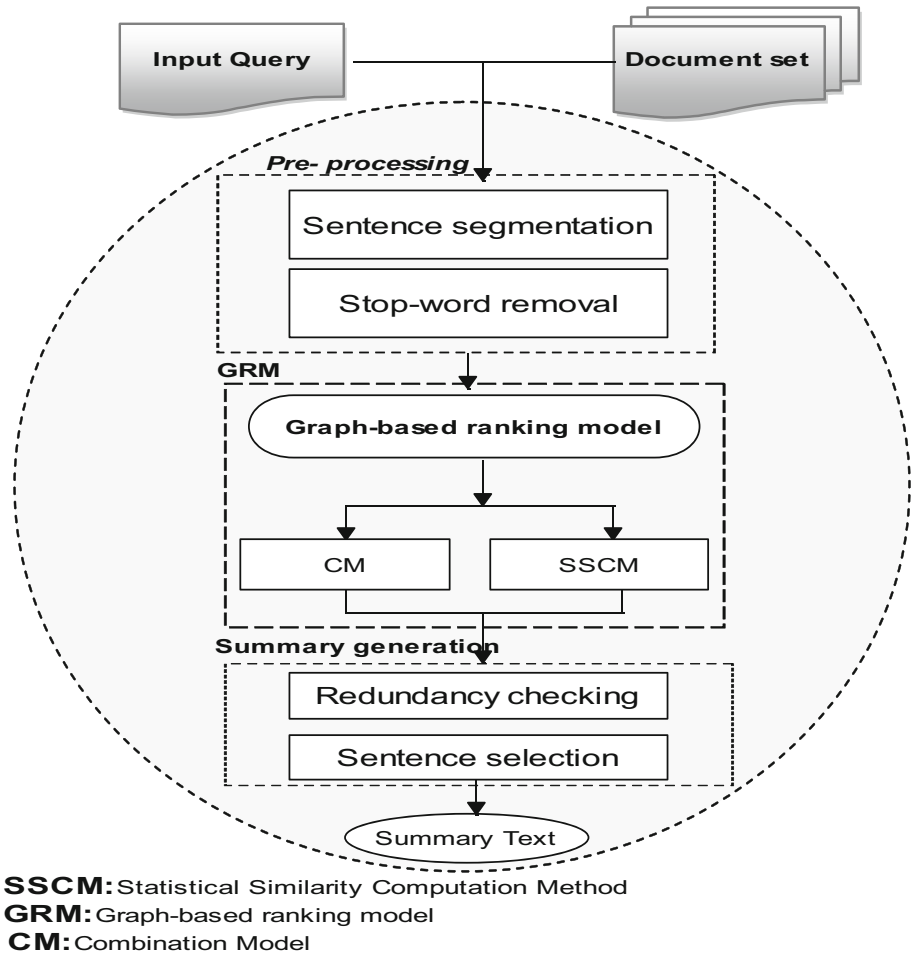

Fig. 1 The architecture of the QSLK

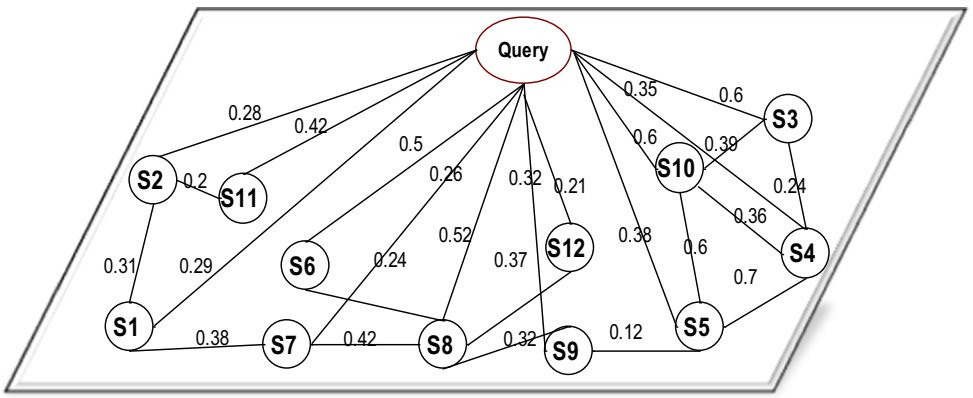

Fig. 2 Graph-based ranking model

3.2.1 Statistical similarity computation method (SSCM)

This method 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 semantic and syntactic information to calculate the similarity measure is shown in Fig. 3. The SSCM works as follows:

1. It takes two sentences S_1 and S_2 as its input.
2. It creates a word set using the two sentences.
3. It creates a semantic vector for each of the two sentences.
4. It creates a word order vector for each of the two sentences.
5. It uses the content word expansion (CWE) method to expand the words in the query and the sentence. Steps 3 and 4 employ this method to create the semantic vector and word order vector.

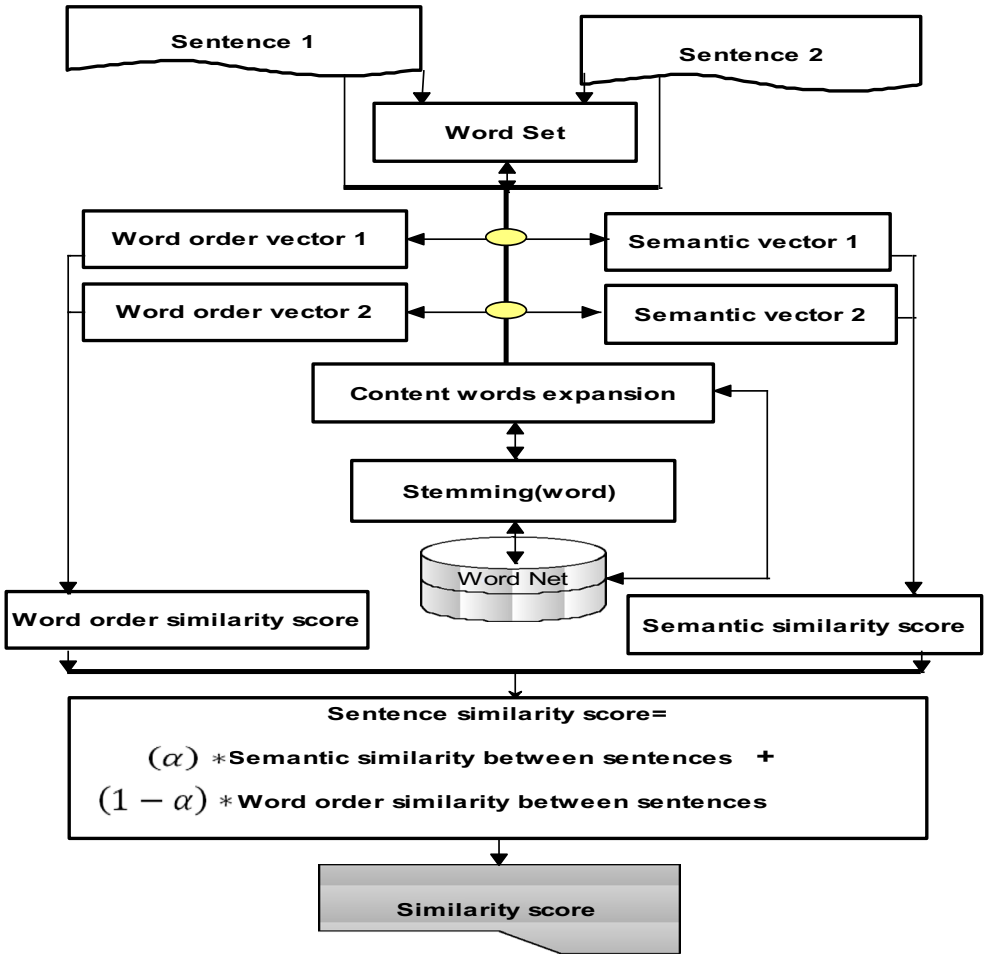

Fig. 3 Sentence (query) similarity computation

6. It computes the semantic similarity measure between two sentences. The semantic similarity measure is determined by the cosine between the two corresponding semantic vectors.
7. It calculates the word order similarity measure between two sentences. The similarity score is determined by the syntactic vector approach (Li et al. 2006b). This approach will be explained in the next section.
8. 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 6 and 7.

9 The final score obtained from the previous step is assigned to the edge between two sentences S_1 and S_2.

Figure 3 includes several components such as word set, context word expansion, semantic similarity and syntactic similarity between sentences. The tasks of each component are as follows:

The word set

Given two sentences S_1 and S_2, a "word set" is created using distinct words from the pair of sentences. Let WS = $\{W_1, W_2, \ldots, W_N\}$ denote word set, where N is the number of distinct words in the word set. The word set between two sentences is obtained as follows:

1. Two sentences are taken two sentences as input.
2. Using a loop for each word, w, from S_1, certain tasks are undertaken, which include:
   (i) Determining the root of w (denoted by RW) using the WordNet.
   (ii) If the RW appears in the WS, jumping to step 2 and continuing the loop using the next word from S_1, otherwise, jumping to step iii;
   (iii) If the RW does not appear in the WS, then assigning the RW to the WS and then jumping to step 2 to continue the loop using the next word from S_1.
   (iv) Conducting the same process for sentence 2.

Content word expansion (CWE) method

As we can recall from Sect. 1, due to the problem of the information limit in the input query and the sentences, the query and sentences need to be expanded. Hence, the QSLK

employs the content word expansion approach 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 these steps:

1. Two words, W_1 and W_2, are taken as input.
2. Stemming: the root of each word is obtained using the lexical database, WordNet.
3. The synonym of each word is obtained using the WordNet.
4. The number of synonyms for each word is determined.
5. The Least Common Subsume (LCS) of two words and their length are determined.
6. The similarity score between words using Eqs. (1) and (2) is computed.

We use the following equations to calculate the semantic similarity between two words (Aytar et al. 2008; Mihalcea et al. 2006; Warin 2004):

$$\mathrm{IC}(w) = 1 - \frac{\mathrm{Log}(\mathrm{synset}(w) + 1)}{\log(\max\_w)} \tag{1}$$

$$\mathrm{Sim}(w_1, w_2) = \begin{cases} \frac{2*\mathrm{IC}(\mathrm{LCS}(w_1,w_2))}{\mathrm{IC}(w_1)+\mathrm{IC}(w_2)} & \text{if } w_1 \neq w_2 \\ 1 & \text{if } w_1 = w_2 \end{cases} \tag{2}$$

where LCS stands for the least common subsume, max_w is the number of words in Word Net, Synset (w) is the number of synonyms of word w, and IC (w) is the information content of word w based on the lexical database WordNet.

Stemming it is used to reduce word to its stem form. It is useful to identify words that belong to the same stem (e.g., *went* and *gone*, both come from the verb *go*). This process obtains the root of each word using the lexical database, Word Net. Word Net is a lexical database for English which was developed at Princeton University (Miller and Charles 1991) It includes 121,962 unique words, 99,642 synsets (each synset is a lexical concept represented by a set of synonymous words) and 173,941 senses of words.

Semantic similarity between sentences

We use the semantic vector approach (Alguliev et al. 2011; Li et al. 2006b) to measure the semantic similarity between sentences. The following tasks are performed to measure the semantic similarity between two sentences.

1. To create the semantic vector.
   The semantic vector is created using the word set and corresponding sentence. Each cell of the semantic vector corresponds to a word in the word set, so the dimension equals the number of words in the word set.
2. To weight each cell of the semantic vector.
   Each cell of the semantic vector is weighted using the calculated semantic similarity between words from the word set and corresponding sentence. As an example:
   (i) If the word w, from the word set appears in the sentence S_1, the weight of w in the semantic vector is set to 1. Otherwise, go to the next step;
   (ii) If the sentence S_1 does not contain w, then compute the similarity score between w and the words from sentence S_1 using the CWE method.
   (iii) If similarity values exist, then the weight of w in the semantic vector 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 semantic vector is set to 0.
3. A semantic vector is created for each of the two sentences. The semantic similarity measure is computed based on the two semantic vectors. The following equation is used to calculate the semantic similarity between sentences:

$$\mathrm{Sim}_{\mathrm{semantic}}(S_1, S_2) = \frac{\sum_{j=1}^{m}(w_{1j} \times w_{2j})}{\sqrt{\sum_{j=1}^{m} w_{1j}^2} \times \sqrt{\sum_{j=1}^{m} w_{2j}^2}} \tag{3}$$

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 jth word in vector S_p, m is the number of words.

Word order similarity between sentences

We use the syntactic vector approach (Li et al. 2006b) to measure the word order similarity between sentences. The following tasks are performed to measure the word order similarity between two sentences.

1. To create the syntactic vector.
   The syntactic vector is created using the word set and corresponding sentence. The dimension of the current 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 the syntactic vector is determined by the following steps:
   (i) For each word, w, from the word set, If w appears in the sentence S_1 the cell in the syntactic vector is set to the index position of the corresponding word in sentence S_1. Otherwise, go to the next step;
   (ii) If the word w does not appear in sentence S_1, then compute the similarity score between w and the words from sentence S_1 using the CWE method.

(iii) If similarity values exist, then the value of the cell is set to the index position of the word from sentence S_1 with the highest similarity measure.
(iv) If there is not a similarity value between w and the words in sentence S_1, the weight of the cell in the syntactic vector is set to 0.

3 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 word order similarity between sentences:

$$\text{Sim}_{\text{word order}}(\boldsymbol{S}_1, \boldsymbol{S}_2) = 1 - \frac{||O_1 - O_2||}{||O_1 + O_2||} \tag{4}$$

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_{pj} is the weight of the jth cell in vector O_p.

Sentence similarity measurement

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

$$\text{Sim}_{\text{sentences}}(S_1, S_2) = \alpha \cdot \text{sim}_{\text{semantic}}(S_1, S_2) + (1 - \alpha) \cdot \text{sim}_{\text{word order}}(S_1, S_2) \tag{5}$$

where $0 < \alpha < 1$ is the weighting parameter, specify the relative contributions to the overall similarity measure of the semantic and syntactic similarity measures. The larger the α, the heavier the weight for semantic similarity. If $\alpha = 0.5$, the semantic and syntactic similarity measures are assumed to be equally important.

3.2.2 Combination model (CM)

The main goal of query-based summarization is to select sentences which are more relevant to the input query. Hence, sentences which are similar to the input query must obtain a high score. 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 similarity between a sentence and a 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 modeled by the following combination model (Badrinath et al. 2011; Chali et al. 2011; Otterbacher et al. 2005; Zhao et al. 2009):

$$P(s|q) = \beta \times \frac{\text{Sim}(s, q)}{\sum_{z \in C} \text{Sim}(z, q)} + (1 - \beta) \times \sum_{v \in C} \frac{\text{Sim}(s, v)}{\sum_{z \in C} \text{Sim}(z, v)} \times P(v|q) \tag{6}$$

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 in the document set.

$0 \leq \beta \leq 1$ 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 in document set. The bigger the β, the heavier the weight for the Q-to-S similarity. If $\beta = 0.5$, 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, $\text{sim}(s, v)$, and the similarity measure between a sentence and the query, $\text{sim}(s, q)$ are calculated based on the statistical similarity computation method.

Following (Erkan and Radev 2004; Otterbacher et al. 2005), Eq.(6) can be written in matrix form as follows:

$$\begin{cases} P_{(k+1)} = M^T P_k \\ M = \beta U + (1 - \beta) W \end{cases} \tag{7}$$

where W, U and M are square matrices. All elements in matrix W represent the similarity measure between sentences and the elements in matrix U represent the similarity measure between sentences and the input query. Both matrices are normalized to make the sum of each row equal to 1. K represents the Kth iteration. β is a weighting parameter between [0,1]. The vector $P = [p_1, \ldots, p_N]$ corresponds to the stationary distribution of the matrix M. The combination model based on Eq. (7) is performed by carrying out the following steps.

1. Given two sentences S_i and S_j, the similarity measure between two sentences is calculated using the SSCM. Additionally, given a sentence and the query, the similarity measure between the sentence and the query is calculated using the SSCM.
2. Create the square matrix W using $W_{i,j} = \text{Sim}(S_i, S_j)$. Additionally, create a square matrix U using $U_{ij} = \text{Sim}(S_i, q)$. W and U should be normalized to make the sum of each row equal to 1.
3. Iterate $P_{(k+1)} = [\beta U + (1 - \beta) W]^T P_k$ until the loop constraint is reached. Usually, the iteration is terminated when $||P_{(k+1)} - P_k||$ is smaller than the threshold value, defined by the user. Vector P is initialized as the uniform distribution $\left[\frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N}\right]$.
4. Let P denote the result. Each sentence S_i obtains its ranking score corresponding to $p_i \quad (1 \leq i \leq N)$.

4 Summary generation

Once the sentences are ranked using the graph-based ranking model, the simple approach to forming the final summary

is just to select the sentences with the highest score values until the required summary length is reached. Finally, the summary includes multiple extracted sentences from several documents. In this case, since various sentences may include similar content, it is necessary to reduce redundancy and increase the information coverage in the summary. In our method, to tackle this problem, we employ two main levels of analysis: first, a scoring level, where every sentence of the document set is scored using the graph-based model and second, a comparison level, where, before adding the sentences to the final summary, the sentences assumed to be significant are compared to each other and only those that are not too similar to other candidates are contained in the final summary. To do this, we use the greedy algorithm (Wan et al. 2007; Zhang et al. 2005) to impose a diversity penalty on the sentences to remove redundancy. The algorithm includes the following steps.

1. Create 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. (7), i.e., score $(S_i) = P(S_i|q), i = 1, 2, \ldots, N$.
2. Sort the sentences in A_2 based on their scores in descending order.
3. If S_i is a sentence with 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 $S_j \in A_2$,

$$\text{Score}(S_j) = \text{Score}(S_j) - (\text{Sim}(S_i, S_j) \times P(S_i|q)) \quad (8)$$

where, $\text{Sim}(s_i, s_j)$ is similarity measure between two sentences defined in Eq. 5.
4. Go to step 2 and iterate until $A_2 = \emptyset$ or the summary length limitation is satisfied.

Finally, the sentences in the set A_1 are added to the summary.

5 Experiments

Our proposed method, QSLK, was applied for query-based multi-document summarization. We conducted the experiments on data sets provided by Document Understanding Conference (http://duc.nist.gov).

5.1 Data set

In this section, we describe the data used throughout our experiments. For assessment of the performance of the proposed method, we used the datasets provided by DUC 2005 and DUC 2006. The DUC 2005 and DUC 2006 data sets include 50 document clusters. Each cluster of DUC 2005 data set consists of 32 relative documents. Each cluster of DUC 2006 data sets consists for 25 relative documents. The query-based multi-document summarization was the only task of DUC 2005 and DUC 2006. In other words, "given a complex question (topic description) and a collection of relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the documents that answers the question(s) in the topic". It should be mentioned that the summary produced by our proposed method is denoted as the candidate summary and the summary provided by the experts for each document set is denoted the reference summary. To evaluate the performance of our method, we conducted two experiments. In the first experiment, we used the data from DUC 2005 for parameter tuning (β and α). In the second experiment, we used the data provided by DUC 2006 to compare our method with the existing systems that had been used in DUC 2006. A brief information is shown in Table 1.

Table 1 Description of dataset

	DUC 2005	DUC 2006
Number of cluster	50	50
Number of documents in each cluster	32	25
Average number of sentences per cluster	1003	816
Data source	TREC	AQUAINT
Summary length	250 words	250 words

5.2 Evaluation metrics

To evaluate and compare the performance of our proposed method, we used the standard ROUGE metric (Lin 2004). ROUGE was adopted by Document Understanding Conference as the official evaluation metric for text summarization. Lin (2004) proposed the automatic summary assessment system named Recall-Oriented Understudy for Gisting Evaluation, which is used to assess the quality of the summary text. The current system includes various automatic assessment approaches, such as ROUGE-N, ROUGE-L, and ROUGE-S. ROUGE-L calculates the similarity between a reference summary and a candidate summary based on the longest common subsequence (LCS). ROUGE-S is a measure of the overlap of skip-bigrams between a candidate and a reference summary. ROUGE-SU4 (skip-bigram based on maximum skip distance of 4, plus unigram). ROUGE-N compares two summaries, the system summary and the human summary, based on the total number of matches. It is calculated as follows:

$$\text{ROUGE}-N = \frac{\sum_{S \epsilon \text{Reference summaries}} \sum_{N\text{-gram} \varepsilon S} \text{Count}_{\text{match}}(N\text{-gram})}{\sum_{S \epsilon \text{Reference summaries}} \sum_{N\text{-gram} \varepsilon S} \text{Count}(N\text{-gram})} \quad (9)$$

Fig. 4 Steps to optimize α and β

```
Data set: DUC 2005;
Sentence segmentation;
Stop word removal;
For β = 0 to β = 1
    For α = 0.1 to α = 0.9
        Begin
            Run the proposed method on the current dataset using Eqs. (5), (7) and (9).
        End
```

where N is used for the length of the N-gram and Count $_{match}$ (N-gram) is the total number of N-grams co-occurring in a reference summary and a candidate summary. Count (N-gram) is the number of N-grams in the reference summaries.

In our evaluation, we used three metrics of ROUGE: ROUGE-1, ROUGE-2 and ROUGE-SU4. We reported the Recall score of ROUGE-1, ROUGE-2 and ROUGE-SU4 to assess and compare our method, QSLK, with other methods.

5.3 Parameter setting

As it is stated before in the Sect. 5.1, QSLK is mainly tested over DUC 2005 and DUC 2006 data sets. In the experiments over datasets, we first focused on optimizing QSLK parameters. To be more specific, we first try to optimize: (1) a weighting parameter (α) for weighting the significance between semantic information and syntactic information and (2) a trade-off parameter (β), which is a trade-off between the similarity of a sentence to the query and to other sentences in the document sets. To accomplish this, we randomly selected set of documents from DUC 2005 dataset and ran QSLK to find the optimal parameter values. All documents were decomposed into sentences. The stop words from both the query and the sentences were removed. We ran our proposed method on the current data set. We used Eqs. (5), (7) and (9). We also used the greedy algorithm to reduce redundancy, described in Sect. 4. Equation 5 was used to calculate the similarity measure. Equation 7 was used to calculate the score value of each sentence in the graph model. Equation 9 was used to calculate the Recall value of ROUGE-1, ROUGE-2 and ROUGE-SU4.

We evaluated our method for each peer (α) between 0.1 to 0.9 with a step of 0.1 and (β) between 0 to 1 with a step of 0.1, (e.g. $\alpha = 0.4$, $\beta = 0.7$). To estimate the values of α and β, we used a nested loop, Fig. 4, where β is outer loop and α is inner loop. In the first pass of the outer loop when the value of β becomes 0 then control enters into the inner loop where α is varied from 0.1 to 0.9 to observe the variation in performance. The second pass of the outer loop also triggers the inner loop again. This repeats until the outer loop finishes. The results of aforementioned nested loop are reported in Table 2 and Fig. 5.

Table 2 and Fig. 5 present the experimental results achieved using various α and β values. We evaluated the results in terms of Recall scores obtained through ROUGE-1, ROUGE-2 and ROUGE-SU4. We also measured the average Recall score using Eq. (10).

$$\text{Average Recall Score (ARS)} = \frac{\text{Recall}_{\text{ROUGE-1}} + \text{Recall}_{\text{ROUGE-12}} + \text{Recall}_{\text{ROUGE-1SU4}}}{3} \tag{10}$$

On analyzing the results, we found that the best performance was achieved with $\alpha = 0.7$ and $\beta = 0.8$. This α and β produced Recall scores for the three ROUGE metrics as follows: 0.3874 (ROUGE-1), 0.0793 (ROUGE-2) and 0.1372 (ROUGE-SU4). We also obtained the best *ARS* 0.2013 with $\alpha = 0.7$ and $\beta = 0.8$. The best values in Table 2 have been marked using boldface. As a result, using the current data set, we obtained the best ARS of 0.2013 when we used 0.7 as the α value and 0.8 as the β value. Therefore, we can recommend these α and β values for use with the DUC 2006 data set.

5.4 Comparison with DUC 2006 systems

To confirm the aforementioned results, we validated our proposed method, QSLK, using a comparison of the overall Recall values obtained by QSLK and the participating systems in DUC 2006 (Hoa 2006): (a) the worst system: System-1 (Baseline) (Hoa 2006). It selects the first sentences from documents to produce the summary until the summary length is satisfied, (b) The top five systems with the highest ROUGE scores: System-8 (JIKD) (Conroy et al. 2006), System-12 (onModer) (Ye and Chua 2006), System-23 (ICL_SUM) (Li et al. 2006a), System-24 (IIITH_Sum) (Jagarlamudi and Varma 2006) and System-28 (LIA_THALES) (Favre B et al. 2006).

We applied our method to the DUC 2006 data set only with α value 0.7 and β value 0.8. Table 3 and Fig. 6 present the results obtained for Recall for the three ROUGE metrics with α of 0.7 and β of 0.8. The results obtained prove that QSLK outperformed the other methods examined and that our method produced very competitive results. QSLK was also able to obtain an ARS of (23.095 %) in comparison with

Table 2 Performance of the QSLK against various α and β values

Trade-off (β)	Weighting (α)	ROUGE-1	ROUG-2	ROUGE-SU4	ARS
$\beta = (0. . .0.7)$	0.1	–	–	–	–
	–	–	–	–	–
	0.9	–	–	–	–
$\beta = 0.8$	0.1	0.3678	0.0751	0.1299	0.1909
	0.2	0.3696	0.0758	0.1307	0.1921
	0.3	0.3733	0.0776	0.1325	0.1945
	0.4	0.3771	0.0789	0.1343	0.1968
	0.5	0.3823	0.0793	0.1375	0.1997
	0.6	0.3855	0.0800	0.1361	0.2005
	0.7	0.3874	0.0793	0.1372	0.2013
	0.8	0.3862	0.0801	0.1371	0.2011
	0.9	0.3853	0.0792	0.1374	0.2006
$\beta = (0.9 \ldots 1)$	0.1	–	–	–	–
	–	–	–	–	–
	0.9	–	–	–	–

Due to the space limitations of this paper, a sample results are shown

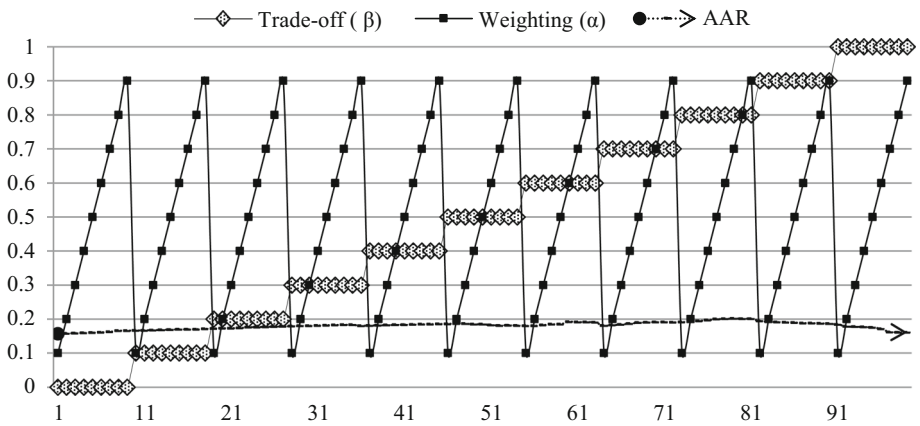

Fig. 5 Performance of the QSLK against various α and β values

Table 3 Performance comparison between QSLK and DUC 2006 systems

ROUGE values of the methods				
System	ROUGE-1	ROUGE-2	ROUGE-SU4	ARS
QSLK	0.42873	0.09682	0.16731	0.23095
OnModer	0.40488	0.08987	0.14755	0.21410
ICL_SUM	0.40440	0.08792	0.14486	0.21239
JIKD	0.38807	0.08707	0.14134	0.20549
LIA_THALES	0.39922	0.08700	0.14522	0.21048
IIITH_Sum	0.40980	0.09505	0.15464	0.21983
Baseline	0.30217	0.04947	0.09788	0.14984

the best existing method, IIITH_Sum, which had an ARS of (21.983 %).

5.5 Comparison with related methods

In this section, the performance of our method is compared with other well-known or recently proposed methods. In particular, to evaluate our methods on DUC 2006, we select the following methods: (1) LEX (Huang et al. 2010), (2) TMR (Tang et al. 2009), (3) SVR (Ouyang et al. 2010), (4) Topical-N (Yang et al. 2013), (5) QEMD (Zhao et al. 2009), (6) Qs-MR (Wei et al. 2011), (7) CTMSUM (Guangbing 2014) and (8) WAASum (Canhasi and Kononenko 2014). These methods have been chosen for comparison because

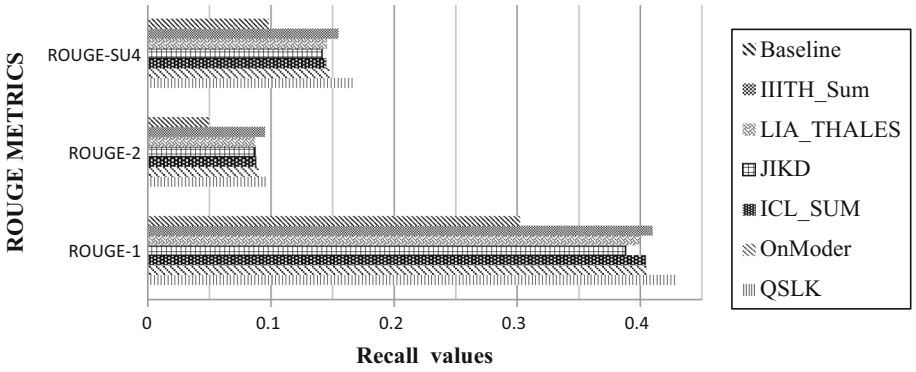

Fig. 6 Performance comparison of the QSLK with DUC 2006 systems

Table 4 Performance comparison between QSLK and other methods on DUC 2006

ROUGE values of the methods				
System	ROUGE-1	ROUGE-2	ROUGE-SU4	ARS
QSLK	0.4287	0.0968	0.1673	0.2310
LEX	0.4030	0.0913	0.1449	0.2131
TMR	0.4063	0.0913	0.1504	0.2160
SVR	0.4018	0.0926	0.1485	0.2143
Topical-N	0.4010	0.0893	0.1459	0.2121
QEMD	0.4026	0.0932	0.1473	0.2144
Qs-MR	0.4012	0.0914	0.1444	0.2123
CTMSUM	0.4157	0.0968	0.1548	0.2224
WAASum	0.4238	0.0917	0.1671	0.2275

they have achieved the best results on the DUC 2006 data set. The summarization accuracy by ROUGE metrics is reported in Table 4 and Fig. 7.

5.6 Detailed comparison

From the comparison of the ROUGE values for DUC 2006 systems and other methods, QSLK obtains a considerable improvement. Tables 5 and 6 show the improvement of QSLK for all three ROUGE metrics. It is clear that QSLK obtained the highest ARS and outperformed all the other methods. We used the relative improvement $(\frac{\text{Our method}-\text{Other method}}{\text{Other method}}) \times 100$ for comparison. In Tables 5 and 6, "+" means the proposed method improves the DUC 2006 systems and existing methods. Table 5 shows among the DUC 2006 systems the IIITH_Sum displays the best results compared to OnModer, ICL_SUM, JIKD, LIA_THALES and Baseline. In comparison with the IIITH_Sum method, QSLK improved its performance as follows: 4.6193 % (ROUGE-1), 1.8622 % (ROUGE-2), 8.1932 % (ROUGE-SU4) in terms of Recall.

Table 6 also displays among the existing methods the WAASum shows the best results compared to LEX, *TMR*, *SVR*, Topical-N, CTMSUM, *QEMD* and Qs-MR. In comparison with the method *WAASum*, QSLK improves the performance of the WAASum method as follows: 1.1633 % (ROUGE-1), 5.5834 (ROUGE-2), 0.1257 % (ROUGE-SU4) in terms of Recall.

To display the comparison of methods more clearly, we present it in histograms. The comparison between the overall performance achieved by QSLK, DUC 2006 systems and the other methods for the similar dataset is presented in Figs. 8 and 9, respectively.

5.7 Statistical significance test

To statistically compare the performance of QSLK with other summarization methods, we use a non-parametric statistical significance test, called Wilcoxon's matched-pairs signed

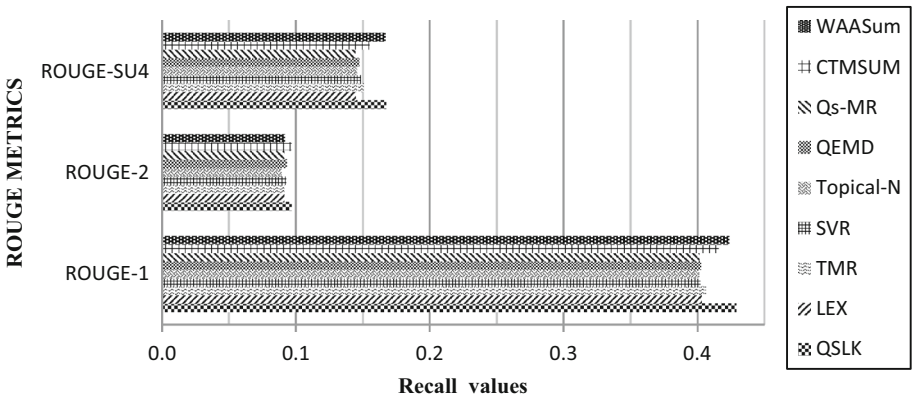

Fig. 7 Performance comparison between QSLK and other methods on DUC 2006

Table 5 Performance evaluation compared between the QSLK and DUC 2006 systems

QSLK improvement (%)						
Metrics	OnModer	ICL_SUM	JIKD	LIA_THALES	IIITH_Sum	Baseline
ROUGE-1	+5.8906	+6.0163	+10.4775	+7.3919	+4.6193	+41.8837
EOUGE-2	+7.7334	+10.1228	+11.1979	+11.2874	+1.8622	+95.7146
ROUGE-SU4	+13.3921	+15.4977	+18.3741	+15.2114	+8.1932	+70.9338

Table 6 Performance evaluation compared between the QSLK and other methods on DUC 2006

QSLK improvement (%)								
Metrics	LEX	TMR	SVR	Topical-N	QEMD	Qs-MR	CTMSUM	WAASum
ROUGE-1	+6.3846	+5.5206	+6.7023	+6.9152	+6.4982	+6.8619	+3.1345	+1.1633
EOUGE-2	+6.0460	+6.0460	+4.5572	+8.4211	+3.8396	+5.9300	+0.0207	+5.5834
ROUGE-SU4	+15.466	+11.243	+12.667	+14.674	+13.570	+15.866	+8.0814	+0.1257

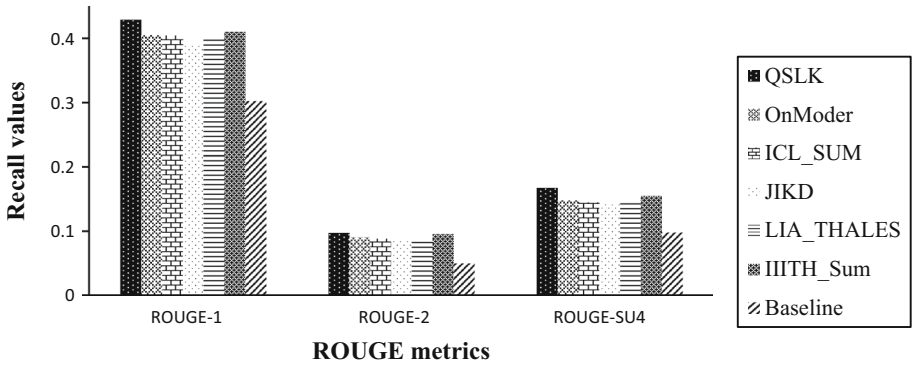

Fig. 8 Performance comparison between the QSLK and DUC 2006 systems

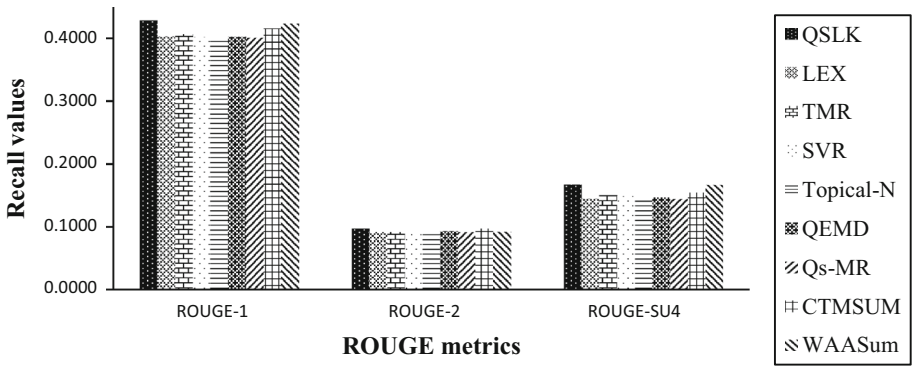

Fig. 9 Performance comparison between the QSLK and other methods on DUC 2006

rank based statistical test, to determine the significance of our results. The statistical significance test for independent samples has been conducted at the 5 % significance level of the summarization results. Nine groups, corresponding to the nine methods: (1) LEX, (2) TMR, (3) SVR, (4) Topical-N, (5) QEMD, (6)Qs-MR, (7) CTMSUM, (8) WAASUM, (9) QSLK, have been created for data set. Two groups are compared at a time one corresponding to QSLK method and the other corresponding to some other method considered in this paper. Each group consists of the ROUGE-1 and ROUGE-2 scores for the data set produced by each corresponding method.

The median values and standard deviation (Stdv.) of ROUGE-1 and ROUGE-2 scores of each method for the data set are presented in Table 7. As is evident from Table 7, the median values of ROUGE-1 and ROUGE-2 for QSLK method on data set are better than that for the other methods. To establish that this goodness is statistically significant, Table 8 reports the *P* values produced by Wilcoxon's matched-pairs signed rank test for comparison of two groups (one group corresponding to QSLK and another group corresponding to some other method) at a time. As a null hypothesis, it is assumed that there are no significant differences between the median values of two groups. Whereas

Table 7 Median values and standard deviation of methods on DOC 2006

Method	ROUGE 1		ROUGE 2	
	Median	Stdv.	Median	Stdv.
QSLK	0.6515	4.9E−02	0.0769	7.5E−03
LEX	0.5870	6.0E−02	0.0649	1.1E−02
TMR	0.6005	7.9E−02	0.0695	1.1E−02
SVR	0.5405	8.9E−02	0.0668	9.6E−03
Topical-N	0.4370	9.4E−02	0.0521	1.4E−02
QEMD	0.5650	7.8E−02	0.0691	1.1E−02
Qs-MR	0.5225	7.0E−02	0.0608	1.9E−02
CTMSUM	0.5635	8.5E−02	0.0709	1.0E−02
WAASum	0.6170	8.5E−02	0.0711	8.7E−03

the alternative hypothesis is that there is significant difference in the median values of the two groups. It is clear from Table 8 that P values are much less than 0.05 (5 % significance level). For example, the Wilcoxon's matched-pairs signed rank test between the algorithms QSLK and WAASum for DUC 2006 provides a P value of 0.037 (ROUGE-1), which is very small. This is strong evidence against the null hypothesis, indicating that the better median values of the performance metrics produced by QSLK is statistically significant and has not occurred by chance. Similar results are obtained for all other methods compared to QSLK method, establishing the significant superiority of the proposed method. From the statistical results, we observe that our QSLK method significantly outperforms the other baseline summarization methods.

A visual comparison of statistical significance is provided in Fig. 10. This figure shows the median values of ROUGE-1 and ROUGE-2 scores obtained by each method on the DUC 2006 data set. It can be observed that ROUGE-1 and ROUGE-2 values of QSLK are noticeably better than that of other methods. In addition, according to the statistical significance test, QSLK is more stable than the other methods. For convincing, we address the readers to pay an attention to the values of the standard deviation (Stdv.) in Table 7.

5.8 Discussion

The current section presents the main findings that were obtained from Tables 3, 4, 5 and 6. Our method was able to outperform all other systems. This is due to the fact that, (a) It is able to identify the synonyms or similar words among all sentences using a lexical database, Word Net. It is very important to consider this aspect (identifying the synonyms or similar words) when measuring the similarity score of $S2S$and $Q2S$, to tackle the information limit of the query and the sentences.

(b) Given two sentences (i.e., S_1*John likes Ravi;* S_2*Ravi likes John*), unlike JIKD, onModer, ICL_SUM, IIITH_Sum, LIA_THALES and Baseline, our method is able to distinguish the meanings of the two sentences using a combination of semantic and syntactic information.

(c) Baseline and ICL_SUM do not consider word expansion in calculating the similarity measure between sentences

Table 8 P values produced by Wilcoxon's matched-pairs signed rank test by comparing QSLK with other methods

Data set	LEX	TMR	SVR	Topical-N	QEMD	Qs-MR	CTMSUM	WAASum
DUC 2006	*Comparing medians of ROUGE-1 metric of QSLK with other methods*							
	0.003	0.040	0.001	0.000	0.001	0.000	0.001	0.037
	Comparing medians of ROUGE-2 metric of QSLK with other methods							
	0.001	0.036	0.012	0.015	0.010	0.007	0.042	0.001

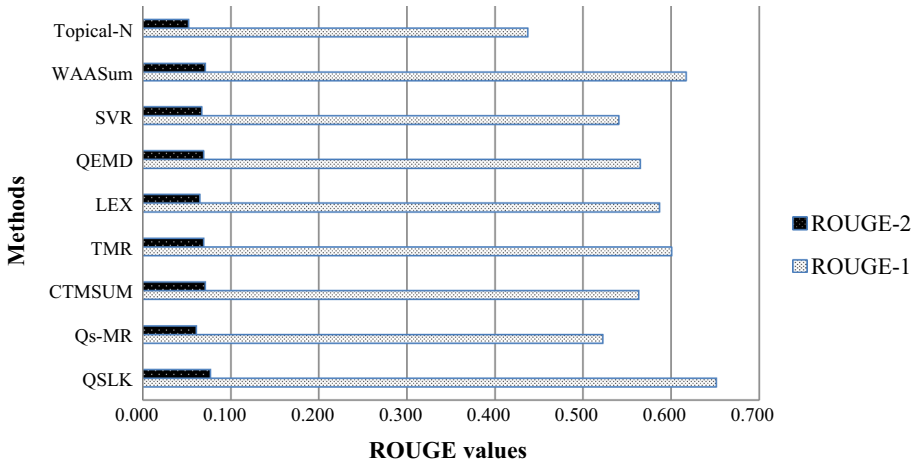

Fig. 10 Median values of different summarization method on DUC 2006

and sentence-to-query. Baseline also does not apply any method for the imposing of a redundancy penalty.

(d) LIA_THALES applies the maximal-marginal-relevance (MMR) (Carbonell and Goldstein 1998; Goldstein et al. 2000) to control sentence redundancy. It runs the MMR on Latent Semantic Analysis (LSA) (Landauer 2002). LSA uses a predefined word list including hundreds of thousands of words (Landauer et al. 1998) for measuring the similarity between two sentences; this drawback can lead to some important words from the input texts not being considered in the LSA space. Our proposed method computes the similarity score between two sentences based on the words in compared sentences. LSA, with high dimensionality and high sparsity, has an impact on the performance of similarity measuring (Burgess et al. 1998; Salton 1989). LSA is a 'bag-of-words' method and does not take into account syntactic information for computing the text similarity score (Kanejiya et al. 2003; Pérez et al. 2005; Wiemer-Hastings and Zipitria 2001).

(e) Tables 4 and 6 show that our method obtained good result in ROUGE score. The results confirm that our method outperforms the other methods. Moreover, the results show that the combination of semantic and syntactic information; and the content word expansion can improve the performance.

6 Conclusion

With the explosive growth of the volume and complexity of document data on the Internet, multi-document summarization provides a useful solution for understanding documents and reducing information overload. Hence, we need effective summarization methods to analyze and extract the important information. A good summary is expected to preserve the important information contained in the documents as much as possible, and at the same time to contain as little redundancy as possible. In this paper, we propose a method to produce summaries for query-based multi-documents tasks. Our method in this work not only combines semantic and syntactic information to capture the meaning when comparing sentences-to-sentence and query-to-sentence, but also considers content word expansion to improve the quality of summaries and extract the more query relevant sentences from a document set.

The evaluation of QSLK is conducted over DUC dataset that comprises a wide variety of text lengths. The proposed method is very easy to follow and requires minimal text processing cost. Initially, parameters of QSLK are optimized over the DUC 2005 dataset. Later, we used the DUC 2006 data set to assess the performance of QSLK using the Recall score of ROUGE metrics. QSLK is compared with the participating system in DUC 2006 and the current methods which are well-known existing methods that are used for query-based multi-documents tasks. The experimental results display that the performance of the proposed method is very competitive when compared with other methods. The results also displayed that PDLK improved the performance of the participating system in DUC 2006 and the current methods. We observed that QSLK IS able to obtain an ARF of 23.10 % in comparison with the best participating system in DUC 2006, (IIITH_Sum), which had ARF of 21.983 % and the best existing system, (WAASum), which had ARF of 22.75 %.

As future work, we plan to improve the proposed method by considering identifying passive and active sentence, and expanding the semantic knowledge base, which are limitations of the current method. (a) The method is not able to distinguish between an active sentence and a passive sentence. Given a suspicious sentence (A: 'Teacher likes his student') and two source sentences (B: 'student likes his teacher'; C: 'student is liked by his teacher'), although the similarity measure between sentences (A and B) and (A and C) is same, but as we can see the meaning of sentence A is more similar to the sentence C. Hence, it is important to know what passive and active sentences are before comparisons can be drawn. (b) The method used WordNet as the main semantic knowledge base to calculate the semantic similarity between words. The comprehensiveness of Word Net is determined by the proportion of words in the text that are covered by its knowledge base. However, the main criticism of WordNet concerns its limited word coverage to calculate semantic similarity between words. Obviously, this disadvantage has a negative effect on the performance of our proposed algorithm. To tackle this problem, in addition to WordNet, other knowledge resources, such as Wikipedia and other large corpus should be used.

In addition to the aforementioned future works, the following works are also considered as future works. In future, we aim to extend our method to generic multi-document summarization. In query-based summarization, it is clear that sentences which are more relevant to the query are selected for the summary. But in generic multi-document summarization, we have to reduce the documents in size and extract the sentences that represent the main ideas of the text collection.

Compliance with ethical standards

Conflict of interest I hereby and on behalf of the co-authors declare all the authors agreed to submit the article exclusively to this journal and also declare that there is no conflict of interests regarding the publication of this article.

References

Abdi A, Idris N (2014) Automated summarization assessment system: quality assessment without a reference summary. In: The international conference on advances in applied science and environmental engineering (ASEE). IRED Press

Abdi A, Idris N, Alguliev RM, Aliguliyev RM (2015) Automatic summarization assessment through a combination of semantic and syntactic information for intelligent educational systems. Inf Process Manag 51:340–358

Alguliev RM, Aliguliyev RM, Mehdiyev CA (2011) Sentence selection for generic document summarization using an adaptive differential evolution algorithm. SwarmEvol Comput 1:213–222

Aliguliyev RM (2009) A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Syst Appl 36:7764–7772

Aytar Y, Shah M, Luo J (2008) Utilizing semantic word similarity measures for video retrieval. In: IEEE conference on Computer vision and pattern recognition (CVPR). IEEE, pp 1–8

Badrinath R, Venkatasubramaniyan S, Madhavan CV (2011) Improving query focused summarization using look-ahead strategy. In: Advances in information retrieval. Springer, pp 641–652

Basak D, Pal S, Patranabis DC (2007) Support vector regression. Neural Inf Process Lett Rev 11:203–224

Burgess C, Livesay K, Lund K (1998) Explorations in context space: words, sentences, discourse. Discourse Process 25:211–257

Canhasi E, Kononenko I (2014) Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. Expert Syst Appl 41:535–543

Carbonell J, Goldstein J (1998) The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 335–336

Chali Y, Hasan SA, Joty SR (2011) Improving graph-based random walks for complex question answering using syntactic, shallow semantic and extended string subsequence kernels. Inf Process Manag 47:843–855

Conroy JM, Schlesinger JD, O'leary DP, Goldstein J (2006) Back to basics: CLASSY 2006. In: Proceedings of DUC

Davidson I, Ravi S (2005) Agglomerative hierarchical clustering with constraints: theoretical and empirical results. In: Knowledge discovery in databases: PKDD. Springer, pp 59–70

Erkan G, Radev DR (2004) LexRank: graph-based lexical centrality as salience in text summarization. J Artif Intell Res 22:457–479

Favre B et al (2006) The LIA-Thales summarization system at DUC-2006. In: Proceedings of document understanding conference (DUC-2006), New York, USA

Goldstein J, Mittal V, Carbonell J, Kantrowitz M (2000) Multi-document summarization by sentence extraction. In: Proceedings of the 2000 NAACL-ANLP workshop on automatic summarization-volume 4. Association for Computational Linguistics, pp 40–48

Guangbing Y (2014) A novel contextual topic model for query-focused multi-document summarization. In: IEEE 26th international conference on tools with artificial intelligence (ICTAI), 10–12 Nov 2014, pp 576–583. doi:10.1109/ICTAI.2014.92

He Q, Hao H-W, Yin X-C (2012) Query-based automatic multi-document summarization extraction method for web pages. In: Proceedings of the 2011 2nd international congress on computer applications and computational science. Springer, pp 107–112

Hoa H (2006) Overview of DUC 2006. In: Document understanding conference. New York City

Hu P, He T, Wang H (2010) Multi-view sentence ranking for query-biased summarization. In: 2010 international conference on computational intelligence and software engineering (CiSE). IEEE, pp 1–4

Huang L, He Y, Wei F, Li W (2010) Modeling document summarization as multi-objective optimization. In: 2010 third international symposium on intelligent information technology and security informatics (IITSI). IEEE, pp 382–386

Idris N, Baba S, Abdullah R (2009) A summary sentence decomposition algorithm for summarizing strategies identification. Comput Inf Sci 2:P200

Jagarlamudi PPJ, Varma V (2006) Query independent sentence scoring approach to duc 2006. In: In Proceeding of document understanding conference (DUC-2006)

Kanejiya D, Kumar A, Prasad S (2003) Automatic evaluation of students' answers using syntactically enhanced LSA. In: Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-volume 2. Association for Computational Linguistics, pp 53–60

Landauer TK (2002) On the computational basis of learning and cognition: arguments from LSA. Psychol Learn Motiv 41:43–84

Landauer TK, Foltz PW, Laham D (1998) An introduction to latent semantic analysis. Discourse process 25:259–284

Lee J-H, Park S, Ahn C-M, Kim D (2009) Automatic generic document summarization based on non-negative matrix factorization. Inf Process Manag 45:20–34

Li S, Ouyang Y, Sun B, Guo Z (2006a) Peking University at DUC 2006. In: Proceedings of DUC2006

Li Y, McLean D, Bandar ZA, O'shea JD, Crockett K (2006b) Sentence similarity based on semantic nets and corpus statistics. IEEE Trans Knowl Data Eng 18:1138–1150

Lin C-Y (2004) Rouge: a package for automatic evaluation of summaries. In: Text summarization branches out: proceedings of the ACL-04 workshop, pp 74–81

Lloret E, Llorens H, Moreda P, Saquete E, Palomar M (2011) Text summarization contribution to semantic question answering: new approaches for finding answers on the web. Int J Intell Syst 26:1125–1152

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

Mendoza M, Bonilla S, Noguera C, Cobos C, León E (2014) Extractive single-document summarization based on genetic operators and guided local search. Expert Syst Appl 41:4158–4169

Mihalcea R, Corley C, Strapparava C (2006) Corpus-based and knowledge-based measures of text semantic similarity. In: AAAI, pp 775–780

Miller GA, Charles WG (1991) Contextual correlates of semantic similarity. Lang Cogn Process 6:1–28

Otterbacher J, Erkan G, Radev DR (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

Ouyang Y, Li W, Li S, Lu Q (2010) Intertopic information mining for query-based summarization. J Am Soc Inf Sci Technol 61:1062–1072

Ouyang Y, Li W, Li S, Lu Q (2011) Applying regression models to query-focused multi-document summarization. Inf Process Manag 47:227–237

Pandit SR, Potey M (2013) A query specific graph based approach to multi-document text summarization: simultaneous cluster and sentence ranking. In: 2013 international conference on machine intelligence and research advancement (ICMIRA). IEEE, pp 213–217

Pérez D, Gliozzo AM, Strapparava C, Alfonseca E, Rodríguez P, Magnini B (2005) Automatic assessment of students' free-text answers underpinned by the combination of a BLEU-inspired algorithm and latent semantic analysis. In: FLAIRS conference, pp 358–363

Saggion H, Poibeau T (2013) Automatic text summarization: past, present and future. In: Multi-source, multilingual information extraction and summarization. Springer, pp 3–21

Salton G (1989) Automatic text processing: the transformation, analysis, and retrieval of. Addison-Wesley, Reading

Sarker A, Mollá D, Paris C (2013) An approach for query-focused text summarisation for evidence based medicine. In: Artificial intelligence in medicine. Springer, pp 295–304

Shekhar S, Xiong H (2008) Nearest neighbor algorithm encyclopedia of GIS:771–771

Tang J, Yao L, Chen D (2009) Multi-topic based query-oriented summarization. In: SDM. SIAM, pp 1147–1158

Varadarajan R, Hristidis V (2006) A system for query-specific document summarization. In: Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, pp 622–631

Wan X, Yang J, Xiao J (2007) Manifold-ranking based topic-focused multi-document summarization. In: IJCAI, pp 2903–2908

Warin M (2004) Using WordNet and semantic similarity to disambiguate an ontology retrieved 25 Jan 2008

Wei F, Li W, He Y (2011) Document-aware graph models for query-oriented multi-document summarization. In: Multimedia analysis, processing and communications. Springer, pp 655–678

Wiemer-Hastings P, Wiemer P (2000) Adding syntactic information to LSA. In: Proceedings of the 22nd annual meeting of the Cognitive Science Society. Citeseer

Wiemer-Hastings P, Zipitria I (2001) Rules for syntax, vectors for semantics. In: Proceedings of the twenty-third annual conference of the Cognitive Science Society, pp 1112–1117

Yang G, Wen D, Sutinen E(2013) A contextual query expansion based multi-document summarizer for smart learning. In: 2013 international conference on signal-image technology & internet-based systems (SITIS). IEEE, pp 1010–1016

Ye S, Chua T-S (2006) NUS at DUC 2006: document concept lattice for summarization. In: Proceedings of DUC

Zhang B et al (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 45:35–41