Information Processing and Management 54 (2018) 318–338

Contents lists available at ScienceDirect

Information Processing and Management

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

QMOS: Query-based multi-documents opinion-oriented summarization

Asad Abdi[a,*], Siti Mariyam Shamsuddin[a], Ramiz M. Aliguliyev[b]

[a] UTM Big Data Centre (BDC), Universiti Teknologi Malaysia, Johor, Malaysia
[b] Institute of Information Technology, Azerbaijan National Academy of Sciences, 9, B. Vahabzade Street, AZ1141 Baku, Azerbaijan

ARTICLE INFO

Keywords:
Sentiment analysis
Sentiment summarization
Sentiment dictionary
Contextual polarity

ABSTRACT

Sentiment analysis concerns the study of opinions expressed in a text. This paper presents the QMOS method, which employs a combination of sentiment analysis and summarization approaches. It is a lexicon-based method to query-based multi-documents summarization of opinion expressed in reviews.

QMOS combines multiple sentiment dictionaries to improve word coverage limit of the individual lexicon. A major problem for a dictionary-based approach is the semantic gap between the prior polarity of a word presented by a lexicon and the word polarity in a specific context. This is due to the fact that, the polarity of a word depends on the context in which it is being used. Furthermore, the type of a sentence can also affect the performance of a sentiment analysis approach. Therefore, to tackle the aforementioned challenges, QMOS integrates multiple strategies to adjust word prior sentiment orientation while also considers the type of sentence. QMOS also employs the Semantic Sentiment Approach to determine the sentiment score of a word if it is not included in a sentiment lexicon.

On the other hand, the most of the existing methods fail to distinguish the meaning of a review sentence and user's query when both of them share the similar bag-of-words; hence there is often a conflict between the extracted opinionated sentences and users' needs. However, the summarization phase of QMOS is able to avoid extracting a review sentence whose similarity with the user's query is high but whose meaning is different. The method also employs the greedy algorithm and query expansion approach to reduce redundancy and bridge the lexical gaps for similar contexts that are expressed using different wording, respectively. Our experiment shows that the QMOS method can significantly improve the performance and make QMOS comparable to other existing methods.

1. Introduction

Nowadays, opinion mining, sentiment analysis, and subjectivity attracted significant attention from both the research community and industry (Khan, Qamar, & Bashir, 2016b). The main goal of sentiment analysis is to use an automated approach to identify positive, negative, or neutral sentiment from documents (Hung & Chen, 2016). Natural Language Processing (NLP) techniques allow people to automatically retrieve, extract, classify and summarize a huge amount of textual information. Text summarization is a process to create a short version of a source text, including important and essential information. There are several types of summary such as a single document, multiple documents, generic, query-based, opinion-based, etc. (Abdi, Idris, Alguliyev, & Aliguliyev,

* Corresponding author.
E-mail address: asadabdi55@gmail.com (A. Abdi).

https://doi.org/10.1016/j.ipm.2017.12.002
Received 26 August 2017; Received in revised form 2 November 2017; Accepted 8 December 2017
Available online 21 December 2017

2015b). Opinion/sentiment summarization is one of the new types. It aims to obtain significant information and the overall opinion/sentiment orientation of an opinionated document (Balahur, Kabadjov, Steinberger, Steinberger, & Montoyo, 2012). Surely, Opinion/sentiment summarization is one of the strong NLP methods. It can be considered as an expert system to assist human or process other tasks for selecting a logical choice or decision making (Lloret et al., 2015). Sentiment summarization includes set of process to determine opinion-oriented information from people's opinions on various subjects (Hu, Chen, & Chou, 2017). The main task of traditional summarization methods is to determine the important information and eliminate the redundant information (Abdi & Idris, 2014a,b; Abdi, Idris, Alguliev, & Aliguliyev, 2015a). Unlike a traditional summarization method, a sentiment summarization method relies on two main factors: sentiment degree and relevant information selection.

This paper proposes a Query based Multi-documents Opinion-oriented Summarization (QMOS) method. It aims to extract and summarize the opinionated sentences related to the user's query. The proposed method needs to perform two main phases: **Phase 1:** sentiment analysis, it includes the following steps; the first step calculates the sentiment score of each sentence. The second step recognizes the polarity of each sentence and user's query (e.g., *positive, negative, neutral*). The third step selects sentences with the same sentiment orientation of the opinion of the correlated user's question. Subsequently, it passes these sentences to the next phase, summarizer. **Phase 2:** Summarizer, it determines user's query relevant sentences using graph-based ranking model. It calculates the total sentence score including query relevant score and sentence sentiment score. This process filters out the sentences with less possibility to answer the question. It extracts the answers from the top ranked sentences. The sentences with higher rank are assumed to more possibly contain the answers.

Phase 1 — in the current phase, the sentiment analysis is performed based on a lexicon-based method. Thus, we combined several sentiment dictionaries in order to create a High-Coverage Lexical resource (HCL). Our aim is to expand the sentiment dictionary coverage in order to improve word coverage limit of the individual lexicon. On the other hand, different sentiment lexicons complement each other. Furthermore, phase 1 also employs Semantic Sentiment Approach (SSA) to determine sentiment score of a word if it is not included in a sentiment lexicon. Since the polarity of a word depends on the context in which the word is being used, the polarity of a word presented by a lexicon can be reversed. On the other hand, the performance of a sentiment analysis method relies on the types of sentences. Thus, to cope the aforementioned challenges, the QMOS considers multiple strategies such as *negation handling, but-clause handling and the subjective/objective handling, sarcastic sentence handling, interrogative/conditional sentences handling* to adjust word prior sentiment orientation and determine the type of a sentence, respectively.

Phase 2 — the summarizer phase takes as input a set of opinionated sentences from the phase 1. Subsequently, it processes all the sentences and generates a summary. This summary must represent the useful content and opinions to the user in a compressed form. The current phase relies on the graph-based model, where the nodes represent review sentences and user's query. The edge between two nodes shows the similarity measure between two sentences (*S* to *S*) or similarity measure between user's query and a sentence (*Q to S*) (*refer to section "3.3.Summarizer phase"*). Most of the previous studies on query based review summarizations used a bag-of-word (BOW) approach to identify query relevant sentences and calculate similarity measures. Although the BOW approach is very simple, it has some important drawbacks as it: *1)* disrupts the word order; *2)* breaks the syntactic structures; *3)* is not able to distinguish the meaning of two sentences. For an instant, given two sentences S_1 and S_2, (i.e., *S_1: 'Father helps son strongly'; S_2:' Son helps father strongly'*), the existing methods based on the BOW approach will infer that two sentences S_1 and S_2 are similar. Therefore, both the syntactic information and semantic information must be used when comparing two sentences (Abdi & Idris, 2014a, b; Abdi et al., 2015a; Abdi, Idris, Alguliyev, & Aliguliyev, 2015a). The summarizer phase also integrates both the semantic and syntactic information to capture the meaning of two sentences when they share similar bag-of-words (BOW). Furthermore, the QMOS uses the Content Word Expansion (CWE) method to *1)* improve the sentence ranking result; *2)* tackle the limit of information expressed in each sentence; *3)* overcome vocabulary mismatch problem in sentence comparison. It also bridges the lexical gaps for similar contexts that are expressed in a different wording. Finally, the current phase checks the redundancy information in order to increase the quality of a summary.

We make the following contributions in this research: *1)* we combine several sentiment dictionaries to improve word coverage limit of the individual lexicon; *2)* the proposed method uses the SSM method to determine the sentiment score of a word if it not included in sentiment lexicon; *3)* it also considers the contextual polarity for sentiment analysis (e.g., *negation, but-clause*); *4)* it considers the type of a sentence (e.g., *sarcasm, subjective/objective and interrogative/ conditional*); *5)* the method also integrates both the semantic and syntactic information using a linear equation to capture the meaning of two sentences; *6)* it uses the CWE method to overcome vocabulary mismatch problem in sentence comparison; *7)* it also checks the redundancy information in order to increase the quality of summary. Although some of previous systems consider the contextual polarity, the type of a sentence, the content word expansion, the combination of semantic and syntactic information or the redundancy removal to produce a query based multi-documents opinion-oriented summarization model, to the best of our knowledge, a method in which the contextual polarity, the type of a sentence, the content word expansion, the combination of semantic and syntactic information or the redundancy removal for a sentiment-oriented summarisation are used has not been thoroughly studied.

We also conduct two extensive experiments on data sets: Experiment 1 aims to evaluate the performances of the proposed method (*compared with the existing methods*) and analyze the effect of different parameters values (i.e., γ *and* β). In experiment 2 we analyze the effect of semantic similarity measure (*SSM*), word order similarity measure (*WOSM*), content word expansion (*CWE*) and semantic sentiment approach (*SSA*) and contextual polarity on QMOS method. We also compare different state-of-the-art sentiment dictionaries.

The rest of this paper is structured as follows. In Section 2 of this paper, we consider related work on sentiment analysis. Section 3 presents the QMOS method. We then summarize the experimental results in Section 4. Finally, we conclude this paper in Section 5.

2. Literature review

2.1. Text summarization

Text summarization systems produce a summary of one or more source text automatically. A summary normally contains the aim, results, conclusions presented in a source text (Abdi & Idris, 2014a,b; Tayal, Raghuwanshi, & Malik, 2017).

Categorization of Text Summarization Systems (Abdi et al., 2015b; Mendoza, Bonilla, Noguera, Cobos, & León, 2014)— the output of the system may be an abstractive or extractive summarization. An extractive summarization includes a set of significant sentences that are determined using statistical and linguistic features of sentences. An *abstractive summarization* tries to develop a comprehension of the main concepts in a text and then expose those concepts. A summarization system can be based on *single document or multiple documents*. In single document summarization system a single-document is used to generate a summary, while in multi-document summarization systems, multiple documents on the same subject are used to generate a single summary. Besides these facts, text summarization system can also be either indicative or informative summarization. The indicative summary only introduces the basic idea of a text to the user. Indicative summaries can be used to encourage the readers to read the main documents. The informative summary produces brief information of the main text that can be considered as a replacement for the original text. Summarization systems can create a general summary, where the system considers the total document. Unlike the general summary, the system can also produce a question-based summary, where the system tries to answer user's question.

2.2. Sentiment analysis and opinion mining

Sentiment analysis also named opinion mining aims to analyze the opinions expressed in a review document and to categorize the opinions as positive, negative or neutral (Rana & Cheah, 2016). Sentiment analysis is one of the new research topics in natural language processing domain. An opinion/sentiment includes two important parts, *E* and *S*. The *E* can be an entity or a feature of an entity and the *S* can be a positive/negative opinion, sentiment, orientation or polarity on *E*.

Different Levels of Analysis — the sentiment analysis can be performed at three main levels (Rana & Cheah, 2016):

1) Document-level: it classifies the whole document as positive, negative or neutral; *2) Sentence level:* the sentence level sentiment analysis module analyzes a sentence and determines whether the sentence has a positive, negative or neutral sentiment; *3) Aspect/feature level:* unlike document level and sentence level sentiment analysis, feature level analysis is able to determine what accurately people like and don't like. It is able to detect whether the tendency on aspect/feature is positive, negative or neutral. In the sentence, "*the color quality of this television is amazing",* the aspect/feature is *'color quality'* of the *'this television'* . The opinion on the *'color quality'* aspect/feature is positive.

There are different types of opinions: *1) regular opinion*, indicates to simply express an opinion (e.g., '*the book has been written well*'). A regular opinion can also be direct opinion or indirect opinion. Direct opinion expresses opinion directly on a feature or an entity, while indirect opinion expresses opinion indirectly on a feature or an entity; *2) Comparative opinion* (Jindal & Liu, 2006) indicates the difference between entities, products, etc. (e.g., '*Sony is better than Samsung*'). Furthermore, an opinion can be expressed explicitly or implicitly in a review text. The *explicit opinions* are a subjective expression that indicates regular or comparative sentiments, while *implicit opinions* are objective expressions that indicate regular or comparative sentiments. There are also two important concepts that are relevant to sentiment/opinion analysis:*1)* subjective sentence expresses subjective views and feelings (e.g., "*I like VAIO*"); *2)* objective sentence expresses factual information, no sentiment or opinion (e.g., "*VAIO is a SONY product*") (Riloff, Patwardhan, & Wiebe, 2006). Emotion is also related to sentiment analysis that presents our feeling and opinion. The researches categorized people's opinions into some groups such as, '*love*', '*joy*', '*surprise*', '*anger*', '*sadness*', and '*fear*' .

Sentiment analysis approaches are split into three groups: *1) dictionary based approach* (*unsupervised approach*) — in this approach, the sentiment/opinion words are used to express a positive/negative opinion/sentiment. As an example, the words like ('*excellent*', '*well*') and ('*poor*', '*ugly*') are used to make a positive and negative sentiment, respectively. However, a list of these words is named opinion/sentiment dictionary or lexicon (e.g., AFINN (Nielsen, 2011), Sentiment140 Lexicon (Mohammad, Kiritchenko, & Zhu, 2013)). Dictionary-based method employs a sentiment lexicon to perform sentiment analysis, using extracting opinion words and scoring sentiment related words; *2) machine learning approach* (*supervised approach*), machine Learning based method — it used a set of popular machine learning approach to perform sentiment analysis (e.g., ANN (Lee & Choeh, 2014), SVM (Shahana & Omman, 2015)). Machine learning method is able to handle large collections of review documents; *3) Hybrid approach* (*composition of supervised and unsupervised approach*).

2.3. Sentiment-based summaries

In recent years, due to the huge amount of reviews documents, a new type of summarization has been presented: sentiment summary. Sentiment summarization can be considered as a kind of *multi-document summarization*. However, a sentiment summarization is different from a traditional text summarization. Unlike text summarization (*short version of factual information*), opinion or sentiment summarization summarizes opinions/sentiments from a large number of reviewers or multiple reviews (Gambhir & Gupta, 2017). Consequently, text summarization and sentiment analysis must be composed in order to produce a sentiment summary.

Text summarization determines most relevant sentences from a reviews text and the sentiment analysis component identifies and categorizes objective or subjective sentences and their polarity (positive, negative or neutral), respectively (Liu, 2012).

Although there are several literatures on text summarization (e.g., *Sadh, Sahu, Srivastava, Sanyal, & Sanyal, 2012*) and sentiment analysis (e.g., *Mladenović, Mitrović, Krstev, & Vitas, 2016*), there is few work on the combination of two areas (Saggionα & Funk, 2010). The multi-text summarization method has been used for review summarization in various proposed methods as follows.

Lu, Duan, Wang, and Zhai (2010) proposed a method based on the online ontologies of entities and aspects to summarize opinions. Their system first selects aspects that capture major opinions. It orders aspects and their corresponding sentences based on a coherence measure, which tries to optimize the ordering so that they best follow the sequences of aspect appearances in their original postings.

In Nishikawa, Hasegawa, Matsuo, and Kikui (2010), a summarization technique was proposed, which generates a traditional text summary by selecting and ordering sentences taken from multiple reviews, considering both informativeness and readability of the final summary. The informativeness was defined as the sum of the frequency of each aspect-sentiment pair. Readability was defined as the natural sequence of sentences, which was measured as the sum of the connectivity of all adjacent sentences in the sequence.

Wang, Zhu, and Li (2013) also proposed a technique to summarize the user's reviews on Rice cooker (Collected from Amazon.com). Paul, Zhai, and Girju (2010) proposed an algorithm. The algorithm generates a macro multi-view summary and a micro multi-view summary. A macro multi-view summary contains multiple sets of sentences, each representing a different opinion. A micro multi-view summary contains a set of pairs of contrastive sentences. The algorithm includes two steps. In the first step, it uses a topic modeling approach to modeling and mining both topics and sentiments. In the second step, a random walk formulation was proposed to score sentences and pairs of sentences from opposite viewpoints based on both their representativeness and their contrastiveness with each other.

Lloret et al., (2015) proposed text summarization approach to produce a concise opinion summary of reviews (collected from both Amazon.com and WhatCar.com). Hu et al. (2017) used a multi-text summarization approach to summarize user's review on the hotel. In addition to text processing approaches (e.g., BOW, semantic approaches), the method also considered four main factors: '*author credibility*', '*review time*', '*review usefulness*' and '*conflicting opinions*' . At the end, the review summarization is generated.

Summing up, various methods have been proposed to summarize users' reviews. There are some problems with the most existing systems that we considered in this paper. We improved the existing systems as follows: *1)* QMOS combines several sentiment dictionaries to improve word coverage limit of the individual lexicon. *2)* QMOS employs the Semantic Sentiment Approach (SSA) to determine the sentiment score of a word if it is not included in a sentiment lexicon. *3)* QMOS also considers the contextual polarity for sentiment analysis. *4)* It considers the type of a sentence that can affect the performance of a sentiment analysis approach.

Furthermore, most of the previous studies on query based review summarizations focused on bag-of-word approach to identify query relevant sentences from a reviews text. They disregard the syntactic information in original review text; hence there is often a conflict between the retrieved sentences and users' needs. Therefore, we solve this problem using the combination of semantic and syntactic information to identify more query relevant sentences. On the other hand, a query consists of very few words. So, identifying important sentences to answer user's question using this little information can be considered as the main problem. However, QMOS employs content word expansion method to tackle the aforementioned problem.

3. Proposed method: QMOS

This section aims to present step-by-step an approach for Question-based Multi-documents Opinion-oriented Summarization. Fig. 1 displays the architecture of the QMOS. The method contains the following stages:

1. *Pre-processing*, in this stage the basic NLP techniques are performed on both review documents and user's question. The review text and user's query will be prepared for more process.
2. *Sentiment analysis*, the task of the current stage is to analyze review sentences (*the query is also considered as a sentence*) in order to identify set of subjective sentences with the same sentiment orientation of the opinion of the correlated user's question. Subsequently, we pass these sentences to the next stage, *Summarizer stage*.
3. *Summarizer*, this stage employs the graph-based ranking model to select more relevant sentences to the user's query. It also calculates the Total Sentence Score (TSS). Finally, the summary generation step creates the final summary.

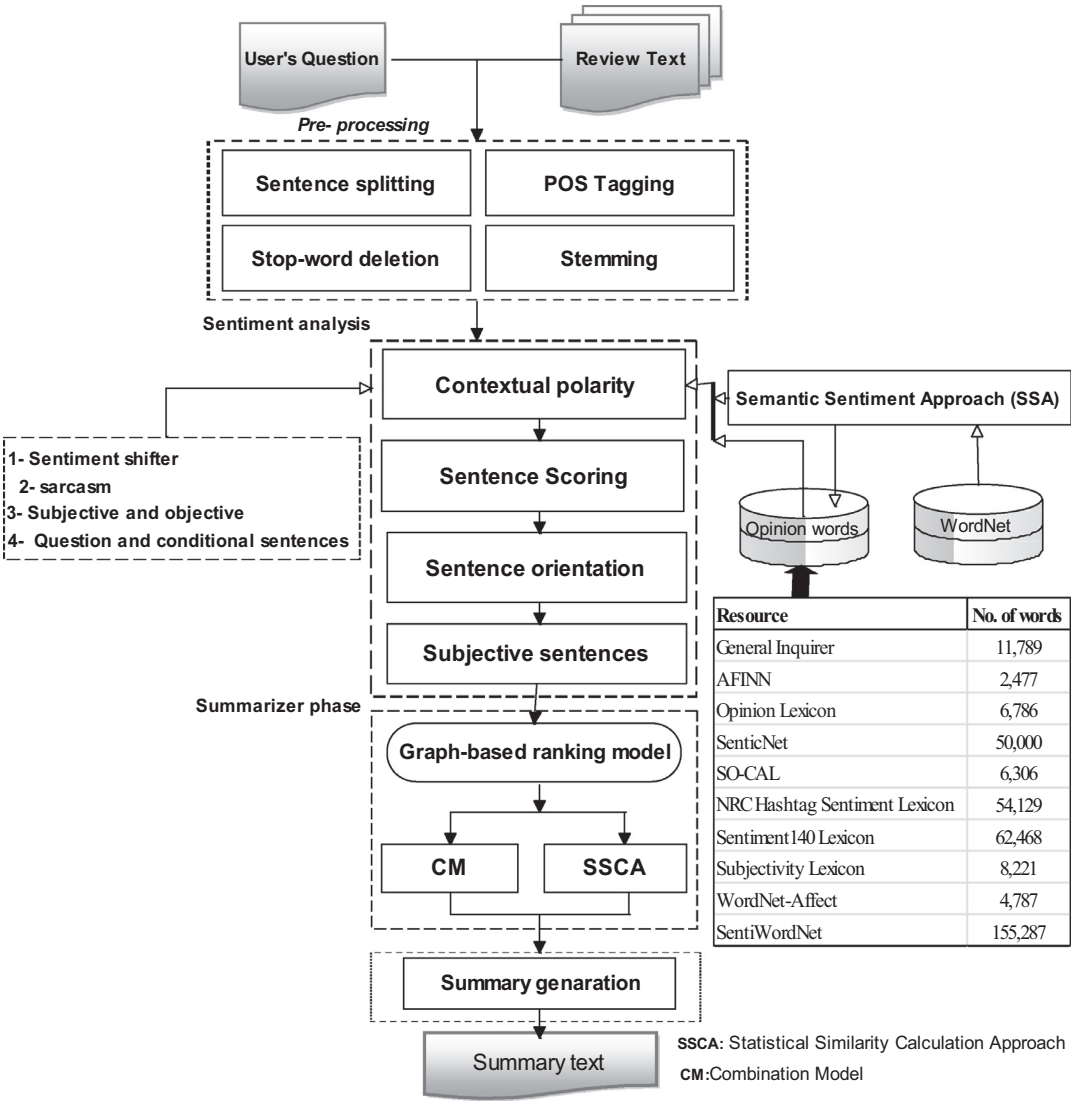
We describe each stage in the following sections. We also used the following list (Table 1) of abbreviations to explain our proposed method.

3.1. Pre-processing

The pre-processing principally contains the following functions: *1)* Sentence splitting; *2)* Stemming; *3)* Stop-word deletion; *4)* Part-of-speech (POS) tagging.

Sentence splitting — since the sentiment analysis is performed at the sentence level, this function split the review text into several sentences. A sentence ends with a sentence delimiter (".", "?", "!").

Stop word deletion — stop words indicate a set of words that, *1)* they are very common within a text and are also considered as noisy terms such as prepositions, articles, etc. (Kolchyna, Souza, Treleaven, & Aste, 2015). *2)* They don't affect the sentiment of the sentence (Kolchyna et al., 2015). *3)* They do not provide worth information in a sentence (Wang, Zhang, Sun, Yang, & Larson, 2015).



Resource	No. of words
General Inquirer	11,789
AFINN	2,477
Opinion Lexicon	6,786
SenticNet	50,000
SO-CAL	6,306
NRC Hashtag Sentiment Lexicon	54,129
Sentiment140 Lexicon	62,468
Subjectivity Lexicon	8,221
WordNet-Affect	4,787
SentiWordNet	155,287

Fig. 1. The Architecture of the QMOS.

Table 1
List of abbreviations.

List of abbreviations	
Word- Set (WS)	Sentiment Analysis Model (SAM)
Part-of-speech (POS)	Semantic Sentiment Approach (SSA)
Compound Model (CM)	Semantic Similarity Measurement (SSM)
Total Sentence Score (TSS)	High-Coverage Sentiment Lexicon (HCL)
Average ROUGE Score (ARS)	Document Understanding Conference (DUC)
Sentence Sentiment Score (SSS)	Word-Order Similarity Measurement (WOSM)
Text Analysis Conference (TAC)	Statistical Similarity Calculation Approach (SSCA)
Content Word Expansion (CWE)	Semantic similarity measure between two words (SWS)
	Sentence similarity measure using SSM and WOSM (SSWOS)

Table 2
Sample of the Stop words.

Stop words							
By	*Often*	*Vol*	*The*	*Of*	*Before*	*Under*	*Until*
whatever	*Nothing*	*Both*	*Beyond*	*Between*	*Besides*	*Usually*	*With*
Voz	*Ok*	*Okay*	*Vol*	*Ord*	*Onto*	*Did*	*Others*
From	*Yet*	*Yes*	*Upon*	*Off*	*Even*	*Own*	*By*

Table 3
An overview of ten lexical resources.

Sentiment lexicons	No. of words	Classifying	Score	(POS)	
				No	Yes
General Inquirer (Stone & Hunt, 1963)	11,789	*'Positiv', 'Negativ', 'Pstv', 'Pstv', 'EMOT', etc.*	Nil	√	
AFINN (Nielsen, 2011)	2477	Nil	[−5, +5]	√	
Opinion Lexicon (M. Hu & Liu, 2004)	6786	*'Positive', 'Negative'*	Nil	√	
SenticNet4 (Cambria et al., 2016)	50,000	*'Positive', 'Negative'*	[−1, +1]	√	
SO−CAL (Taboada et al., 2011)	6306	Nil	[−5, +5]	√	
NRC Hashtag Sentiment Lexicon (Mohammad et al., 2013)	54,129	*'Positive', 'Negative'*	[−7, +7]	√	
Sentiment140 Lexicon (Mohammad et al., 2013)	62,468	*'Positive', 'Negative'*	[−7, +7]	√	
Subjectivity Lexicon (Riloff & Wiebe, 2003)	8221	*'Positive', 'negative', 'neutral'*	Nil		√
WordNet-Affect (Strapparava & Valitutti, 2004)	words:4787 synsets:2874	The synset are first grouped into *'behaviour', 'situation', 'trait', etc.* and these groups are classified into *'positive', 'negative', 'ambiguous', 'neutral'*	Nil		√
SentiWordNet (Baccianella et al., 2010)	words:155,287 synsets:117,659	*'Positive', 'negative', 'objective'*	[0, 1]		√

Table 2 displays a sample of the stop words[1,2,3]. It is worth nothing that we excluded a set of words as explained in section ('*3.2.3. Contextual polarity for sentiment analysis, Sentiment shifter*').

Stemming – it aims to get the stem or root of a word. It is useful to identify words that belong to the same stem. This process obtains the root of each word using the WordNet lexical database (Miller & 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.

Part-of-speech (POS) tagging — the POS tagging allows to automatically tag each word its morphological category (e.g., "*Teacher/NNP helps/VBZ her/PRP$ students/NNS*"). We used an English part-of-speech tagger which was developed by Tsuruoka and Tsujii (2005) University of Tokyo.

3.2. Sentiment analysis

The pipeline of the sentiment analysis model (SAM), which goes from identifying opinion sentences to sentence orientation, is shown in Fig. 1. The SAM performs the following main steps:

1. Taking all sentences from the review text as input. Let $Sentenc_{review\ text} = \{S_1, S_2 \cdots S_N\}$ includes all sentences from the review text, where N is the number of sentences;
2. Determining type of sentences and contextual polarity;
3. Computing sentence sentiment score;
4. Determining sentence orientation.

3.2.1. Sentiment lexicons combination

In this section, we aim to create High-Coverage sentiment Lexicon, HCL. We merge several existing sentiment lexicons with different size and format. We also employ the semantic sentiment method (SSM) in order to expand the sentiment dictionary coverage and improve the individual dictionary limited word.

Many sentiment dictionaries (e.g., *SentiWordNet (Baccianella, Esuli, & Sebastiani, 2010), Micro-WNOp (Cerini, Compagnoni, Demontis, Formentelli, & Gandini, 2007), WordNet-Affect (Strapparava & Valitutti, 2004)*) have been manually or automatically

[1] http://dev.mysql.com/doc/refman/5.5/en/fulltext-stopwords.html.
[2] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop.
[3] http://www.cs.cmu.edu/~mccallum/bow/rainbow/.

constructed to classify positive and negative opinions in a text. We employed a number of sentiment dictionaries. An overview of the most commonly used sentiment lexicons is presented in Table 3. As shown in Table 3, some of the lexicons include sentiment scores with various numerical ranges. Furthermore, some of them categorized sentiment words into positive, negative and neutral while some of them classified into types of emotions (e.g., *"bad", "joy", "happy", "sadness"*). Since these lexicons have a different format, we standardize them to have one of the sentiment values 1, 0, −1. The processes of the standardization are explained as follows. It is worth noting, the sentiment score of each word in the combined dictionary, HCL, is calculated using the averaging the sentiment values of the overlapping words.

The processes of merging the sentiment dictionaries are the following steps:

General Inquirer (GI) (Stone & Hunt, 1963) — sentiment word of GI have been classified into more than 180 groups. Therefore, we consider *'positiv', 'affil', 'strong', 'active', 'doctrin', 'pstv', 'virtue', 'PosAff'* and *'yes'* in this classification as positive words and assigned them a sentiment score of (+1). We also considered *'negative',' ngtv', 'week', 'fail', 'passive', 'decrease', 'finish', 'no', 'negaff'* in this classification as positive words and assigned them a sentiment score of (−1).
AFINN (Nielsen, 2011) — we normalize the sentiment score from [−5, +5] to [−1, +1].
Opinion Lexicon (M. Hu & Liu, 2004) — the sentiment words in opinion lexicon have been categorized into positive and negative words. Therefore, we assigned sentiment value of +1to positive words. We also assigned sentiment value of −1 to negative words. Finally, we dedicated sentiment score of 0 to words which appear in both positive and negative categories.
SenticNet4 (Cambria, Poria, Bajpai, & Schuller, 2016) — in this dictionary, a sentiment value (within a range of [−1, +1]) has been assigned to each sentiment word. However, we employed the sentiment score of each word.
SentiWordNet (Baccianella et al., 2010) — we used the following equations to calculate the sentiment value for each word within the range of [−1, 1] (Khan, Qamar, & Bashir, 2016a).

$$Sentiscore = (Posscore - Negscore) \tag{1}$$

Where, *posscore* and *negscore* are positive sentiment score and negative sentiment score of each word, respectively. If *senti_score* > 0 the sentiment word orientation is positive. If *senti_score* < 0 the sentiment word orientation is negative. Finally, If *senti_score* = 0 the sentiment word objective.
SO−CAL (Taboada, Brooke, Tofiloski, Voll, & Stede, 2011) — the sentiment value of each sentiment word is normalized from [−5, +5] to [−1, +1].
Subjectivity Lexicon (Riloff & Wiebe, 2003) — words have been categorized into positive, negative, both and neutral words. These categories: positive, negative, both and neutral are considered +1, −1, 0 and 0 respectively.
WordNet-Affect (Strapparava & Valitutti, 2004) — there are various classification (e.g., *'positive-emotion', 'negative- emotion', 'ambiguous-emotion', and 'neutral- Emotion'*) to categorize each word. Each category is assigned +1, −1, 0 and 0 respectively.
NRC Hashtag Sentiment Lexicon and Sentiment140 Lexicon (Mohammad et al., 2013) — words are normalized from [−7, +7] to [−1, +1]. A positive value illustrates a positive orientation. A negative value illustrates a negative orientation.

3.2.2. Semantic sentiment approach (SSA)

One of the main restrictions of sentiment dictionaries, as mentioned above, they are limited by their sentiment words. In other words, a sentiment word will be discarded if it is not included in the dictionary. Therefore, to tackle the aforementioned problem and the lexical gaps, we employ the SSA approach to determine sentiment score of a word if it not included in HCL. In this approach, we considered some specific POS (i.e. *Noun, Adjective, Adverb, and verb*).

Given a word (*W*), Let $WS = \{W_1, W_2 \cdots W_N\}$ denote the synonymous words that are collected using WordNet. In the second step, by a loop for each word *W* of *WS,* it undertakes certain tasks: *1)* it checks if the word appears in the HCL, a) the sentiment score is obtained; b) if the sentiment score is equal to 1, add (+1) unit to the positive sentiment score (Pos_{sw}); c) if sentiment score is equal to (−1), add (−1) unit to the negative sentiment score (Neg_{sw}). Finally the total sentiment value of the word *W* is calculated using the Eq. (2).

$$S_{sw} = \frac{1}{n} \times \sum Pos_{sw} - \frac{1}{m} \times \sum Neg_{sw} \tag{2}$$

Where, *n* and *m* are number of positive and negative words respectively.

3.2.3. Contextual polarity for sentiment analysis

Usually, a dictionary-based approach used pre-defined sentiment score to determine the overall sentiment orientation of a text. However, the pre-defined sentiment score may affect the performance of a dictionary-based approach. This is due to the fact that, the polarity of a word presented by a sentiment lexicon can be reversed since the polarity of a word (positivity or negativity) depends on the context in which it appears.

As an example, in the sentence '*The bed is not well*', the polarity of the word '*well*' is positive while the polarity of the whole sentence is negative because of the negation word 'not' (Chen, Xu, He, & Wang, 2017). On the other hand, since the type of a sentence also affects the performance of sentiment analysis approach, we also considered the various types of sentences in sentiment analysis. There are different types of sentences (e.g., *subjective sentences, comparative sentences, conditional/question sentences, sarcastic sentences, objective sentence*) that can be used for sentiment analysis (Chen et al., 2017). In our work, we considered the subjective/ objective

Table 4
Sample of interjection words.

Interjection words					
Wow	*Aha*	*Bah*	*New*	*Yay*	*Uh*
Wah	*Achoo*	*Ack*	*Eek*	*Duh*	*Doh*
Eh	*Alas*	*Gee*	*Uggh*	*Woops*	*Tut*
Feh	*Huh*	*Hup*	*Hurrah*	*Oh*	*ouch*

sentence handling, question/ conditional sentences handling, sarcastic sentence handling and sentiment shifter (e.g. *negation handling, but-clause handling*) for sentiment analysis.

Subjective and objective sentence — a subjective sentence includes a sentiment word (i.e., *'good', bad', 'excellent', 'poor'*) or expresses an opinion, while an objective sentence does express an opinion or expresses factual information (Chen et al., 2017).

Question and conditional sentences — a review sentence including a sentiment word may not present any opinion. Question and conditional sentences can be considered as this type of sentences (Narayanan, Liu, & Choudhary, 2009). A question word, "*may you tell us which Samsung TV is good?*", "*If I can find a good TV in the shop, I will buy it*" and "*is your car in a good condition?"* . All sentences include sentiment words (e.g., "*good*"), but they do not express a positive or negative opinion on TV. However, all conditional and question sentences do not express opinion or sentiments (Liu, 2012).

Sarcasm handling — it can also be considered as one of the challenges in sentiment analysis. Sarcasm detection automatically is very difficult, because the lexical features do not provide enough information to detect it (Bharti, Babu, & Jena, 2015). It is used to dispraise and mock (Lunando & Purwarianti, 2013). Unlike the negation, a sarcastic sentence presents a negative sentiment using positive words (Povoda, Burget, & Dutta, 2016). In other words, the surface of a sarcastic sentence is a positive opinion but the meaning is negative opinion (Liu, 2012). As an example: the sentence, "*I love waiting forever for the doctor*", presents a positive opinion (*love*), but the overall sentence expresses a negative opinion.

The QMOS detects a sarcastic sentence based on the interjection words and a set of following heuristic rules proposed by Bharti et al. (2015). If an interjection word, Table 4, appears in a sentence, the sentence tends to be a sarcastic sentence (Bharti et al., 2015; Lunando & Purwarianti, 2013).

Heuristic rules to identify sarcasm:

1) *Interjection words + (adjective OR adverb).*
2) *Interjection words + adjective + adverb.*
3) *Interjection words + adverb + adjective.*
4) *Interjection words + adjective + noun.*
5) *Interjection words + adverb + verb.*

Sentiment shifter — sentiment shifter includes a set of words that change the sentiment orientation of a sentence (Xia, Xu, Yu, Qi, & Cambria, 2016). A sentiment shifter contains negations, but-clause (*contrasts*), etc. Due to the result of research S. Li, Lee, Chen, Huang, and Zhou (2010) the negations and the but-clause (*contrasts*) cover more than 60 percentage structures sentiment shift.

1. *Negation handling.* A negation includes some special words, Table 5, that can change the sentiment of a sentence from negative to positive and vice versa. However, we can detect a negation sentence using a set of pre-defined negation words: if a negation word appears in a sentence, the polarity of a sentiment word will be changed. Usually, the sentiment word appears between the negation word and the punctuation mark ('.', ',', '!', '?', ':', ';'). Given the sentence "*He does not like red car*", the negation word, "*does not"* change the sentiment orientation of the word "*Like*".
   It is worth nothing, we don't consider a negation word that it is a part of phrase such as, "*not only*", "*not wholly*", "*not all*", "*not just*", "*not quite*", "*not least",* "no question", "not to mention" and "no wonder".
2. *But-clause handling.* It contains some words like "*but*", "*with the exception of*", "*except that*", "*except for*", "*however*", "*yet*", "*unfortunately*", "*thought*", "*although*" and "*nevertheless*". These words usually change the sentiment orientation of the sentence following them. In other words, the sentiment orientations before the contrary word (e.g., but) and after the contrary word are opposite to each other (Liu, 2012). As an example, given the sentence "*I don't like this laptop, but the CPU speed is high*". The but-clause changes the sentiment orientation of the previous phrase "*I don't like this laptop*". However, the polarity of a sentence

Table 5
Sample of Negation words (Kolchyna et al., 2015; Li et al., 2010).

Negation words								
no	*not*	*Don't*	*hardly*	*Can not*	*None*	*Never*	*Nothing*	*nowhere*
are not	*Was not*	*Did not*	*lacking*	*Would not*	*Nobody*	*Nothing*	*Nowhere*	*Cant*
Were not	*Have not*	*neither*	*nor*	*without*	*Seldom*	*Wont*	*Couldn't*	*Doesn't*
Does not	*Should not*	*lack*	*Had not*	*Lacks*	*Nothing*	*Isn't*	…	…

can be set as follows:" "*I don't like [−1] this laptop, but the CPU speed is high [+1]*".

3.2.4. Sentence sentiment score calculation

Let $QRS = \{S_1, S_2 \cdots S_n\}$ indicate all sentences from the review text, where n is the number of sentences. The $Sss = \{(S_1, Sentence_{score}), (S_2, Sentence_{score}) \cdots (S_n, Sentence_{score})\}$ refers to sentence sentiment score, where S_i is a sentence and $Sentence_{score}$ is a corresponding sentiment score. For each sentence of *QRS* the following tasks are performed.

In the first step, the system checks whether the sentence S_i is an interrogative (Q) or conditional (C) sentence. If the S_i is a Q/C sentence: *i)* the sentiment score of sentence S_i equals 0; *ii)* the S_i and its sentiment score are added to *Sss; iii)* the process will be performed by the next sentence of *QRS*.

If the S_i is not a Q/C sentence, the method by a loop for each word of sentence S_i that belongs to the word class (i.e., "*noun*", "*verb*", "*adjective*", " *adverb*"), performs the following task:

i. Each word (*W*) is looked up in sentiment lexicon, HCL; if the word appears in HCL, its sentiment sore is obtained; finally, the pair of sentiment score and word is added to an array. Let $WSC = \{(W_1, value), (W_2, value) \cdots (W_n, value)\}$ indicate the sentiment word and its sentiment score.
ii. If the word is not included in the HCL, the method used the SSA to determine the sentiment score of the current word. If the SSA returned any value, the pair of sentiment score and word is added to *WSC*. If the SSA did not return a value, the method continues the process by the next word.

In the second step, the method checks if the current sentence includes a sentiment word, it considers the sentence as a subjective sentence and then, *i)* calculates the sentiment score using the third step; *ii)* the S_i and its sentiment score are added to *Sss; ii)* the process will be performed by the next sentence of *QRS*.

If the current sentence does not include a sentiment word, the sentence is an objective sentence, and then *i)* the sentence sentiment score equals 0. Then, *ii)* the S_i and its sentiment score are added to *Sss; c)* the process will be performed by the next sentence of *QRS*.

In the third step, the method checks negation handling, but-clause handling, and sarcasm handling. It calculates the sentiment score of the current sentence using the Eq. (3). Finally, the current sentence and its sentiment score are added to *Sss*.

$$Sentence_{score}(S) = \frac{1}{K} \times \sum_{i=1}^{k} Score\ (W_i) \tag{3}$$

Where, k is the number of the sentiment word in a sentence, S_i. $Score\,(W_i)$ indicates the sentiment score of word, W_i.

The aforementioned tasks are performed for each sentence of *QRS*.

3.2.5. Sentence orientation

Given the ***Sss***, the method determines the orientation of each sentence (*user's question also is considered as a sentence*). Therefore, a sentence is positive, if its sentiment score is above 0; a sentence is negative if its sentiment score is below 0; a sentence is neutral if its sentiment score is zero. Let $POS_{sentence} = \{S_1, S_2 \cdots S_p\}$ and $Neg_{sentence} = \{S_1, S_2 \cdots S_q\}$ indicate positive sentences and negative sentences, respectively. The sentence polarity is defined as following:

$$Sentence\ polarity\ (S) = \begin{cases} Positive\ \ if\ \ Sentence_{score}(S) > 0 \\ Negative\ \ if\ \ Sentence_{score}(S) < 0 \\ Neutral\ \ if\ \ Sentence_{score}(S) = 0 \end{cases} \tag{4}$$

Finally, the sentiment analysis phase sends the $SUB_{sentences}$, the $Neg_{sentence}$, the $POS_{sentence}$ and Sss to the next phase, summarizer phase. It is worth noting, the $SUB_{sentences}$, $Neg_{sentence}$ and $POS_{sentence}$ include sentences with the same sentiment orientation of the opinion of the correlated user's question.

3.3. Summarizer phase

The current phase produces a summary using the sentences that are relevant to the user's query and also expressed opinions. To do this, the summarizer phase performs the following tasks.

3.3.1. Graph-based ranking model

The graph model is created as follows. The nodes on the graph represent the review sentences and user's query. There are also two kinds of edges: *a) edge 1:* similarity measure between two sentences (*S to S*); *b) edge 2:* similarity measure between user's query and a sentence (*Q to S*). The similarity measures of (*S to S*) and (*Q to S*) as weight, are assigned to edge 1 and edge 2, respectively. The score of each node, sentence, is computed by the combination of similarity measures (*S to S*) and (*Q to S*). As an example, a review sentences graph based on the user's question and review sentences is shown in Fig. 2. Let $SUB_{sentences} = \{S_1, S_2 \cdots S_m\}$ includes all subjective sentences from the review text, where m is the number of sentences, $m \leq N$. Let S_j denotes a review sentence and Q represent user's query. Firstly, the model calculates the similarity measure between sentences S_j and each sentence of $SUB_{sentences}$ (e.g., $Sim\ (S_j, S_i)$, where $S_i \in SUB_{sentences}$; $i \leq m$). Secondly, it calculates the similarity measure between user's question and sentence, S_j (e.g., *Sim*

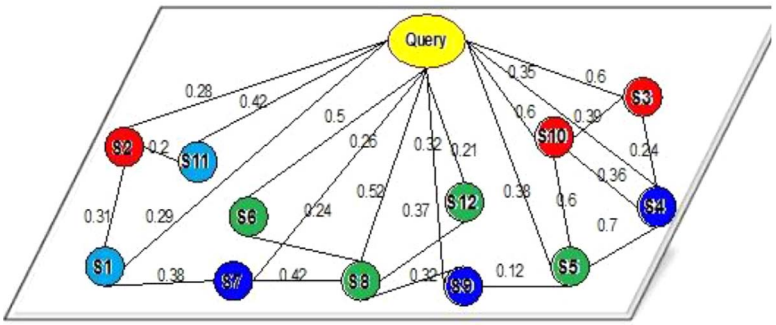


Fig. 2. Graph-based ranking model.

(S_j,Q)). Finally, the similarity score between sentence S_j and Q is calculated using Eq. (9).

3.3.2. Statistical similarity calculation approach (SSCA)

The SSCA computes the similarity measures of (*S to S*) and (*Q to S*). The overall task of SSCA is to compute similarity measure using the combination semantic and syntactic information are as follows. It is worth noting the query is also considered as a sentence.

1. Take two sentences S_1 and S_2;
2. Create a word set using S_1 and S_2; */* section a) The Word- Set (WS) */*
3. Create a semantic-vector for S_1 and S_2, separately; */* section c) Semantic Similarity Measurement (SSM)*/*
4. Create a word-order vector for S_1 and S_2, separately; */* section d) Word-order Similarity Measurement (WOSM)*/*
5. Compute the semantic similarity measure (*SSM*) between S_1 and S_2; */* using Eq. (6)*/*
6. Compute the word-order similarity measure (*WOSM*) between S_1 and S_2; */* using Eq. (7)*/*
7. Combine *SSM* and *WOSM* as a final similarity score (*FSS*) using a linear equation; */* using Eq. (8)*/*
8. The FSS is assigned to the edge between S_1 *and* S_2.

We explain each task in the following sections.

a) The Word- Set (WS)

Let $S_1 = \{W_1, W_2, \cdots W_K\}$ and $S_2 = \{W_1, W_2, \cdots W_L\}$ be two sentences. Let $WS = \{W_1, W_2, ..., W_N\}$ be a '*word-set*', where N the number of distinct words from sentences S_1 and S_2. The *WS* is created using the following steps:

1. Take two sentences, S_1 and S_2; $WS = \phi$
2. For each word (*W)* From sentence S_1,
   2.1. If the $W \in WS$ Then, continue step 2 by next word; otherwise, go to step 2.2;
   2.2. If the $W \notin WS$ Then, Add W to *WS*; jump to step 2;
3. Repeat steps (2- 2.2) for sentence S_2.

The corresponding process is shown in Algorithm 1.

b) Content Word Expansion (CWE) method

CWE method is generally used to solve the fundamental problem of word mismatch in the comparison between user's question and sentences, and information limit. The CWE method is based on semantic word similarity. The semantic similarity measure between two words (SWS) is calculated using the Eq. (5).

Algorithm 1
The creation of "word-set".

Input: Sentence 1, Sentence 2;
Output: $WS = \{W_1, W_2 \ldots W_n\}$, *WS* denotes an array that includes all distinct words from two sentences;
1: Let *W* be a word of the Sentence 1 or Sentence 2;
2: Let *RW* be the root of word *W*, it is obtained using WordNet;
3: Let *L* be the length of Sentence1 or Sentence2;
4: Set $l = 0$;
5: For each *W*,
   i. l = l + 1;
   ii. Get *RW*;
   iii. Look for *RW* in word set;
   iv. If the *RW* was not in *WS*, then assign *RW* to *WS*, Jump to step 6; otherwise, jump to step 6;
6: Jump to step 5; iterate until $l \leq L$;

Algorithm 2
semantic-vector.

Input: sentence 1, sentence 2,"word-set";
Output: semantic vector;
1: Let S be either senteence1 or sentence2;
2: Let W_t be a word of the word set;
3: Let RW be the root of the word W_t, it is obtained using the WordNet;
4: Let W be a word of S;
5: Let SWS denotes the semantic similarity measure between two words;
6: Let L be the length of S;
7: Set $l = 0$;
8: For each W_t,
  i. l = l + 1;
  ii. Get RW;
  iii. Look for RW in S;
  iv. If RW was in S, then set the corresponding element in semantic vector to "1";
  v. Otherwise,
    a. For each W,
      1. $SWS\ (W, W_t)$ is calculated using Eq. (5);
      2. If $SWS > 0$, Then Add SWS to array1; /* array1indicates similarity value between two words*/
      3. Iterate until $l \le L$;
    b. If array1=Null, then jump to step 9; otherwise,
    c. Select the most similarity value from array1;
    d. Set the corresponding element of the vector to the most value of similarity measure; set $l = 0$; jump to step8;
9: Assign '0" to the corresponding element of the vector; jump to step8; iterate until $l \le L$;

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

$$\mathrm{Sim}_{\mathrm{Dice}}(w_1, w_2) = \begin{cases} \frac{2\cdot|syns(w_1)\cap\ syns(w_2)|}{|syns(w_1)|+|syns(w_2)|}\ if\ \ \mathrm{w_1} \neq \mathrm{w_2} \\ 1\ if\ \ \mathrm{w_1} = \mathrm{w_2} \end{cases} \tag{5}$$

Where, $syns(w)$ is the set of words (synonyms) based on the WordNet. $|syns(w)|$ represents the cardinality of the set $syns(w)$.

c) Semantic Similarity Measurement (SSM)

The current section introduces a popular measure to calculate semantic similarity measure.

The Jaccard measure (Jaccard, 1912) — the Jaccard measure calculates semantic similarity between S_1 and S_2 using the following steps:

1. Create the semantic-vector for both sentences S_1 and S_2 using Algorithm 2;
2. Calculate the semantic similarity between two sentences using Eq. (6).

$$\mathrm{Sim}_{\mathrm{Jaccard}}\ (S_1, S_2) = \frac{\sum_{j=1}^{m}(w_{1j}\cdot w_{2j})}{\sum_{j=1}^{m} w_{1j}^2 + \sum_{j=1}^{m} w_{2j}^2 - \sum_{j=1}^{m}(w_{1j}\cdot w_{2j})} \tag{6}$$

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.

d) Word-order Similarity Measurement (WOSM)

WOSM employed syntactic-vector approach (Li, McLean, Bandar, O'shea, & Crockett, 2006) to compute word-order similarity. The WOSM is calculated using the following steps:

1. Create the syntactic-vector for both sentences S1 and S2 using Algorithm 3;
2. Calculate the word order similarity measure between two sentences using Eq. (7).

$$Sim_{\mathrm{wosm}}(S_q, S_x) = 1 - \frac{||\boldsymbol{O}_1 - \boldsymbol{O}_2||}{||\boldsymbol{O}_1 + \boldsymbol{O}_2||} \tag{7}$$

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

e) Sentence similarity measure using SSM and WOSM (SSWOS)

As analyzed in Pérez et al. (2005), the strength of semantic measure and syntactic measure can complete each other. Therefore, SSWOS integrates semantic and syntactic information in the comparison between two sentences using the following linear equation.

Algorithm 3
Word order Vector.

Input: sentence1, sentence2,"word set";
Output: Lexical vector;
1: Let S be either senteence1 or sentence2;
2: Let W_t be a word of the word set;
3: Let RW be the root of the word W_t, it is obtained using the WordNet;
4: Let W be a word of S;
5: Let SWS denotes the semantic similarity measure;
6: Let L be the length of S;
7: Set $l = 0$;
8: For each W_t,
  i. $l = l + 1$;
  ii. Get RW;
  iii. Look for RW in S;
  iv. If RW was in Sen, then set the corresponding element in vector to index position of word in S;
  v. Otherwise,
    a. For each W,
      1. $SWS\ (W,W_t)$ is calculated using Eq. (5);
      2. If $SWS > 0$, Then assign SSM to array1; /* array1indicates similarity value between two words*/
      3. Iterate until $l \le L$;
    b. If array1=Null, then jump to step 9; otherwise,
    c. Select the most similarity score from array1;
    d. Set the corresponding element of vector to index position of word with the most similarity score; set $l = 0$; jump to step8;
9: Assign '**0**" to the corresponding element of the vector; jump to step 8; iterate until $l \le L$;

$$\mathrm{Sim}_{\mathrm{sswos}}(\mathrm{S}_1, \mathrm{S}_2) = \beta \cdot Sim_{SSM}\ (S_1, S_2) + (1 - \beta) \cdot Sim_{\mathrm{WOSM}}(S_1, S_2) \tag{8}$$

Where $\beta \in (0, 1)$ is a parameter to create effective use of both Sim_{SSM} and Sim_{WOSM}.

3.3.3. Compound model (CM)

The CM includes two main steps: *1)* identifying query relevant sentences; *2)* total sentence score computing.

Identifying query relevant sentences — in the first step, the CM identifies query relevant review sentences using Eq. (9) (Badrinath, Venkatasubramaniyan, & Madhavan, 2011; Chali, Hasan, & Joty, 2011) to identify query relevant sentences. Let $P(s|q)$ indicate the similarity measure of a sentence S given a query Q. $P(s|q)$ is calculated using the sum of the similarity measure (S to Q) and the similarity measure (S to other sentences in review text), as shown in following equation.

$$P(s|q) = \gamma\ \times \frac{Sim(s,\ q)}{\sum_{l \in T} Sim(l,\ q)} + (1 - \gamma)\ \times \sum_{d \in T} \frac{Sim(s,\ d)}{\sum_{l \in T} Sim(l,\ d)} \times P(d|q) \tag{9}$$

Where, T includes all sentences in the review text. $0 < \gamma < 1$ is the weighting parameter, to determine the relative contribution of two similarities. Following (Erkan & Radev, 2004; Otterbacher, Erkan, & Radev, 2005), Eq. (9) can be considered in matrix form as follows:

$$\begin{cases} P_{(k+1)} = M^T P_k \\ M = \beta U + (1 - \beta) W \end{cases} \tag{10}$$

Where, W, U and M are square matrices. Let matrix W indicate the similarity measure (S to other sentences. Let U indicate the similarity measure (S to Q). K indicates the k^{th} iteration. $0 < \gamma < 1$ is the weighting parameter. The vector $P = [p_1, ..., p_N]$ corresponds to the stationary distribution of the matrix M. The combination model based on Eq. (10) follows the following tasks.

1. Make the square matrix W, where $W_{i,j} = Sim(\mathrm{S_i},\ \mathrm{S_j})$ is calculated using Eq. (8).
2. Make the square matrix U, where $U_{ij} = Sim(\mathrm{S_i},\ \mathrm{q})$ s calculated using Eq. (8).
3. Repeat $P_{(k+1)} = [\gamma U + (1 - \gamma)W]^T P_k$ 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 $\left[\frac{1}{N}, \frac{1}{N}, ..., \frac{1}{N}\right]$.
4. Let P denote the result. Each sentence $\mathrm{S_i}$ obtains its ranking score corresponding to p_i $(1 \le i \le N)$.

Total sentence score computing — in the current step, the Total Sentence Score (TSS) is calculated using the following equation:

$$TSS\ (S) = P(s|q)\ \times SPS\ (S) \tag{11}$$

Where, $P(s|q)$ is calculated using Eq. (9). SPS indicates sentence polarity score, where $SPS(S) = Sentence_{score}(S)$ if the sentence is positive. $SPS(S) = Sentence_{score}(S)\ \times\ (-1)$ if the sentence is negative.

3.3.4. Summary generation: sentence selection and redundancy removal

The final summary includes the opinionated sentences that are relevant to the user's query. Since the opinionated sentences are

Question	Why do people dislike George Clooney?
Sentence1,(S_1)	George Clooney is a bad actor.
Sentence2,(S_2)	George Clooney is a cute actor.
POS tagging(S_1)	George/NNP Clooney/NNP is/VBZ a/DT bad/JJ actor/NN./.
POS tagging(S_2)	George/NNP Clooney/NNP is/VBZ a/DT cute/JJ actor/NN./.
POS tagging(Q)	Why/WRB do/VBP people/NNS dislike/IN George/NNP Clooney/NNP? /.
Stop-word removal(S_1)	George Clooney bad actor.
Stop-word removal(S_2)	George Clooney cute actor.
Stemming(S_1)	George Clooney bad actor.
Stemming(S_2)	George Clooney cute actor.
Sentiment score using Eq.(3)	S_1 = -0.36; S_2 = +0.53; Q = -0.91
Polarity(S_1,S_2,Q) using Eq.(4)	S_1 = Neg; S_2 = Pos; Q = Neg
***Sentences with the same sentiment orientation of the opinion of the (Q)**	S_1
***Total Sentence Score (TSS) using Eq.(11)**	0.0982
We repeat the aforementioned processes (*) for all sentences.	
Summary generation	Follow the section "*3.3.4. summary generation*"

Fig. 3. Example of brief process of QMOS.

extracted from several documents, the sentences may include the same content. Therefore, the method must consider the redundancy problem. It considers two main tasks to cope the aforementioned problem. First, each review sentence score is calculated using the Eq. (11). Second, before adding a review sentence to concise summary text, each review sentence is compared to other sentences (Eq. (12)) and the sentence that is not more similar to other candidate sentences is included in the concise summary text. The method used the greedy algorithm (Wan, Yang, & Xiao, 2007; Zhang et al., 2005) to eliminate redundancy. The greedy algorithm contains 5 main steps as follows:

1. Let $A_1 = \varnothing$ and $A_2 = \{S_i | i = 1, 2...,N\}$, where A_1 is a null set and A_2 indicates the score of each sentence computed using Eq. (11).
2. A_2 is sorted based on their sentence scores in descending order.
3. A sentence, S_i, with high score, is moved from A_2 to A_1.
4. Then again, compute the scores of all sentences in A_2 by taking into account the redundancy penalty (Eq. (12)). However, for each sentence $S_j \in A_2$,

$$Score(S_j) = Score(S_j) - (Sim(S_i, S_j) \times TSS\ (S_i)) \quad (12)$$

   Where, $Sim(s_i, s_j)$ is computed using Eq. (8).
5. The steps (2)–(4) are repeated until $A_2 = \varnothing$ or the number of sentences in final summary has been satisfied. As the result, the set A_1 is considered as a final summary.

3.4. How QMOS works?

The current section provides an example to better convey how the QMOS functions. Given two sentences and a user's question, as shown in Fig. 3, the QMOS applies Pos-tagger to sentences and user's query. The result is shown in "*Part of speech tagging*" field. After this step, the QMOS stems all different words, as shown in "*Stemming*" field. Finally, the QMOS applies the stop word-removal procedure to sentences and question. The output is also given in "*Stop word-removal*" field. Furthermore, the method provides the significant information such as, the sentiment score of each sentence, the classification of positive, negative and neutral sentences, Total Sentence Score (TSS), identifying sentences with the same sentiment orientation of the opinion of the correlated user's question, identifying user's query relevant sentences and extracting the answers from the top ranked sentences using section "*3.3.4. summary generation*".

Table 6
Sample of the Questions.

Dataset	Questions
TAC 2008	Why did people enjoy "A Million Little Pieces"? What do people like about System of a Down? What reasons are given for positive opinions of David Irving's arrest and trial? What do people like about System of a Down? Why do people dislike George Clooney? Why don't people like Subway sandwiches?
DUC 2006	< /narr > Discuss conditions on American Indian reservations or among Native American communities. Include the benefits and drawbacks of the reservation system. Include legal privileges and problems. < /narr > < /narr > What are the advantages and disadvantages of home schooling? Is the trend growing or declining? < /narr > < /narr > Why are wetlands important? Where are they threatened? What steps are being taken to preserve them? What frustrations and setbacks have there been? < /narr >

4. Experimental evaluation

We conducted two experiments on datasets: *a)* Blog06 corpus provided by the Text Analysis Conference[4] (TAC) 2008; *b)* DUC 2006 datasets provided by Document Understanding Conference[5] (DUC). We describe the details of experiment 1 and experiment 2 in the followings sections.

4.1. Data set

This section describes the dataset used throughout the experiments. We used the data produced by the TAC 2008 and DUC 2006 to assess the performance of our proposed method, QMOS. The TAC 2008 includes three main tasks: *1) Question Answering; 2) Recognizing Textual Entailment; 3) Summarization (Update, Opinion Pilot)*. In this project, we focused on the opinion task of text summarization. Opinion summarization pilot (OSP) aims to summarize the opinions expressed in the blog documents. The OSP task is a complex task which contains *question answering, multi-document summarization* and *sentiment/opinion analysis*. Given a set of blog documents and a list of questions, a summary that answered to opinion-oriented questions is produced. The data for OSP task belongs to Blog06 corpus that has been used in several TREC tracks in 2006 and 2007. The data includes 609 blog documents relevant to 25 topics. Each topic contains 10–39 blog pages. It is worth noting, the summary text generated by QMOS is considered as a candidate summary and the summary text generated by the TAC 2008 is denoted the reference summary. The DUC 2006 data sets include 50 document clusters. Each cluster of DUC 2006 data sets includes 25 relative documents. To evaluate the QMOS, we need a gold standard data, which is a set of all correct results. For this purpose, in order to create a gold standard, the text summaries are produced using the opinionated sentences. Each text summary of multi-documents is created using the following steps:

1) *Assessment of opinionated questions* — the annotators determine the polarity of the user's question. In other words, they tag the opinion of the user's query with *'Polarity = Pos', 'Polarity = Neg' or 'Polarity = Neu'*. Table 6 shows an example of some questions of TAC 2008 and DUC 2006 datasets. In DUC 2006, we focused on opinion-asking topics.
2) *Sentence polarity annotation* — in the current step the annotators split the text contents into sentences, and then tag the opinion of each sentence with *'Polarity = Pos', 'Polarity = Neg', 'Polarity = Neu'* or *'Polarity = null'*.
3) *Sentence categorization* — the sentences are categorized into two distinct groups: objective sentences (*factual sentences, a sentence not expressed any opinion, 'Polarity = Neu' or 'Polarity = null'*) and subjective sentences (*'Polarity = Pos', 'Polarity = Neg'*).
4) *Production of summaries to answer each question* — at this step only the subjective sentences are selected to answer the user's question. They consider the following conditions to select appropriate sentences: *i)* the sentences should be close to the user's question; *ii)* the user's question and the sentences should have the same sentiment orientation (*positive or negative*). Finally, the produced summary for each question is considered as the gold standard dataset. The dataset includes approximately 5071 positive sentences, 3489 negative sentences, 239 but-clauses and 302 negations.

We conducted two experiments to assess the performance of the QMOS. We select a set of documents randomly. These documents are split into two distinct datasets (training dataset (70%) and testing dataset (30%)).

[4] https://tac.nist.gov/2008/index.html.
[5] http://duc.nist.gov.

4.2. Evaluation metrics

We used the standard ROUGE-N metric (Eq. (13)) (C.-Y. Lin, 2004) to evaluate the performance of the QMOS. ROUGE has been adopted by DUC[6] as the official evaluation metric for text summarization. ROUGE-N is calculated as follows:

$$\text{ROUGE} - \text{N} = \frac{\sum_{\text{S} \in \text{Reference summaries}} \sum_{\text{N-gram} \in \text{S}} \text{Count}_{\text{match}}(\text{N-gram})}{\sum_{\text{S} \in \text{Reference summaries}} \sum_{\text{N-gram} \in \text{S}} \text{Count}(\text{N-gram})} \tag{13}$$

Where, N indicates the length of the n-gram and Count $_{\text{match}}$ (N-gram) indicates the total NO. of the n-grams occurring in both a reference and a candidate summaries. Count (N-gram) indicates the No. of the n-grams in the reference summaries. In our experiments, we employed two metrics ROUGE-1 and ROUGE-2. We also measured the average ROUGE score using Eq. (14).

$$\textit{Average ROUGE Score}\ (ARS) = \frac{\text{ROUGE}-1 + \text{ROUGE} - 2}{2} \tag{14}$$

4.2.1. What is ROUGE and how it works for evaluation of summarization tasks?

ROUGE[7,8] stands for Recall-Oriented Understudy for Gisting Evaluation. It includes a set of metrics (e.g., ROUGE-N (*an n-gram of size 1, 2 and 3 is referred to as "unigram", "bigram" and "trigram", respectively*); ROUGE-L (*Longest Common Subsequence (LCS)*); ROUGE-W (*Weighted LCS-based statistics*); ROUGE-S (*Skip-bigram*) and ROUGE-SU (*Skip-bigram plus unigram*)) for evaluating automatic summarization of texts. It works by comparing an automatically produced summary (system-produced) against a reference (*gold*) summary (human-produced). Given a gold summary: "*diet must have apple and banana*" and a system summary: "*Apple and banana are for good diet*", we calculate the ROUGE-1, ROUGE-2 and ARS as follows.

1) We extract the individual words (*unigram*) and two adjacent words (*bigram*) from both gold and system summaries:

$$\textit{Gold summary}_{unigram} = \{diet,\ must,\ have,\ apple,\ and,\ banana\};$$
$$\textit{System summary}_{unigram} = \{Apple,\ and,\ banana,\ are,\ for,\ good,\ diet\};$$
$$\textit{Gold summary}_{\text{bigram}} = \{diet\ must,\ must\ have,\ have\ apple,\ apple\ and,\ and\ banana\};$$
$$\textit{System summary}_{\text{bigram}} = \{Apple\ and, and\ banana, banana\ are, are\ for, for\ good, good\ diet\}.$$

2) To calculate ROUGE-1 (*refers to the overlap of unigrams between the system summary and gold summary*) using Eq. (13), we first calculate the denominator of the ratio – which is the number of unigrams in the gold summary. So, it is 6. Then, for the numerator, we find how many unigrams co-occur in gold summary and system summary. So, it is 4. Finally, ROUGE-1 value would be:(*4/6 = 0.6666*).
3) To calculate ROUGE-2 (*refers to the overlap of bigrams between the system and gold summaries*) using Eq. (13), we first calculate the denominator of the ratio – which is the number of bigrams in the gold summary. So, it is 5. Then, for the numerator, we find how many bigrams co-occur in gold summary and system summary. So, it is 2. Finally, ROUGE-2 value would be:(*2/5 = 0.4*).
4) The ARS value is calculated using the Eq. (14): ((*0.4 + 0.6666*)*/2 = 0.5333*).

4.3. Experiment 1

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

4.3.1. Parameter setting

In the current section, we focused on optimizing QMOS parameters. To be more specific, we try to optimize the β (*refer to section e) Sentence similarity measure using SSM and WOSM (SSWOS)*) and the γ (*refer to Section 3.3.3. Compound Model (CM)*). In order to accomplish this, we ran the QMOS on the training dataset. We carry out experiments with different γ and β ranging from 0.1 to 0.9 with a step of 0.1. To estimate the values of γ and β, we used gradient search strategy (*a nested loop*) (Abdi et al., 2015a), Algorithm 4, where γ is outer loop and β is inner loop. In the first pass of the outer loop when the value of γ becomes 0.1 then control enters into the inner loop where β 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 7. Table 7 presents the experimental results achieved by using various β and γ values. We evaluated the results in terms of ROUGE-1, ROUGE-2 and ARS. On analyzing the results, we found that the best performance was achieved with $\beta = 0.8$ and $\gamma = 0.\ 5$. The β and γ produced values for the three metrics as follows: 0.4079 (ROUGE-1), 0.0824 (ROUGE-2), 0.2452 (ARS). The best values in Table 7 have been marked using boldface. As a result, we can recommend the setting $\beta = 0.8$, $\gamma = 0.\ 5$ to evaluate the QMOS on the test dataset.

[6] http://duc.nist.gov/.
[7] http://www.rxnlp.com/how-rouge-works-for-evaluation-of-summarization-tasks/.
[8] https://en.wikipedia.org/wiki/ROUGE_(metric).

Algorithm 4
optimizing.

```
Data set: Training data;
For γ = 0.1 to γ = 0.9
    For β = 0.1 to β = 0.9
        Begin
            Apply the QMOS to dataset;
        End
```

Table 7
Performance of the QMOS against different values of γ and β.

β	γ	ROUGE-1	ROUG-2	ARS
β = (0.1 .. 0.7)	*0.1*	.	.	.
	.	.	.	.
	0.9	.	.	.
β = 0.8	0.1	0.3872	0.0769	0.2321
	0.2	0.3914	0.0772	0.2343
	0.3	0.3953	0.0792	0.2373
	0.4	0.3980	0.0801	0.2391
	0.5	**0.4079**	**0.0824**	**0.2452**
	0.6	0.4071	0.0822	0.2447
	0.7	0.4061	0.0808	0.2435
	0.8	0.3950	0.0802	0.2376
	0.9	0.3910	0.0811	0.2361
β = (0.9)	*0.1*	.	.	.
	.	.	.	.
	0.9	.	.	.

Table 8
The Performance of the QMOS against other methods.

ROUGE values of the methods β = 0.8, γ = 0.5 System	ROUGE-1	ROUGE-2	ARS
QMOS	0.4123	0.0985	0.2554
IIITSum	0.1511	0.0323	0.0917
Summarizers	0.3279	0.0439	0.1859
CCNU	0.2011	0.0554	0.1283
PolyU	0.2932	0.0760	0.1846
NUS	0.3484	0.0546	0.2015
Italica	0.3796	0.0745	0.2271

4.3.2. Comparison with related methods

In this section, we will compare the QMOS performance with the other existing well-known methods. We compare QMOS with more other methods: *1)* NUS (Lin, Hoang, Qiu, Ye, & Kan, 2008); *2)* CCNU (He, Chen, Gui, & Li, 2008); *3)* Italica (Cruz, Troyano, Ortega, & Enríquez, 2008); *4)* Summarizers (Cruz et al., 2008); *5)* IIITSum (Varma et al., 2008); 6) PolyU (Li, You, Hu, & Wei, 2008). Since these systems obtained good results on the TAC 2008 and DUC 2006 datasets, we selected them for comparison. We ran our method on testing dataset only with $\gamma = 0.5$ and $\beta = 0.8$. The evaluation measures values are displayed in Table 8. Table 8 shows the QMOS obtained the best result in comparison with the Italica, which is the best existing approach and has an ARS of (22.71%). However, due to the result, the QMOS outperformed the other existing method.

Table 9
Detailed comparison between the QMOS and other approaches.

QMOS improvement (%) Metrics	Italica	NUS	PolyU	CCNU	Summarizers	IIITSum
ROUGE-1	+8.61	+18.34	+40.62	+105.02	+25.74	+172.87
ROUGE-2	+32.21	+80.40	+29.61	+77.80	+124.37	+204.95

Table 10
P-values produced by Wilcoxon's test by comparing QMOS with other methods.

Data set 5% significance level	IIITSum	Summarizers	CCNU	PolyU	NUS	Italica
Blog06	*ROUGE-1 metric*					
	0.011	0.022	0.012	0.004	0.002	0.030
	ROUGE-2 metric					
	0.010	0.023	0.011	0.031	0.030	0.022

4.3.3. Detailed comparison

We used the relative improvement Eq. (15) for comparison between the QMOS and other approaches.

$$\left(\frac{\text{Our method} - \text{Other method}}{\text{Other method}}\right) \times 100 \tag{15}$$

Table 9 displays the results. In Tables 9 "+" indicates that QMOS method improves the existing approaches. Table 9 shows that among the existing approaches the Italica obtained the best results. However, In comparison with the approach Italica, QMOS improved the performance of the Italica approach as follows: 8.61% (ROUGE-1), 32.21% (ROUGE-2).

4.3.4. Statistical significance test

We compared the QMOS with other existing method statistically using the Wilcoxon signed rank test ('*a non-parametric statistical hypothesis test*'). To do this, we create seven groups: *1) Italica, 2) IIITSum, 3) Summarizers, 4) CCNU, 5) PolyU 6) NUS and 7) QMOS*. Each two groups are compared at a time (e.g. QMOS, IIITSum). Each group includes the ROUGE-1 and ROUGE-2 scores. Table 10 presents the P-values produced by Wilcoxon's signed rank test for comparison of two groups at a time. We considered two hypotheses: H_0 (Null): there is not the difference between the ROUGE values of two groups. H_A (alternative): there exists a significant difference. All P-values are less than 0.05, As shown in Table 10. For example, the test between QMOS and the IIITSum produced a P-value of 0.011 (ROUGE-1). The same result is also achieved by all other methods. However, this is a strong evidence to accept the alternative hypothesis and refuse the null hypothesis.

4.4. Experiment 2

This section aims to examine the effectiveness of sentiment lexicon size, analyze the effect of *SSM, WOSM, CWE* and *SSA* approaches, and examine the influence of contextual polarity in sentiment analysis.

4.4.1. Comparison with $QMOS^{2}_{SSM+WOSM+CWE}$, $QMOS^{3}_{SSM+WOSM+SSA}$ and $QMOS^{4}_{SSM+WOSM}$

In this section, we compare $QMOS\ ^{1}_{SSM+WOSM+CWE+SSA}$, with $QMOS\ ^{2}_{SSM+WOSM+CWE}$, $QMOS\ ^{3}_{SSM+WOSM+SSA}$ and $QMOS\ ^{4}_{SSM+WOSM}$ where *SSM + WOSM + CWE* (semantic and syntactic similarity measures *combined with content word expansion*), *SSM + WOSM + SSA* (semantic and syntactic similarity measures *combined with Semantic Sentiment Approach*), *SSM + WOSM* (semantic similarity measure *combined with* syntactic similarity measure), and *SSM + WOSM + CWE + SSA* (*all four types of approaches*). In this experiment, our aim is to examine the efficiency of the combination *SSM + WOSM + CWE + SSA* on QMOS method.

The results of $QMOS\ ^{1}_{SSM+WOSM+CWE+SSA}$, $QMOS\ ^{2}_{SSM+WOSM+CWE}$, $QMOS\ ^{3}_{SSM+WOSM+SSA}$ and $QMOS\ ^{4}_{SSM+WOSM}$ are reported in Table 11. From the result, we can figure out that the performance of $QMOS\ ^{1}_{SSM+WOSM+CWE+SSA}$ is better than $QMOS\ ^{2}_{SSM+WOSM+CWE}$, $QMOS\ ^{3}_{SSM+WOSM+SSA}$ and $QMOS\ ^{4}_{SSM+WOSM}$ in terms of the ARS value. In other words, a QMOS method that combines all four approaches could achieve better ARS value that one that combines a subset of approaches. Due to the results, we used the combine *SSM + WOSM + CWE + SSA* for the proposed method.

Sentiment lexicon size — we also investigated whether more sentiment words (*bigger sentiment dictionary*) can lead to promising results. The relationship between the number of dictionary words and the performance (*in terms of the ARS value*) is shown in Fig. 4. The horizontal axis of Fig. 4 illustrates the number of words of each sentiment dictionaries. The vertical axis shows the *ARS* values. We see that the two smallest dictionaries, i.e., AFINN (*word size: 2477*) and WordNet-Affect (*word size: 4787*) perform poorly. There

Table 11
Performance of the QMOS against various tests (*SSM + WOSM + CWE + SSA*), *SSM + WOSM + CWE, SSM + WOSM + SSA* and *SSM + WOSM*.

ROUGE values of the methods $\beta = 0.8$, $\gamma = 0.5$ Methods	ROUGE-1	ROUGE-2	ARS
$QMOS\ ^{1}_{SSM+WOSM+CWE+SSA}$	0.4123	0.0985	0.2554
$QMOS\ ^{2}_{SSM+WOSM+CWE}$	0.3812	0.0745	0.2279
$QMOS\ ^{3}_{SSM+WOSM+SSA}$	0.3351	0.0546	0.1949
$QMOS\ ^{4}_{SSM+WOSM}$	0.2013	0.0513	0.1263

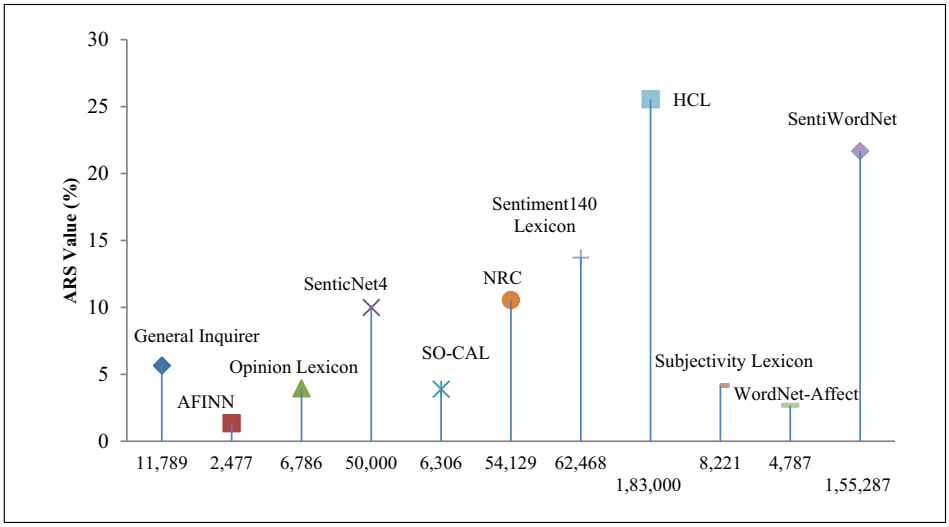


Fig. 4. Performance and the size of sentiment dictionary words.

Table 12
Performance of the QMOS against various tests.

ROUGE values of the methods $\beta = 0.8$, $\gamma = 0.5$ Methods	ROUGE-1	ROUGE-2	ARS
QMOS $_{(contextual\ polarity)}$	0.4123	0.0985	0.2554
QMOS $_{(without\ contextual\ polarity)}$	0.2802	0.0650	0.1726

seems to be a correlation that larger is better. It can be observed that the SentiWordNet (*word size: 155,287*) leads to the highest throughput. The reason is due to the wide coverage of sentiment words.

4.4.2. Influence of contextual polarity in sentiment analysis

In this section, we compare QMOS $_{(contextual\ polarity)}$ with QMOS $_{(without\ contextual\ polarity)}$. QMOS $_{(contextual\ polarity)}$ employs the subjective and objective sentence handling, question and conditional sentences handling, sentiment shifter (e.g. *negation handling, but-clause handling, sarcasm handling*) to analysis user's reviews, while QMOS $_{(without\ contextual\ polarity)}$ only uses the subjective and objective sentence handling to analysis user's reviews.

In this experiment, our aim is to examine the efficiency of the contextual polarity in sentiment analysis. The results of QMOS $_{(contextual\ polarity)}$ and QMOS $_{(without\ contextual\ polarity)}$ are presented in Table 12. It can be seen that the performance of QMOS $_{(contextual\ polarity)}$ is better than QMOS $_{(without\ contextual\ polarity)}$ in term of the results of ARS. Due to the results, we also considered the QMOS $_{(contextual\ polarity)}$ to analysis user's reviews.

4.5. Discussion

From Tables 7–9, we obtained the following observations. The QMOS 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 *CWE method,* while other methods (e.g., *Italica, PolyU and IIITSum*) do not use content word expansion approach.

2*)* Unlike other method (e.g, *NUS, CCNU, Summarizers*), the QMOS 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.

3) Since a dictionary-based method has limited word coverage, we combined several sentiment lexicons in order to tackle the aforementioned challenge, while other methods (*e.g, IIITSum, Summarizers, CCNU, NUS and Italica*) do not use the combination of several sentiment lexicons. Furthermore, in the case where the word does not appear in the lexicon (HCL), QMOS uses SSA method to compute the sentiment score of that word. We also examined whether more sentiment words can lead to better performance of the QMOS. Fig. 4 shows a bigger sentiment dictionary leads to the promising result.

4) The QMOS also considers contextual polarity and the types of sentences in sentiment analysis, while other methods (e.g, IIITSum, Summarizers (*excluding types of sentences*), CCNU (*excluding interrogative sentences handling*), PolyU, NUS (*excluding negation handling*) and Italica (*excluding types of sentences)*. Regarding to Table 12, it can be observed that contextual polarity handling improves the performance of QMOS method.

5. Conclusion and future work

Nowadays, due to the huge amount of information in the form of reviews and blogs, sentiment analysis, which extracts automatically user's opinion from large review text, has attracted much attention of the researchers. In this paper, we proposed a query based multi-documents opinion-oriented summarization (QMOS) method. QMOS includes two main stages: *1)* sentiment analysis, to identify sentiment orientation or subjective information; *2)* summarizer, to identify and extract user's query relevant sentences which contain an expression of opinion. QMOS combines multiple sentiment lexicons to expand the sentiment dictionary coverage. It also employed the SSA to improve word coverage limit and to determine sentiment score of a word if it not included in a sentiment lexicon.

On the other hand, QMOS integrated multiple strategies to tackle the problem, sentiment shifter: the sentiment orientation of a word defined by a sentiment dictionary can be reversed since the sentiment orientation of a word depends on the context in which it appears (e.g. *negation handling, but-clause handling, sarcasm handling*). Furthermore, QMOS also considers the type of sentence (e.g., *subjective and objective sentence, question and conditional sentence*), since they can affect the performance of the method. The summarizer stage integrated semantic and syntactic information to distinguish the meaning when comparing two sentences (*user's query is considered as a sentence*). It also employs query expansion approach to solve the fundamental problem of word mismatch in the comparison between user's question and sentences.

We conducted two experiments. In the first experiment, we evaluated our method over TAC 2008 and DUC 2006 datasets. Initially, we optimized the parameters of QMOS. Later, we evaluated the performance of the QMOS over testing dataset using the ROUGE metrics. We compared the QMOS and existing methods. The results display that the QMOS outperforms the existing methods. In the second evaluation, we obtained the following observations: *1) we* analyze the effect of *SSM, WOSM, CWE* and *SSA* approaches on QMOS. The result shows a QMOS method that combines all four approaches could achieve better ARS value that one that combines a subset of approaches; *2)* Regarding the '*influence of contextual polarity in sentiment analysis,* the QMOS produces good results when it considers the contextual polarity, e.g., negation handling, but-clause handling, sarcasm handling.

In future work, we plan to study in more depth the problem of comparative sentences and sarcastic sentence handling. We also aim to consider the passive and active voice in the comparison between two sentences, since the current method is not able to distinguish between an active sentence and a passive sentence. Given three sentences (A: 'Teacher likes his student.'; 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, 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.

Acknowledgement

This work is supported by The Ministry of Higher Education (MOHE) under Q.J130000.21A2.03E53 - Statistical Machine Learning Methods to Text Summarizations. The authors would like to thank Research Management Centre (RMC), Universiti Teknologi Malaysia (UTM) for the support in R & D, UTM Big Data Centre (BDC) for the inspiration in making this study a success. The authors would also like to thank the anonymous reviewers who have contributed enormously to this work.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ipm.2017.12.002.

References

Abdi, A., & Idris, N. (2014a). Automated summarization assessment system: Quality assessment without a reference summary. *The International Conference On Advances In Applied Science And Environmental Engineering - ASEE 2014*. IRED Press.

Abdi, A., Idris, N., Alguliev, R. M., & Aliguliyev, R. M. (2015a). Automatic summarization assessment through a combination of semantic and syntactic information for intelligent educational systems. *Information Processing & Management, 51*, 340–358.

Abdi, A., Idris, N., Alguliyev, R. M., & Aliguliyev, R. M. (2015b). Query-based multi-documents summarization using linguistic knowledge and content word expansion. *Soft Computing,* 1–17.

Abdi, S. A., & Idris, N. (2014b). An analysis on student-written summaries: Automatic assessment of summary writing. *International Journal of Enhanced Research in Science Technology & Engineering, 3*, 466–472.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *LREC. Vol. 10. LREC* (pp. 2200–2204).

Badrinath, R., Venkatasubramaniyan, S., & Madhavan, C. V. (2011). *Improving query focused summarization using look-ahead strategy. Advances in information retrieval*. Springer641–652.

Balahur, A., Kabadjov, M., Steinberger, J., Steinberger, R., & Montoyo, A. (2012). Challenges and solutions in the opinion summarization of user-generated content. *Journal of Intelligent Information Systems, 39*, 375–398.

Bharti, S. K., Babu, K. S., & Jena, S. K. (2015). Parsing-based sarcasm sentiment recognition in Twitter data. *Advances in social networks analysis and mining (ASONAM), 2015 IEEE/ACM international conference on* (pp. 1373–1380). IEEE.

Cambria, E., Poria, S., Bajpai, R., & Schuller, B. W. (2016). SenticNet 4: A semantic resource for sentiment analysis based on conceptual primitives. *COLING* (pp. 2666–2677). .

Cerini, S., Compagnoni, V., Demontis, A., Formentelli, M., & Gandini, G. (2007). Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. *Language resources and linguistic theory: Typology, second language acquisition, English linguistics,* 200–210.

Chali, Y., Hasan, S. A., & Joty, S. R. (2011). Improving graph-based random walks for complex question answering using syntactic, shallow semantic and extended string subsequence kernels. *Information Processing & Management, 47*, 843–855.

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications,*

72, 221–230.
Cruz, F. L., Troyano, J. A., Ortega, F. J., & Enríquez, F. (2008). The italica system at TAC 2008 opinion summarization task. *TAC*.
Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research,* 457–479.
Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review, 47*, 1–66.
He, T., Chen, J., Gui, Z., & Li, F. (2008). CCNU at TAC 2008: Proceeding on using semantic method for automated summarization yield. *TAC*.
Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 168–177). ACM.
Hu, Y.-H., Chen, Y.-L., & Chou, H.-L. (2017). Opinion mining from online hotel reviews–A text summarization approach. *Information Processing & Management, 53*, 436–449.
Hung, C., & Chen, S.-J. (2016). Word sense disambiguation based sentiment lexicons for sentiment classification. *Knowledge-Based Systems, 110*, 224–232.
Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New phytologist, 11*, 37–50.
Jindal, N., & Liu, B. (2006). Identifying comparative sentences in text documents. *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 244–251). ACM.
Khan, F. H., Qamar, U., & Bashir, S. (2016a). A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet. *Knowledge and information Systems,* 1–22.
Khan, F. H., Qamar, U., & Bashir, S. (2016b). SWIMS: Semi-supervised subjective feature weighting and intelligent model selection for sentiment analysis. *Knowledge-Based Systems, 100*, 97–111.
Kolchyna, O., Souza, T. T., Treleaven, P., & Aste, T. (2015). Twitter sentiment analysis: Lexicon method, machine learning method and their combination. *arXiv preprint arXiv:1507.00955*.
Lee, S., & Choeh, J. Y. (2014). Predicting the helpfulness of online reviews using multilayer perceptron neural networks. *Expert Systems with Applications, 41*, 3041–3046.
Li, S., Lee, S. Y. M., Chen, Y., Huang, C.-R., & Zhou, G. (2010). Sentiment classification and polarity shifting. *Proceedings of the 23rd international conference on computational linguistics* (pp. 635–643). Association for Computational Linguistics.
Li, W., You, O., Hu, Y., & Wei, F. (2008). PolyU at TAC 2008. *TAC*.
Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006). Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on, 18*, 1138–1150.
Lin, C.-Y. (2004). Rouge: A package for automatic evaluation of summaries. *Text summarization branches out: Proceedings of the ACL-04 workshop* (pp. 74–81). .
Lin, Z., Hoang, H. H., Qiu, L., Ye, S., & Kan, M.-Y. (2008). NUS at TAC 2008: Augumenting timestamped graphs with event information and selectively expanding opinion contexts. *TAC*.
Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies, 5*, 1–167.
Lloret, E., Boldrini, E., Vodolazova, T., Martínez-Barco, P., Muñoz, R., & Palomar, M. (2015). A novel concept-level approach for ultra-concise opinion summarization. *Expert Systems with Applications, 42*, 7148–7156.
Lu, Y., Duan, H., Wang, H., & Zhai, C. (2010). Exploiting structured ontology to organize scattered online opinions. *Proceedings of the 23rd international conference on computational linguistics* (pp. 734–742). Association for Computational Linguistics.
Lunando, E., & Purwarianti, A. (2013). Indonesian social media sentiment analysis with sarcasm detection. *Advanced computer science and information systems (ICACSIS), 2013 international conference on* (pp. 195–198). IEEE.
Mendoza, M., Bonilla, S., Noguera, C., Cobos, C., & León, E. (2014). Extractive single-document summarization based on genetic operators and guided local search. *Expert Systems with Applications, 41*, 4158–4169.
Miller, G. A., & Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes, 6*, 1–28.
Mladenović, M., Mitrović, J., Krstev, C., & Vitas, D. (2016). Hybrid sentiment analysis framework for a morphologically rich language. *Journal of Intelligent Information Systems, 46*, 599–620.
Mohammad, S. M., Kiritchenko, S., & Zhu, X. (2013). NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. *arXiv preprint arXiv:1308.6242*.
Narayanan, R., Liu, B., & Choudhary, A. (2009). Sentiment analysis of conditional sentences. *Proceedings of the 2009 conference on empirical methods in natural language processing: Volume 1-volume 1* (pp. 180–189). Association for Computational Linguistics.
Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. *arXiv preprint arXiv:1103.2903*.
Nishikawa, H., Hasegawa, T., Matsuo, Y., & Kikui, G. (2010). Opinion summarization with integer linear programming formulation for sentence extraction and ordering. *Proceedings of the 23rd international conference on computational linguistics: Posters* (pp. 910–918). Association for Computational Linguistics.
Otterbacher, J., Erkan, G., & Radev, D. R. (2005). Using random walks for question-focused sentence retrieval. *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 915–922). Association for computational linguistics.
Paul, M. J., Zhai, C., & Girju, R. (2010). Summarizing contrastive viewpoints in opinionated text. *Proceedings of the 2010 conference on empirical methods in natural language processing* (pp. 66–76). Association for Computational Linguistics.
Pérez, Gliozzo, A. M., 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. *FLAIRS Conference* (pp. 358–363). .
Povoda, L., Burget, R., & Dutta, M. K. (2016). Sentiment analysis based on support vector machine and big data. *Telecommunications and signal processing (TSP), 2016 39th international conference on* (pp. 543–545). IEEE.
Rana, T. A., & Cheah, Y.-N. (2016). Aspect extraction in sentiment analysis: Comparative analysis and survey. *Artificial Intelligence Review, 46*, 459–483.
Riloff, E., Patwardhan, S., & Wiebe, J. (2006). Feature subsumption for opinion analysis. *Proceedings of the 2006 conference on empirical methods in natural language processing* (pp. 440–448). Association for Computational Linguistics.
Riloff, E., & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. *Proceedings of the 2003 conference on Empirical methods in natural language processing* (pp. 105–112). Association for Computational Linguistics.
Sadh, A., Sahu, A., Srivastava, D., Sanyal, R., & Sanyal, S. (2012). Extraction of relevant figures and tables for multi-document summarization. *Computational Linguistics and Intelligent Text Processing,* 402–413.
Saggionα, H., & Funk, A. (2010). Interpreting SentiWordNet for opinion classification. *Proceedings of the seventh conference on international language resources and evaluation LREC10*.
Shahana, P., & Omman, B. (2015). Evaluation of features on sentimental analysis. *Procedia Computer Science, 46*, 1585–1592.
Stone, P. J., & Hunt, E. B. (1963). A computer approach to content analysis: Studies using the general inquirer system. *Proceedings of the May 21-23, 1963, spring joint computer conference* (pp. 241–256). ACM.
Strapparava, C., & Valitutti, A. (2004). WordNet affect: An affective extension of WordNet. *LREC. Vol. 4. LREC* (pp. 1083–1086). Citeseer.
Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational Linguistics, 37*, 267–307.
Tayal, M. A., Raghuwanshi, M. M., & Malik, L. G. (2017). ATSSC: Development of an approach based on soft computing for text summarization. *Computer Speech & Language, 41*, 214–235.
Tsuruoka, Y., & Tsujii, J.i. (2005). Bidirectional inference with the easiest-first strategy for tagging sequence data. *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 467–474). Association for Computational Linguistics.
Vani, K., & Gupta, D. (2014). Using K-means cluster based techniques in external plagiarism detection. *Contemporary Computing and Informatics (IC3I), 2014 International Conference on* (pp. 1268–1273). IEEE.

Varma, V., Pingali, P., Katragadda, R., Krishna, S., Ganesh, S., Sarvabhotla, K., et al. (2008). IIIT Hyderabad at TAC 2008. *TAC*.
Wan, X., Yang, J., & Xiao, J. (2007). Manifold-ranking based topic-focused multi-document summarization. *IJCAI. Vol. 7. IJCAI* (pp. 2903–2908).
Wang, D., Zhu, S., & Li, T. (2013). SumView: A Web-based engine for summarizing product reviews and customer opinions. *Expert Systems with Applications, 40*, 27–33.
Wang, G., Zhang, Z., Sun, J., Yang, S., & Larson, C. A. (2015). POS-RS: A random subspace method for sentiment classification based on part-of-speech analysis. *Information Processing & Management, 51*, 458–479.
Xia, R., Xu, F., Yu, J., Qi, Y., & Cambria, E. (2016). Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis. *Information Processing & Management, 52*, 36–45.
Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., et al. (2005). Improving web search results using affinity graph. *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 504–511). ACM.