Searching web documents using a summarization approach

Rani Qumsiyeh and Yiu-Kai Ng
Department of Computer Science, Brigham Young University, Provo, Utah, USA

Received 18 November 2015
Revised 18 November 2015
Accepted 27 November 2015

Abstract

Purpose – The purpose of this paper is to introduce a summarization method to enhance the current web-search approaches by offering a summary of each clustered set of web-search results with contents addressing the same topic, which should allow the user to quickly identify the information covered in the clustered search results. Web search engines, such as Google, Bing and Yahoo!, rank the set of documents *S* retrieved in response to a user query and represent each document *D* in *S* using a title and a snippet, which serves as an abstract of *D*. Snippets, however, are not as useful as they are designed for, i.e. assisting its users to quickly identify results of interest. These snippets are inadequate in providing distinct information and capture the main contents of the corresponding documents. Moreover, when the intended information need specified in a search query is ambiguous, it is very difficult, if not impossible, for a search engine to identify precisely the set of documents that satisfy the user's intended request without requiring additional information. Furthermore, a document title is not always a good indicator of the content of the corresponding document either.

Design/methodology/approach – The authors propose to develop a query-based summarizer, called Q_{Sum}, in solving the existing problems of Web search engines which use titles and abstracts in capturing the contents of retrieved documents. Q_{Sum} generates a concise/comprehensive summary for each cluster of documents retrieved in response to a user query, which saves the user's time and effort in searching for specific information of interest by skipping the step to browse through the retrieved documents one by one.

Findings – Experimental results show that Q_{Sum} is effective and efficient in creating a high-quality summary for each cluster to enhance Web search.

Originality/value – The proposed query-based summarizer, Q_{Sum}, is unique based on its searching approach. Q_{Sum} is also a significant contribution to the Web search community, as it handles the ambiguous problem of a search query by creating summaries in response to different interpretations of the search which offer a "road map" to assist users to quickly identify information of interest.

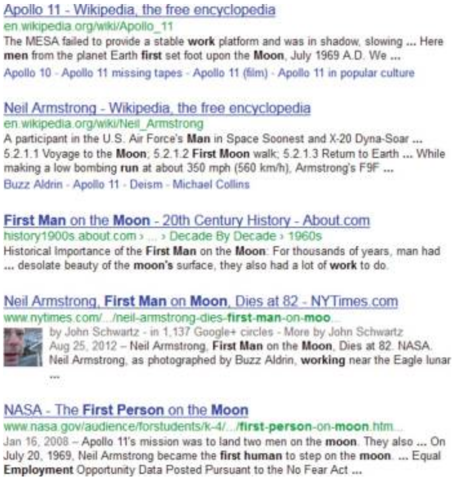
Keywords Web search, Query processing, Summarization

Paper type Research paper

1. Introduction

Current Web search engines rank retrieved documents based on their likelihood of relevance to a user query *Q* and represent each document using a title and a snippet[1]. The snippet, however, is often very similar to others created for documents retrieved in response to *Q* and generated using sentences/phrases in the corresponding document *D* in where the keywords in *Q* appear, which may not capture the main content of *D*. Consider the top-5 results retrieved by Google (on February 16, 2015) for the query "First man to walk on the moon" as shown in Figure 1. The titles and snippets of the results show the same information, i.e. Neil Armstrong was the first man to walk on the moon. If the user who submitted the query was interested in specific information, such as the shuttle used during the mission, astronauts that accompanied Neil Armstrong, length of the journey, etc., the user must scan through the

International Journal of Web Information Systems
Vol. 12 No. 1, 2016
pp. 83-101
© Emerald Group Publishing Limited
1744-0084
DOI 10.1108/IJWIS-11-2015-0039



Apollo 11 - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Apollo_11
The MESA failed to provide a stable **work** platform and was in shadow, slowing ... Here **men** from the planet Earth **first** set foot upon the **Moon**, July 1969 A.D. We ...
Apollo 10 - Apollo 11 missing tapes - Apollo 11 (film) - Apollo 11 in popular culture

Neil Armstrong - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Neil_Armstrong
A participant in the U.S. Air Force's **Man** in Space Soonest and X-20 Dyna-Soar ... 5.2.1.1 Voyage to the **Moon**; 5.2.1.2 **First Moon** walk; 5.2.1.3 Return to Earth ... While making a low bombing **run** at about 350 mph (560 km/h), Armstrong's F9F ...
Buzz Aldrin - Apollo 11 - Deism - Michael Collins

First Man on the **Moon** - 20th Century History - About.com
history1900s.about.com › ... › Decade By Decade › 1960s
Historical Importance of the **First Man** on the **Moon**: For thousands of years, man had ... desolate beauty of the **moon's** surface, they also had a lot of **work** to do.

Neil Armstrong, **First Man** on **Moon**, Dies at 82 - NYTimes.com
www.nytimes.com/.../neil-armstrong-dies-**first-man**-on-**moo**...
by John Schwartz - in 1,137 Google+ circles - More by John Schwartz
Aug 25, 2012 – Neil Armstrong, **First Man** on the **Moon**, Dies at 82. NASA. Neil Armstrong, as photographed by Buzz Aldrin, **working** near the Eagle lunar ...

NASA - The **First Person** on the **Moon**
www.nasa.gov/audience/forstudents/k-4/.../**first-person**-on-**moon**.htm...
Jan 16, 2008 – Apollo 11's mission was to land two men on the **moon**. They also ... On July 20, 1969, Neil Armstrong became the **first human** to step on the **moon**. ... Equal **Employment** Opportunity Data Posted Pursuant to the No Fear Act ...

Figure 1.
The top-5 results retrieved by Google for the query "first man to walk on the moon"

retrieved documents one by one, as there is no indication in which retrieved documents additional information might be included, which is a time-consuming and tedious process. A solution to this problem is to create a *summary* of documents belonged to a subject area, i.e. topic, relevant to the user query that captures the main content of the documents, which allow the users to quickly draw a conclusion on a topic or its summary that includes materials satisfying their information needs.

Document summarization systems have emerged which automatically create a summary of a document or set of documents based on a search query. In these query-based summarization systems, a summary is generated on (each of) the top-N ($\geq$1) documents retrieved by a search engine in response to a user query, which allows ordinary Web users, as well as professional information consumers and researchers, to quickly familiarize themselves with a large volume of retrieved information. If such a system generates a single summary on multiple documents, it is a *query-based multi-document summarization system*.

A multi-document summary offers a brief review of the subject area covered in a set of documents *SD* by:

- extracting mutual content across the documents while *avoiding repetition*;
- capturing *unique* (related, respectively) information in *SD*;
- providing an overview of various subtopics, if they exist, of the subject area; and
- identifying the events that evolve over time.

However, developing a fully automated multi-document summarization system is a challenging task, as the system must:

- eliminate *redundancy*, i.e. same or similar information presented in different documents should be filtered;
- account for the *temporal dimension*, i.e. a new piece of information should override out-dated information;

- choose an ideal *compression ratio* to ensure that a summary includes sufficient contents of the corresponding documents in a reasonable length;
- achieve a (near-) complete *coverage* to capture the essential contents of the documents; and
- resolve the *co-reference* issue of documents by detecting various references on the same item.

In this paper, we introduce a *query-based multi-document summarizer*, called Q_{Sum}, which enhances Web search. Q_{Sum} allows novice, as well as expert, users to post a query Q and quickly locate the desired information captured in the summary of a clustered set of topically related documents. Q_{Sum} queries three major Web search engines, Google, Bing and Yahoo!, using Q, assigns retrieved documents (based on their topics) to labeled clusters and creates a single summary of each cluster of documents.

A summary of clustered documents is useful, as typical Web search queries are *short* and often *ambiguous* in meaning (Shen and Pan, 2006). For this reason, existing Web search engines consider various interpretations of the intended information needs of a user query Q and retrieve documents that cover related topics of Q. During the process of answering Q, Q_{Sum} creates a cluster label and a summary on the corresponding set of clustered documents in capturing the main contents of the documents. For example, if the search query is "tiger", the retrieved documents can be various in terms of their contents, which might discuss the Mac OS, a fish, the golf player Tiger Woods, etc. A cluster summary distinguishes the content of the clustered documents from other cluster summaries on different subject areas, and a summary can serve as a cluster label surrogate when a user's confidence on the cluster label is *low*.

We have evaluated the quality of Q_{Sum}-generated summaries using the document understanding conference (DUC) data set and compared the summaries against:

- those created by existing state-of-the-art query-based multi-document summarization tools; and
- snippets generated by Google in terms of the time required to locate desired information.

Furthermore, we have conducted several controlled experiments to analyze the quality of a Q_{Sum}-generated summary in terms of grammar, anti-redundancy, referential clarity, coverage and structure and coherence. Experimental results show that Q_{Sum} is *highly effective* and *efficient* in generating a concise and comprehensive summary for a cluster of documents retrieved for a Web query.

Q_{Sum} is a contribution to the Web and information retrieval community, as it:

- creates summaries, one for each relevant topic derived from a user query, which is missing in existing popular Web search engines;
- provides the user with an unbiased information source on a particular topic, as the creation of each summary is fully automated, without any subjective human intervention;
- enhances Web search by eliminating redundant retrieved information while achieving high coverage and helping the user quickly locate desired information; and
- establishes, as a by-product, a new source of information for answering users' questions, as a summary which contains significant information from various documents likely contains the answers to the related questions.

Q_{Sum} is unique, as unlike snippets generated by current Web search engines which may not reflect the main contents of their respective retrieved documents, Q_{Sum} creates a summary for a collection of retrieved documents *C* that captures related information of the subject area indicated by the cluster label of *C*. Moreover, Q_{Sum} does not require training/learning in creating summaries, a merit of Q_{Sum}.

We present our work as follows. In Section 2, we discuss existing multi-document summarization methods. In Section 3, we detail the design of Q_{Sum}. In Section 4, we present the performance evaluation of Q_{Sum}. In Section 5, we give a concluding remark.

2. Related work

Q_{Sum} extracts sentences from documents to create a summary. MEAD (summarization.com/mead), an extractive summarization method, scores sentences using sentence-level and inter-sentence features. NeATS Lin and Hovy (2002) is a multi-document summarizer based on SUMMARIST, a single-document summarizer. MEAD and NeATS consider the sentence space but ignore topics covered in documents. Sentence position, term frequency and topic signature have been considered for selecting important content from documents for summarization, which are analyzed by Q_{Sum} for creating a summary of documents in a cluster.

The authors of Bhandari *et al.* (2008) score sentences based on the representation of each sentence in the latent topic space provided by a trained Probabilistic Latent Semantic Analysis model. Arora and Ravindran (2008) use Latent Dirichlet Allocation to create multi-document summaries by selecting sentences from the topic with the largest likelihood. Compared with the summarization approach of Q_{Sum}, these systems neither perform any *redundancy checking* nor achieve *high coverage*, as they focus on sentences addressing the same topic.

The graph-based PageRank algorithm Altman and Tennenholtz (2005) determines the sentences that are the most salient in a collection of documents and closest to a given topic. Graph-based methods, however, do not account for multiple topics within a document. Leskovec *et al.* (2004) construct a document graph using subject-verb- object triples, semantic normalization and co-reference resolution and consider node degree, PageRank and Hubs to generate statistics for the nodes, which represent sentences, to rank the sentences. Amini and Usunier (2009) present a transductive approach that learns the ranking function over sentences in retrieved documents using labeled instances. Q_{Sum} does not require labeled instances, as no training is involved in its summarization and thus minimizes the overhead and at the same time avoids the system scalability problem.

3. The summarization approach

As stated in Section 1, titles and snippets created by existing Web search engines may not capture the contents of their corresponding documents. A summary of a cluster *C*, which consists of search results retrieved in response to a query submitted by user *U*, addresses the problem of titles and snippets. (Detailed design and performance evaluation of Q_{Sum}-created *labels* and their *clusters* of retrieved documents generated in response to a user query can be found in Qumsiyeh and Ng (2013). (See Figure 2 for a sample of cluster labels and cluster of documents).

Summarization is a promising approach in dealing with the problem of ineffective snippets and information overload, as it provides a summary (abstract) that includes the key concepts covered in a (subset of clustered) document(s). An ideal text summary of a (given set of) document(s) *S*:

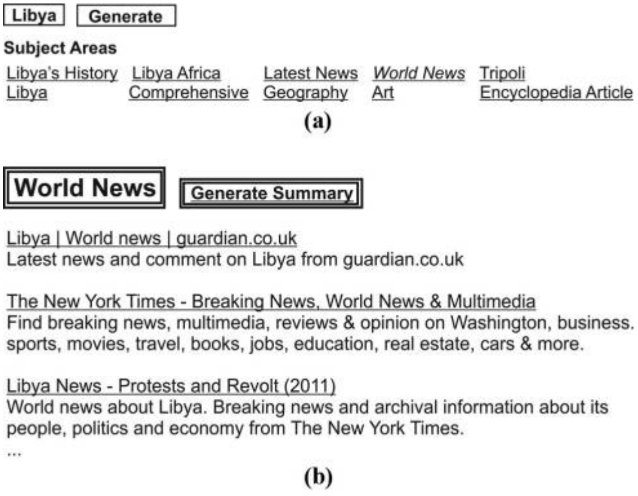
- includes *unique*, but excludes extraneous and redundant, information presented in (various documents in) *S* (as discussed in Section 3.2.4);
- must be coherent and comprehensible, which can be achieved using natural language processing to handle *co-reference* and the *temporal dimension* of information (to be introduced in Sections 3.2.2 and 3.2.5, respectively); and
- is appropriate in length, as a *very brief* summary is likely to exclude some important information in *S*, whereas a *very detailed* one is likely to repeat the same or include non-essential information in *S* (addressed in Section 3.2).

3.1 Multi- versus single-document summaries

Multi-document summarization of a set of documents *S* can be created by concatenating the summary of each document in *S*. This approach, however, can yield a summary with poor quality. For example, the same referencing expression "president" in two different documents may not necessarily refer to the same person. Moreover, useful pieces of information could be ignored due to the temporal ordering of the documents when newer information override older ones in the summary. Six issues have been addressed and emphasized in the design of a (query-based) multi-document summarizer Ou *et al.* (2006) as compared with the design of a single-document summarization method:

(1) *Redundant information*: A multi-document summary is expected to eliminate sentences in a set of topically related articles that convey the same piece of information, which is much higher than its counterpart in a single article.

(2) *Temporal dimension*: A multi-document summarization approach orders sentences in a given set of documents partially based on their publication dates.



Subject Areas

Libya's History | Libya Africa | Latest News | World News | Tripoli
Libya | Comprehensive | Geography | Art | Encyclopedia Article

Libya | World news | guardian.co.uk
Latest news and comment on Libya from guardian.co.uk

The New York Times - Breaking News, World News & Multimedia
Find breaking news, multimedia, reviews & opinion on Washington, business, sports, movies, travel, books, jobs, education, real estate, cars & more.

Libya News - Protests and Revolt (2011)
World news about Libya. Breaking news and archival information about its people, politics and economy from The New York Times.
...

Notes: (a) The top-10 ranked cluster labels; (b) top-3 documents in the "World News" cluster

Figure 2. Cluster labels and documents in the cluster labeled "world news" created and retrieved by Q_{Sum}, respectively, in response to the query "Libya"

(3) The *length* of a summary is smaller for a collection of dozens/hundreds of topically related documents than for concatenated single-document summaries.

(4) *The co-reference problem*: A summarization approach must identify whether two references in two different sentences address the same object. A multi-document summary may contain sentences extracted from several documents, which may include a pronoun without its preceding referent.

(5) Achieving good *coverage* in multi-document summaries is difficult, as there are a number of informative sentences in topically related articles that can be selected for creating a summary due to the variety of "subtopics", whereas a single document tends to focus on a few subtopics.

(6) *User interface* must be simple, easy to use and allow the user to view the context of the original document by clicking the corresponding sentence in the summary.

A multi-document summary has several advantages over single-document summaries, as the former:

- provides an overview of various subtopics, if they exist, of a particular subject;
- gives the user more information about the subject while eliminating common information across many documents; and
- identifies a subject or research topic that evolves over time.

We have chosen the multi-document summarization over the single-document summarization approach for Q_{Sum}, as its advantages outweigh its complexity.

Two of the commonly used multi-document summarization methods are extractive and abstractive summarization. *Extractive summarization* assigns saliency scores to units, such as sentences or paragraphs in a document, such that each assigned score reflects the *significance* of the corresponding unit in capturing key concepts presented in the set of documents *SD* to be summarized and units with the highest scores are extracted, whereas *abstractive summarization*, which requires information fusion and sentence reformulation, rewrites sentences in *SD* to be included in the summary so that they are readable and grammatically correct. Q_{Sum} adopts the extractive summarization strategy at the sentence level.

3.2 $\mathrm{Q}_{\mathrm{Sum}}$*-generated summaries*

Given a user query Q, Q_{Sum} creates a summary for each cluster C of documents by:

- downloading and preprocessing the top-33 documents retrieved by each of the three Web search engines, Google, Bing and Yahoo!, for Q (discussed in Section 3.2.1), as a collection of 100 documents is an *ideal* set for generating clusters and summaries (Dunlavy *et al.*, 2007);
- identifying and associating all (pro)nouns in the retrieved documents with their referents (detailed in Section 3.2.2);
- assigning each sentence S in documents in C a score, denoted RS, which reflects the *relative significance* of S in capturing the key concepts covered in documents in C according to a set of features (defined in Section 3.2.3);
- choosing the top-M (≥1) sentences (based on their RS scores) from the documents in C, such that $(\Sigma_{i=1}^{M-1} L_i) < 9 \times \mathit{Size}$ and $(\Sigma_{i=1}^{M} L_i) \geq 9 \times \mathit{Size}$, where L_i is the

number of words in a sentence *i* in *C* and *Size* is approximately 10 per cent of the total number of words[2] in *C*;

- clustering the *M* sentences to yield *sentence clusters* using the hierarchical agglomerative clustering (HAC) algorithm based on word-correlation factors[3] (Qumsiyeh and Ng, 2013) (as presented in Section 3.2.4);
- selecting the top-*N* sentences (based on their *RS* scores) from each sentence cluster created in Step (v) such that $(\Sigma_{i=1}^{N-1} L_i) < Size$ and $(\Sigma_{i=1}^{N} L_i) \geq Size$ and, if desired; and
- re-weighting the selected sentences based on their temporal dimensions to capture the flow of events (as explained in Section 3.2.5).

If the number of sentences *N* to be selected for a summary is *less* than the number of created *sentence clusters* of *C*, the *N* sentences (one from each top-*N* ranked sentence cluster) with the highest *RS* score are chosen.

Q_{Sum} starts with 9 × size words in creating a cluster summary, as Schlesinger *et al.* (2008) claim that 9 × size words are required to generate a sufficient, distinct-content summary. Each Q_{Sum}-generated multi-document summary:

- extracts mutual content across the documents while avoiding *repetition*;
- captures *unique* (*related*, respectively) information in the documents; and
- allows a *click* on a sentence in the summary to view the corresponding document.

3.2.1 Document preprocessing. The set of 99 documents retrieved from Google, Bing and Yahoo! are first preprocessed, where each retrieved document is in HTML format. We consider HTML pages for creating multi-document summaries, as:

- other formats are complex to process and require additional overhead time; and
- over 99 per cent of the documents retrieved by Google, Bing and Yahoo! are in HTML format.

Each one of the 99 retrieved documents *D* is parsed to remove surplus data, which include links to other documents, advertisements and non-textual data, such as images and videos, and retain only textual information, i.e. title, text, date and the URL of *D*, which are converted into uniform XML format for easy data lookup. Text in each document is segmented into sentences using a short list of end-of-sentence punctuation marks, along with regular expressions for detecting decimals, email addresses and ellipse, to ensure reliable identification of sentence boundaries[4]. Hereafter, each sentence is parsed into a sequence of word tokens using the Connexor Parser (www.connexor.com/nlplib/?q=demo/syntax). For each word token, its *Doc(ument) ID*, *Sent(ence) ID*, *word form* (in the text), *stem* (generated using the Porter stemming algorithm) and *creation date* of the corresponding document are stored. The *Doc ID* and *Sent ID* identify the document from where sentences are extracted and the relative positions of sentences in the corresponding document, respectively, the *stem* of a word is used in different *sentence/document similarity* formulas, and the *date* is for re-weighting the sentences in a summary based on their *temporal dimension*.

3.2.2 Solving the co-reference resolution problem. Co-reference resolution refers to the problem of determining which (common) (pro)noun phrases refer to which real-world entity as given in a document. Consider the sentence, "I study computer science. It is a very demanding major". In solving the co-reference problem, the pronoun "It" is replaced by "Computer science". In summarization, it is required to replace a (pro)noun in a sentence with its referencing entity, as sentences in the summary can lose their original orders and yield a false indication of what the (pro)noun refers to. Q_{Sum} uses an open source package (markwatson.com/ opensource/) for performing *co-reference resolution* in solving the co-reference problem to begin with.

3.2.3 Ranking sentences in clusters. Each sentence *S* in a document cluster *C* is assigned a *weight*, denoted *RS*, which indicates its relative significance in capturing the contents of the documents in *C*. To compute the *weight* (i.e. *RS*) of *S*, Q_{Sum} uses the following features:

- *Title Frequency* (*TiF*) is the number of words in *S* that appear in the *cluster label* of *C*.
- As a summary of the documents in *C* reflects the content of *C*, it should contain sentences that include frequently occurred, significant words in *C*. We define the *significance factor*, denoted *SF*, of *S* based on significant words (Croft *et al.*, 2010) in *S*, denoted *SF*(*S*), and is defined as:

$$SF(S) = \frac{|significant\ words|^2}{|S|} \tag{1}$$

where $|S|$ is the number of words in *S* and $|significant\ words|$ is the number of significant words in *S*. A word *w* in *C* is *significant* in *C* if:

$$f_{C,w} \geq \begin{cases} 7 - 0.1 \times (25 - Z) & if\, Z < 25 \\ 7 & if\, 25 \leq Z \leq 40 \\ 7 + 0.1 \times (Z - 40) & otherwise \end{cases} \tag{2}$$

where $f_{C,w}$ is the *frequency of occurrence* of *w* in *C*, *Z* is the number of sentences in *C* and 25 and 40 are the predefined low and high-frequency cutoff values, respectively.

- The *similarity score* of a sentence S_i in *C*, denoted $Sim(S_i)$, indicates the relative degree of S_i in capturing the overall semantic content of *C*. Q_{Sum} computes $Sim(S_i)$ using the *word-correlation factors* (*wcf*) Qumsiyeh and Ng (2013) of every word in S_i and words in each remaining sentence S_i in *C* and the *Odd ratio* $= p/1 - p$ (Luger, 2008).

$$Sim(S_i) = \frac{\sum_{j=1,\, i \neq j}^{|C|} \sum_{k=1}^{n} \sum_{l=1}^{m} wcf(w_k,\ w_l)}{1 - \sum_{j=1,\, i \neq j}^{|C|} \sum_{k=1}^{n} \sum_{l=1}^{m} wcf(w_k,\ w_l)} \tag{3}$$

Where $|C|$ is the number of sentences in *C*, *n* (*m*, respectively) is the number of words in S_i (S_j, respectively), w_k (w_l, respectively) is a word in S_i (S_j, respectively) and the *Odds ratio* is applied to the *odds of (non-)occurrence* of keywords in S_i and *C*.

- *Label-Sentence Similarity* (*LSS*) measures the *similarity* between *S* in *C* and the *cluster label L* of C and is computed using the Vector Space Model as follows:

$$LSS(S) = sim(L,S) = \frac{\Sigma_{i=1}^{N} w_{i,S} \times w_{i,L}}{\sqrt{\Sigma_{i=1}^{N} w_{i,S}^{2}} \times \sqrt{\Sigma_{i=1}^{N} w_{i,L}^{2}}} \quad (4)$$

where $w_{i,S}$ ($w_{i,L}$, respectively) is the weight of word i in S (L, respectively) and is defined as $w_{i,S} = tf(i,S) \times idf(i)$ ($w_{i,L} = tf(i,L) \times idf(i)$, respectively), $idf(i) = log_2 N/N_i$, N_i is the number of sentences in C that includes word i and N is the total number of distinct keywords in C. The *higher* the *LSS* value of S is, the *higher* is the degree of S in reflecting the topic T covered in C, as L captures T of the documents in C.

- *Named Entity* (*NE*) is the *name-entity weight* of S in C, which is defined as:

$$NE(S) = \frac{\Sigma_{i=1}^{|E|} f(E)}{f(E)} \quad (5)$$

where a named entity is an atomic element, which can be the name of a person, an organization, a location, etc., $|E|$ is the number of named entities in S, $f(E_i)$ is the frequency of occurrence of entity E_i in C and $f(E)$ is the sum of the frequency of occurrence of all named entities in C. A sentence that contains a named entity usually captures more useful information in a document than sentences that do not Osinski (2006). Q_{Sum} uses the Stanford Name Entity Recognizer (http://nlp.stanford.edu/software/CRF-NER.shtml) in detecting name entities in a document.

- A penalty is given to each short sentence (with less than 15 words) or long sentence (with more than 30 words) Schiffman *et al.* (2002), as *short* sentences often require some introduction, reference resolution or some kind of interjection, whereas *long* sentences often cover multiple concepts that can be found elsewhere in single sentences in C. Q_{Sum} computes the *Sentence Length*, denoted *SL*, of S as:

$$SL(S) = \begin{cases} -1 & if\ |S|<15\ or\ |S|>30 \\ 0 & otherwise \end{cases} \quad (6)$$

where $|S|$ is the number of (stop)words in S.

- It has been shown that the *first* sentence of the *first* paragraph and the *last* sentence of the *last* paragraph contain the most important words (information) in a document (Baxendale, 1958). Q_{Sum} defines the *Sentence Position* (*SP*) value to S as:

$$SP(S) = \begin{cases} 1, & if\ S\ is\ the\ 1st\ sentence\ of\ the\ 1st\ paragraph\ or\ the\ last \\ & sentence\ of\ the\ last\ paragraph\ in\ any\ document \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

Using the *Stanford Certainty Factor* Luger (2008), Q_{Sum} computes the *relative degree of significance* (*RS*) of S in capturing the contents of C based on the features introduced above.

$$RS(S) = \frac{TiF(S) + SF(S) + Sim(S) + LSS(S) + NE(S) + SL(S) + SP(S)}{1 - Min\{T\,iF(S), SF(S), Sim(S), LSS(S), NE(S), SL(S), SP(S)\}} \quad (8)$$

Since *TiF*(*S*), *SF*(*S*), *Sim*(*S*), *LSS*(*S*), *NE*(*S*), *SL*(*S*) and *SP*(*S*) are in different scales, they are normalized to the same range using a logarithmic scale before *RS*(*S*) is computed.

3.2.4 Solving the redundancy and coverage problems. Before selecting sentences for creating the summary *Sum* of a document cluster *C*, Q_{Sum} clusters the top-*M* (≥1) ranked sentences (based on their *RS* scores) in *C*, where $|M|$ is *nine* times the length of *Sum*, using the HAC algorithm. The HAC algorithm initially assigns each sentence to a *singleton* sentence cluster. Hereafter, it repeatedly merges sentence clusters until a specified termination criterion is satisfied. As the HAC algorithm relies on a *similarity metric* among sentences in any two sentence clusters for merging clusters, Q_{Sum} uses the *Sim* measure, as defined in equation (3) with the first summation removed, to compute the similarity between any two sentences in two (intermediate) sentence clusters. To determine the termination criterion for HAC, Q_{Sum} implements the algorithm in Alguliev and Alyguliev (2008) to define the *optimal* number of sentences covered in a set of documents, which dictates the *ideal number* of *sentence clusters* in C to be generated by HAC.

In general, Q_{Sum} selects sentences from each *sentence cluster ST* created by HAC to be included in the summary *Sum* of *C*. The first sentence *S* to be chosen is from a *ST* with the highest *RS* value in *C* and the sentence with the highest *RS* value in each remaining sentence cluster is chosen in order. After the first round of selection, Q_{Sum} chooses the next sentence *S*' from each *ST* with the *lowest similarity score* relative to its first chosen sentence *S*, which is computed as the *sum* of the word-correlation factors between each word in *S*' and *S*. Using this selection strategy, Q_{Sum} ensures that selected sentences are *distinct* in contents, which avoids *redundancy* and maximizes the *coverage* of the information included in *Sum*. The selection terminates whenever the length of the newly selected sentence and other sentences that are already included in *Sum* exceeds *Size*.

3.2.5 Adding the temporal dimension. The information captured in a set of documents on a particular topic might have been dynamically changed over time, such as an incident in news. An updated document contains the most recent development (i.e. information) compared with its older editions. Q_{Sum} accounts for the *temporal dimension* in a set of documents by re-weighting each sentence in a document based on its *timestamp* (the date when it was last updated). The *RS* weight of each sentence *S* is modified based on its temporal dimension weight, denoted *TD*(*S*).

$$RS_T(S) = RS(S) \times TD(S) \tag{9}$$

where *S* is a sentence in a document cluster *C*, and *TD*(*S*) is a time-based weight of *S*. The *earlier* a document in *C* which includes *S* is published, the *smaller* the *TD*(*S*) is. As *exponential average* is extensively used in time-series prediction, Q_{Sum} uses the *decay rate formula* in computing *TD*(*S*), which decreases the sentence weight exponentially based on *time* (Yu *et al.*, 2005) and is defined as:

$$TD\,(S) = DecayRate^{\frac{y-t}{24}} \tag{10}$$

where *y* is the current time (i.e. day, hour and minute), *t* is the publication time of the document including *S*[5], (*y* – *t*) is the time gap in hours and *DecayRate* is a variable experimentally set to 0.5 (Yu *et al.*, 2005).

We have made it an option to include (exclude, respectively) the temporal dimension as a *feature* to compute *RS* of *S* and treat it as a separate *weighting factor* in determining the ranking of *S* in *C* prior to selecting sentences in *C* to generate the summary of *C*. This option is appropriate, as a given set of documents may not discuss events that override one another, i.e. old information are just as important as new ones.

3.2.6 Generating summaries through $\mathrm{Q}_{\mathrm{Sum}}$*-interface.* The user *U* who has submitted a query *Q* can:

- view all the relevant topics (captured by cluster labels) of *Q*;
- click on a cluster label *T* to examine all the documents clustered under *T*; and
- request Q_{Sum} to generate the summary *Sum* of the documents on *T*.

(See, as an example, Figure 2 which shows the top-10 cluster labels and the top-five documents in the "World News" cluster.) The created summary is a collection of sentences, each of which is included in one of the documents in the cluster labeled *T* and chosen according to the summarization approach of Q_{Sum}. By clicking on any sentence *S* in *Sum*, *U* can view the content of the document *D* in which *S* resides, which allows *U* to access detailed information covered in *D*, a unique feature of Q_{Sum}.

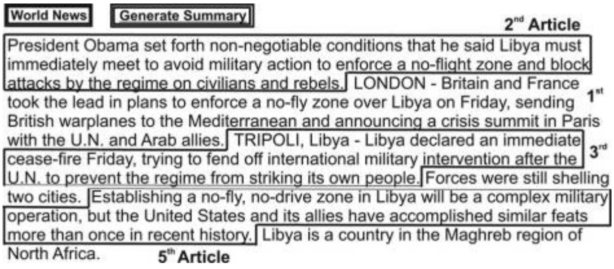
Example 1 Figure 3 shows the summary *Sum* generated using the documents in the "World News" cluster, along with the titles and snippets of the first six documents in the cluster as partially displayed in Figure 2. *Sum*:

- includes *distinct sentences* with different information such that sentences with *older* dates are ranked towards the bottom;
- covers most *subtopics* associated with Libya in the news, which include the *military action*, *summit meeting*, *political agenda*, for the events developed in Libya;
- does not include any sentences with *unidentified (pro)nouns*; and
- is *appropriate* in length (10 per cent of the size of the documents in the cluster).

The first sentence in *Sum* is extracted from the second document, whereas the second sentence is from the first article in the cluster. As it turns out, the third to sixth sentences in *Sum* as shown in Figure 3 are extracted from sentences in the sentence clusters in the corresponding order. □

4. Experimental results

To assess the performance of Q_{Sum}, we first determined the data sets used for the empirical study and chose the statistical approach that identifies the ideal number of



President Obama set forth non-negotiable conditions that he said Libya must immediately meet to avoid military action to enforce a no-flight zone and block attacks by the regime on civilians and rebels. LONDON - Britain and France took the lead in plans to enforce a no-fly zone over Libya on Friday, sending British warplanes to the Mediterranean and announcing a crisis summit in Paris with the U.N. and Arab allies. TRIPOLI, Libya - Libya declared an immediate cease-fire Friday, trying to fend off international military intervention after the U.N. to prevent the regime from striking its own people. Forces were still shelling two cities. Establishing a no-fly, no-drive zone in Libya will be a complex military operation, but the United States and its allies have accomplished similar feats more than once in recent history. Libya is a country in the Maghreb region of North Africa.

Figure 3. The summary generated by Q_{Sum} for the documents in the cluster labeled "world news"

appraisers and queries required for validating the grammatical correctness, referential clarity, anti-redundancy, structure and coherence and responsiveness quality of Q_{Sum}-generated summaries. We have also compared the time to locate information between Q_{Sum} and Google and measured the time for generating summaries using Q_{Sum}.

4.1 The data sets

In this section, we present the data sets used for analyzing the quality of Q_{Sum}-created summaries.

Generic multi-document summarization analysis has been one of the designated tasks of DUC 2005, DUC 2006 and DUC 2007, each of which is an open benchmark data set created and archived by the DUC (nlpir.nist.gov/projects/duc/). We used all three data sets for evaluating Q_{Sum}-generated summaries. Table I provides a summary of the three data sets, where TDT (projects.ldc.upenn.edu/TDT/) and AQUAINT (ldc.upenn.edu/Catalog/docs/LDC2002T31/) are corpora from where the DUC data sets are extracted.

NIST assessors, who organized DUC and created each data set as shown in Table I, selected various topics and chose a set of Web documents relevant to each topic. Given a DUC topic T and a collection of documents C relevant to T, a summarization approach to be evaluated is expected to create a brief (approximately 10 per cent of the size of C in our case), well-organized and fluent summary that captures the key concepts covered in C on T. The summary is compared with the *reference summaries* of C, which were created by NIST assessors, to analyze its quality.

4.2 Number of appraisers and test queries used for the controlled experiments

We first determine the ideal number of appraisers and test queries to be used in evaluating Q_{Sum} so that the performance evaluation is reliable and objective.

4.2.1 The number of appraisers. In statistics, two types of errors, Types I and II, are defined (Jones and Kenward, 2003). Type I errors, also known as a errors or *false positives*, are the *mistakes* of *rejecting* a null hypothesis when it is true, whereas Type II errors, also known as β errors or *false negatives*, are the *mistakes* of *accepting* a null hypothesis when it is false. We apply the formula (Jones and Kenward, 2003) below to determine the ideal number of appraisers, n, which is dictated by the probabilities of occurrence of Types I and II errors, to evaluate Q_{Sum}-created summaries.

$$n = \frac{\left(Z_{\frac{\alpha}{2}} + Z_{\beta}\right)^2 \times 2\sigma^2}{\Delta^2} + \frac{\left(Z_{\frac{\alpha}{2}}\right)^2}{2} \tag{11}$$

where Δ is the *minimal expected difference* to compare Q_{Sum} with Google, which is set to 1 in our study, as we expect Q_{Sum} to perform as good as Google in terms of

Table I. DUC data sets used for evaluating the quality of Q_{Sum}-created summaries

Dataset	DUC 2005	DUC 2006	DUC 2007
Number of clusters	50	50	45
Number of Docs/cluster	32	25	25
Data source	TDT	AQUAINT	AQUAINT

generating high-quality summaries in comparison with document titles and snippets created by Google, respectively; σ^2 is the *variance*[6] of the generated summaries, which is 3.82 in our study; α (β, respectively) denotes the probability of making a Type I (II, respectively) error, which is set to be 0.05 (0.20, respectively) and $1 - \beta$ determines the probability of a false null hypothesis that is correctly rejected, and Z is the value assigned to the standard *normal distribution* of generated summaries. Based on the standard normal distribution, when $\alpha = 0.05$, $Z^{\alpha} = 1.96$, and when $\beta = 0.20$, $Z_{\beta} = 0.84$.

We conducted an experiment using a randomly sampled 100 test queries extracted from the *AOL query log*[7] to determine the value of σ^2. We chose only 100 queries, as the *minimal expected difference* and *variance*, which are computed on a *simple random sample*, do not change with a larger sample set of queries. σ^2 is computed by averaging the sum of the square difference between the mean and the actual number of *useful* summaries[8] created for each one of the 100 test queries. We obtained 3.82, which is the value of σ^2 for cluster summaries.

The values of α and β are set to be 0.05 and 0.20, respectively, which imply that we have 95 per cent *confidence* on the correctness of our analysis and that the *power* (i.e. probability of avoiding false negatives/positives) of our statistical study is 80 per cent. According to Kazmier (2003), 0.05 is the commonly used value for a, whereas 0.80 is a conventional value for $1 - \beta$, and a test with $\beta = 0.20$ is considered to be statistically powerful. Based on the values assigned to the variables in equation (11), the ideal number of appraisers for our study is:

$$n = \frac{(1.96 + 0.84)^2 \times 2 \times 3.82}{1^2} + \frac{1.96^2}{2} \cong 62 \tag{12}$$

The results collected from the 62 appraisers are expected to be comparable with the results that are obtained by the actual population (Jones and Kenward, 2003), i.e. Web users who query Web search engines.

4.2.2 The number of test queries. To determine the ideal number of test queries to be included in the controlled experiments, we rely on two different variables:

(1) the *average attention span* of an adult; and

(2) the *average number of search queries* that a person often creates in one session when using a Web search engine.

As mentioned in Rozakis (2002), the average attention span of an adult is between 20 and 30 minutes. Furthermore, Jansen *et al.* (2000), who have evaluated Web users' behavior especially on:

- the amount of time Web users spend on a Web search engine;
- the average size of users' queries; and
- the average number of queries submitted by a user, estimate that the average number of queries created by each user in one session on a Web search engine is approximately 2.8.

Based on these studies, each appraiser was asked to evaluate Q_{Sum} using *three* queries, as evaluating the summaries on the retrieved results of each one of the three queries takes approximately *30* minutes, which falls into an adult time span. We randomly selected *186* (= 62 × 3) queries from the AOL query log for evaluating Q_{Sum}-created summaries.

4.3 Performance measures of $\mathrm{Q_{Sum}}$

We have developed various applications on Facebook for its appraisers to evaluate the *quality* of each Q_{Sum}-created *summary*. Facebook appraisers were used, as Facebook is a social network with users diverse in nationalities, ages, genders and cultures who can provide unbiased evaluations.

Using the DUC 2005, 2006 and 2007 data sets and an evaluation guideline, which is a set of *quality questions* developed in 2001 (Lin and Hovy, 2002), a summary created by a summarization system can be evaluated. These questions address the quality of *grammaticality*, *non-redundancy*, *referential clarity*, *structure* and *coherence* and *responsiveness* of a generated summary. These qualities are measured on a five-point scale as suggested by DUC. We have posted on Facebook:

- the 186 queries extracted randomly from the AOL query logs;
- their respective Q_{Sum}-created summaries; and
- the set of quality questions for the appraisers to evaluate.

We have also considered the ROUGE toolkit (version 1.5.5), which is widely adopted for *summary evaluation*. ROUGE measures the quality of a summary by counting the *overlapped* units between a generated summary *Sum* and a set of reference summaries created by DUC experts using the same set of documents. The *higher* the ROUGE score is, the *better* the summarization method that generates *Sum* performs. The *n*-gram ROUGE score is defined as:

$$ROUGE_n = \frac{\sum_{R \in Ref\,Sum} \sum_{n-gram \in R} Count_{match}(n_{gram})}{\sum_{R \in Ref\,Sum} \sum_{n-gram\, \in R} Count(n_{gram})} \tag{13}$$

where n (≥1) is the size of the (overlapped) *n*-gram, $Count_{match}\,(n_{gram})$ is the number of *overlapped n*-grams in *Sum* and the set of reference summaries *Ref Sum* and *Count* (n_{gram}) is the number of *n*-grams in the set of reference summaries. We computed ROUGE-2 (unigram-based and bigram-based co-occurrence statistics), ROUGE-SU4 (trigram and four-gram-based co-occurrence statistics) and ROUGE-BE (all co-occurrence statistics such that matched keywords have the same part of speech tag), as the DUC Web site includes the ROUGE-2, ROUGE-SU4 and ROUGE-BE scores of 30 multi-document summarization systems for each data set, which we compare with Q_{Sum}-generated summaries.

4.4 Performance evaluation of $\mathrm{Q_{Sum}}$

In this section, we present the experimental results that quantify the performance of Q_{Sum} on generating high-quality summaries. A Facebook appraiser evaluates the grammar, anti-redundancy, referential clarity, coherence and responsiveness of a summary *Sum*, whereas the ROUGE score, as introduced earlier, reflects the amount of information covered in *Sum* that address the corresponding query (topic) substantially.

We have collected the responses on the *quality questions* of each Q_{Sum}-created summary on documents in the DUC data sets, i.e. DUC 2005-2007, which were provided by the 62 Facebook appraisers who reviewed the summaries in response to the 186 test queries. The results are obtained by the comparisons of contents captured in the Q_{Sum}-generated summaries with the ones in the *reference summaries* created by the DUC experts on the same set of documents. In addition, we have also compared the various ROUGE scores of Q_{Sum}-created summaries with the ones achieved by the *30* automated multi-document summarization systems participated in DUC as depicted in Table II.

As demonstrated in Table II, Q_{Sum} achieves the highest score on *non-redundancy*, second highest on *referential clarity* and *responsiveness*, fourth on *structure and coherence* and fifth on *Grammar*. The comparatively lower scores on grammar, besides structure and coherence, among the five quality measures are due to the fact that the summarization approach of Q_{Sum} is *extractive*, which is not sophisticated in summarizers connecting (i.e. combining) extracted sentences in a summary. This is not a major drawback, as Q_{Sum} is ranked in the top 5 on each measure among the 30 summarizers.

Table II also shows that Q_{Sum} achieves the second (third, respectively) highest ROUGE-2 and ROUGE-SU4 (ROUGE-BE, respectively) score(s) among the 30 summarizers involved in the evaluation. This indicates that the information included in Q_{Sum}-created summaries are of high quality, i.e. Q_{Sum}-generated summaries address a user query in a substantial way, compared with other lower ranking summarizers. Note that none of the 30 summarizers outperforms the others, including Q_{Sum}, in all the three ROUGE scores.

4.4.1 $\mathrm{Q_{Sum}}$ *versus Google.* We have analyzed the evaluations provided by 62 Facebook appraisers who have compared the *time* and *extracted results* in locating desired information retrieved by Q_{Sum} and Google, respectively, on each one of the 186 test queries (as described earlier). The evaluations show that it takes a Facebook appraiser an average of 63 (72, respectively) seconds to locate the *desired information* on Google (Q_{Sum}, respectively) based on the test queries.

We created another two Facebook applications, App_1 and App_2, which include a number of performance evaluation questions for another group of Facebook appraisers, other than the 62 appraisers mentioned earlier. The applications were posted under Facebook for the appraisers to provide their feedbacks.

For App_1, the application includes two pages in a panel, the *left* page displayed the (traditional) top-10 results generated by Google on a query arbitrarily created by an appraiser, whereas the *right* one is the Q_{Sum}-created summary of the ten documents shown on the left page. The purpose of this study is to analyze whether Q_{Sum}-generated summaries are really useful to its users who browse through search results and enrich

Quality/rouge measures	Achieved by Q_{Sum}	Outperformed by	Outperform
Grammar	4.35	5	25
Anti-redundancy	4.81	1	29
Referential clarity	4.01	2	28
Structure and coherence	3.15	4	26
Responsiveness	4.33	2	28
ROUGE-2	0.12	2	28
ROUGE-SU4	0.17	2	28
ROUGE-BE	0.06	3	27

Table II. Comparing the quality of Q_{Sum}-created summaries with the reference summaries created by the 30 DUC summarizers

their search experiences. After submitting a query and examine the results displayed on each (left/right) page, an appraiser responded to each of following questions:

Q1. On which system did you spend less time locating the intended information?

Q2. Did the system on the left offer vital information not contained in the system on the right?

For the first question, the responses are 12 per cent for Google, 6 per cent for Q_{Sum} and 82 per cent for the same, whereas for the second question, 27 per cent said "Yes" and 73 per cent said "No". Based on the responses, we conclude that the appraisers have found Q_{Sum}-generated summaries to be *useful* and *informative* compared with the traditional results retrieved by Google. Altogether, there are 288 responses to App_1.

For $App_{2,}$ the application requires the involved appraisers to:

- first *identify a task* that each one often performs on a search engine;
- *create a query* that represents the task; and
- *submit the query* to both systems (Google and Q_{Sum}).

Hereafter, the appraisers were asked to answer the question, "Which system helped you perform this task faster?" The tasks (which were clustered based on their similarity), the number of responses for each type of tasks and their answers to the question are shown in Table III in which each highlighted number indicates the preference of the appraisers on either Google or Q_{Sum} in accomplishing the corresponding task. The responses have verified that Q_{Sum}-created summaries on results of queries for different tasks were highly regarded by Facebook appraisers than the results generated by Google, with the exception of the two tasks, "Find Tools/Software" and "Navigate to a Site". The results are anticipated, as Q_{Sum}-created summaries include information on products but exclude URL links to download them, which are provided in the results generated by Google for its users to access. Moreover, finding the URL of a Web site *W* using its name provided by the user is a strength of Google, while a summary on *W* offers no such value. There are 58 responses to App_2.

Even though the empirical study of App_2 reflects that Q_{Sum} cannot handle navigation-type Web queries, an online report published by Wordtracker (www.top-keywords.com/longterm.html) on February 2, 2015, shows that of the top 500 most popular query keywords created by Web search engine users, only 51 of them include keywords explicitly specify a Web site, such as facebook.com, amazon.com and ebay.com. The report illustrates that the percentage of navigation-typed Web queries is not a dominating type of commonly used Web queries.

4.4.2 Query processing time of $\mathrm{Q}_{\mathrm{Sum.}}$ We have measured the *processing time* of creating a summary using Q_{Sum} based on the 186 queries from the AOL query log. The

Table III. Facebook appraisers' responses to different tasks posted as queries under Google and Q_{Sum}

Tasks (posted as queries on Google and Q_{Sum})	No. of responses	Prefer Google	Prefer Q_{Sum}
Research a topic	9	3	**6**
Find news on an event	11	3	**8**
Find answers to questions	5	**3**	2
Find information on an item	17	6	**11**
Find tools/software	8	**7**	1
Navigate to a site	8	**8**	0

processing time required to generate a summary is less than *2 seconds* on an average. While a Q_{Sum} user is viewing a summary generated for the documents in a cluster, summaries of other clusters are created in sequence behind the screen, which is a time-saving process.

Q_{Sum} is implemented on an intel dual core desktop with dual 2.66 GHz processors, 3 GB RAM size and a hard disk of 300 GB running under the Windows XP operating system.

5. Conclusions

Current Web search engines offer users a mean to locate desired information available on the Web. In response to a user query, current Web search engines, such as Google, Bing and Yahoo!, retrieve a list of ranked documents and display each with a title and a snippet to help users quickly identify the document(s) of interest. However, whenever a user query is *ambiguous*, it is very difficult, if not impossible, for a search engine to determine precisely the set of documents that satisfy the user's information need. Moreover, as snippets are created using sentences/phrases in the corresponding retrieved documents in which the keywords in the user query also appear, they may not capture the document contents and are similar to one another and thus are not useful in distinguishing their differences. To enhance Web search, we have developed Q_{Sum} which summarizes the contents of each clustered set of documents on a specific topic related to a query to assist its users in identifying results of interest. Q_{Sum} is a contribution to the Web search community, as it handles the ambiguous problem of a search query by creating summaries in response to different interpretations of the search which offer a "road map" to assist users to quickly identify information of interest. Experimental results using well-known datasets and Facebook applications show that Q_{Sum} creates high-quality summaries. The results verify that Q_{Sum} is an elegant Web search tool.

For future work, we plan to extend Q_{Sum} so that it can process user queries in multiple languages other than English. The extension requires that Q_{Sum} to be equipped with models that recognize natural language encoding schemes and handle internationalization.

Notes

1. A snippet of a document *D* is treated as a summary of *D*.
2. The Text Analysis Conference (nist.gov/tac) recommends a multi-document summary with the length of Size.
3. Word-correlation factors quantify the similarity (degree of closeness) of two words in terms of their semantic meaning.
4. End-of-sentence punctuation marks, such as periods, question marks and exclamation points, are less ambiguous as end-of-sentence indicators. However, as a period is not exclusively used to indicate sentence breaks, which may indicate an abbreviation, a decimal point, parts of an e-mail address, etc., a list of common abbreviations, such as "i.e.", "u.s.", and "e.g.", are maintained to minimize the detection errors.
5. If a sentence contains a date, then it overrides the publication time of the document, as it explicitly states the time of the information presented in the sentence.
6. *Variance* is widely used in statistics, along with standard deviation (which is the square root of the variance), to measure the average dispersion of the scores in a distribution.

7. The logs of AOL (gregsadetsky.com/aol-data/) include 50 million queries created by millions of AOL users between March 01, 2006 and May 31, 2006, and the AOL logs are available for public use.
8. A summary is considered *useful* if it is of high quality (4 or 5 on a five-point scale) as defined by DUC.

References

Alguliev, R. and Alyguliev, R. (2008), "Automatic text documents summarization through sentences clustering", *Automation and Information Science*, Vol. 40 No. 6, pp. 53-63.

Altman, A. and Tennenholtz, M. (2005), "Ranking systems: the pagerank axioms", *Proceedings of the 13th ACM Conference on Electronic Commerce (ACM EC), Valencia, 4-8 June*, pp. 1-8.

Amini, M. and Usunier, N. (2009), "Incorporating prior knowledge into a transductive ranking algorithm for multi-document summarization", *Proceedings of the International ACM Conference on Research and Development in Information Retrieval (ACM SIGIR), Boston, MA, 19-23 July*, pp. 704-705.

Arora, R. and Ravindran, B. (2008), "Latent dirichlet allocation based multi-document summarization", *Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data (AND)*, *Singapore*, *24 July*, pp 91-97.

Baxendale, P. (1958), "Machine-made index for technical literature - an experiment", *IBM Journal of Research and Development (JRD*), Vol. 2 No. 4, pp. 354-361.

Bhandari, H., Shimbo, M., Ito, T. and Matsumoto, Y. (2008), "Generic text summarization using probabilistic latent semantic indexing", *Proceedings of the International Joint Conference on Natural Language Processing (JCNLP)*, *Hydrabad*, *7-12 January*, pp. 133-140.

Croft, B., Metzler, D. and Strohman, T. (2010), *Search Engines: Information Retrieval in Practice*, Addison Wesley, Boston.

Dunlavy, D., O'Leary, D., Conroy, J. and Schlesinger, J. (2007), "QCS: a system for querying, clustering, and summarizing documents", *Information Processing & Management (IPM*), Vol. 43 No. 6, pp. 1588-1605.

Jansen, B., Spink, A. and Saracevic, T. (2000), "Real life, real users and real needs: a study and analysis of user queries on the web", *Information Processing & Management (IPM*), Vol. 36 No. 2, pp. 207-227.

Jones, B. and Kenward, M. (2003), *Design and Analysis of Cross-Over Trials*, 2nd ed., Chapman and Hall, London.

Kazmier, L. (2003), *Schaum's Outline of Business Statistics*, McGraw-Hill, New York, NY.

Leskovec, J., Grobelnik, M. and Milic-Frayling, N. (2004), "Learning sub-structures of document semantic graphs for document summarization", *Proceedings of the Workshop on Link Analysis and Group Detection (LinkKDD-2004), Seattle, WA, 22-25 August*, pp. 133-138.

Lin, C. and Hovy, E. (2002), "From single to multi-document summarization: a prototype system and its evaluation", *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), Stroudsburg, PA*, pp. 457-464.

Luger, G. (2008), *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, 6th ed., Addison-Wesley, Boston.

Osinski, S. (2006), "Improving quality of search results clustering with approximate matrix factorisations", *Proceedings of the annual European Conference on Information Retrieval (ECIR 2006)*, *London*, pp 167-178.

Ou, S. Khoo, C. and Goh, D. (2006), "Automatic multi-document summarization for digital libraries", *Conference on Library & Information Education & Practice in Proceedings of the Asia-Pacific(ALIEP), London, 10-12 April*, pp 72-82.

Qumsiyeh, R. and Ng, Y.K. (2013), "Enhancing web search using query-based clusters and labels", *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI'13), Atlanta, GA, 17-20 November*, pp. 159-164.

Rozakis, L. (2002), *Test Taking Strategies and Study Skills for the Utterly Confused*, McGraw Hill, New York, NY.

Schiffman, B., Nenkova, A. and McKeown, K. (2002), "Experiments in multidocument summarization", *Proceedings of the Human Language Technology Conference (HLT), pp.* 52-58.

Schlesinger, J., Leary, D. and Conroy, J. (2008), "Arabic/English multi-document summarization with classy - the past and the Future", *Proceedings of the Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Heidelberg*, pp. 568-581.

Shen, D. and Pan, R. (2006), "Query enrichment for web- query classification", *ACM Transactions on Information Systems (ACM TOIS*), Vol. 24 No. 3, pp. 320-352.

Yu, P., Li, X. and Liu, B. (2005), "Adding the temporal dimension to search – a case study in publication search", *IEEE/WIC/ACM International Conference on Web Intelligence Proceedings of the 2005 (WI), Compiègne, 19-22 September*, pp. 543-549.

Corresponding author
Yiu-Kai Ng can be contacted at: ng@compsci.byu.edu