2011-03-15

Easy to Find: Creating Query-Based Multi-Document Summaries to Enhance Web Search

Rani Majed Qumsiyeh
Brigham Young University - Provo

Easy to Find: Creating Query-Based Multi-Document Summaries to Enhance Web Search

Rani Qumsiyeh

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Yiu-Kai Dennis Ng, Chair
Christophe Giraud-Carrier
Mark Clement

Department of Computer Science

Brigham Young University

April 2011

ABSTRACT

Easy to Find: Creating Query-based Multi-Document Summaries to Enhance Web Search

Rani Qumsiyeh

Department of Computer Science

Master of Science

Current web search engines, such as Google, Yahoo!, and Bing, rank the set of documents *S* retrieved in response to a user query *Q* and display each document with a title and a snippet, which serves as an abstract of the corresponding document in *S*. Snippets, however, are not as useful as they are designed for, i.e., to assist search engine users to quickly identify results of interest, if they exist, without browsing through the documents in *S*, since they (i) often include very similar information and (ii) do not capture the main content of the corresponding documents. Moreover, when the intended information need specified in a search query is *ambiguous*, it is difficult, if not impossible, for a search engine to identify precisely the set of documents that satisfy the user's intended request. Furthermore, a document title retrieved by web search engines is not always a good indicator of the content of the corresponding document, since it is not always informative. All these design problems can be solved by our proposed query-based, web informative summarization engine, denoted *Q-WISE*. *Q-WISE* clusters documents in S, which allows users to view segregated document collections created according to the specific topic covered in each collection, and generates a concise/comprehensive summary for each collection/cluster of documents. *Q-WISE* is also equipped with a query suggestion module that provides a guide to its users in formulating a keyword query, which facilitates the web search and improves the precision and recall of the search results. Experimental results show that *Q-WISE* is *highly effective* and *efficient* in generating a high quality summary for each cluster of documents on a specific topic, retrieved in response to a *Q-WISE* user's query. The empirical study also shows that *Q-WISE*'s clustering algorithm is *highly accurate*, labels generated for the clusters are *useful* and often reflect the topic of the corresponding clustered documents, and the performance of the query suggestion module of *Q-WISE* is comparable to commercial web search engines.

Keywords: Clustering, Summarization, Query Suggestion

ACKNOWLEDGEMENTS

I owe my deepest gratitude to Dr. Erlend D. Peterson, the Associate International Vice President at Brigham Young University, for his continuous support throughout my educational journey as a Bachelor and a Graduate student here at BYU.

I also want to thank Dr. Christophe Giraud-Carrier for his feedback and suggestions for improving this work. I would also like to thank Dr. Giraud-Carrier and Dr. Mark Clement for their willingness to serve as my committee members.

Special thanks to my advisor, Dr. Yiu-Kai Ng, for being an incredible mentor, for his encouragement, patience and help in completing this work and throughout my time at BYU. His recommendations and suggestions have been invaluable for the project and for software improvement.

Last but not least, I would like to thank and give my sincere appreciation to my family Majed, Samar, Ashraf, and my friend Lina, who were always there for me.

To the memory of my Father, Majed Qumsiyeh

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With hundreds of thousands of new electronic documents added to the Web each day [Trout 10], it is essential for web search engines to continuously enhance their current mechanisms for extracting and ranking relevant results to meet web users' information needs effectively. Current web search engines rank retrieved documents based on their likely relevance to a user's query *Q* and display their titles with(out) a short snippet[1]. A snippet *S*, however, is (i) often very similar to the other snippets created for the same query, unless the search engine eliminates similar snippets [Goldstein 00] and (ii) created using sentences/phrases in the corresponding retrieved document solely based on the keywords that appear in *Q*, which may not capture the main content of the corresponding document. Moreover, a web search engine user often scans only a small number of snippets, usually in the range of 10 – 30 [Spink 00]. From the perspective of web users and search engine designers, web search engines must accurately capture the main content of retrieved documents in the form of summaries or a single summary, which provide(s) a snapshot view of the retrieved information to the users who can quickly draw a conclusion on their relevance with respect to their information needs.

Recently, document summarization systems have emerged which automatically create a summary of a document or set of documents based on a *search query* [Daume 05, Yih 07, Wang 09]. In these query-based (query-oriented) summarization systems, a summary is generated on (each of) the top-N (≥ 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

[1] A snippet, which is an abstract of a document, includes a couple of sentences intended to capture the content of the document and can be treated as a single-document summary [Yih 07].

generates a single summary on multiple documents, it is a *query-based multi-document summarization system*.

Multi-document summarization systems are regarded higher than their single-document counterparts [Khoo02]. The former provide an overview of the topics presented in a set of documents by (i) extracting mutual content across the documents while *avoiding repetition*, (ii) capturing *unique* (related, respectively) information in the documents, (iii) providing an overview of various subtopics, if they exist, of a particular subject, and (iv) identifying a subject or research topic that evolves over time. The automation of a multi-document summarization system is, however, a challenging task, since the system must (i) eliminate *redundancy*, i.e., same or similar information presented in different documents should be filtered, (ii) account for the *temporal dimension*, i.e., a new piece of information should override out-dated information, (iii) choose an ideal *compression ratio* to ensure that a summary includes sufficient contents of the corresponding documents in a reasonable length, (iv) achieve a (near-) complete *coverage*, i.e., capture the essential contents of the documents, and (v) resolve the *co-reference* issue among documents by detecting multiple references on the same subject [Kibble 00].

We propose to develop a *query-based web informative summarization engine*, called *Q-WISE*, which solves all the system problems listed above. *Q-WISE* allows novice, as well as expert, users to post a query *Q* and quickly locate the desired information, if they exist, captured in a short summary. *Q-WISE* (i) queries three major web search engines (Google, Yahoo!, and Bing) using *Q*, (ii) clusters the retrieved documents (on the same topic) in response to *Q* and assigns meaningful labels, which are extracted from phrases in the titles and snippets of the retrieved documents, to the clusters, and (iii) creates a single summary of each cluster, which includes the most representative and informative sentences in the cluster. Although the titles and

snippets are not effective in capturing the contents of their corresponding documents, they are used instead of the entire documents in our clustering algorithm, since (i) they have sufficient informative content for the clustering step and (ii) experiments show that using the titles and snippets keeps clustering effectiveness the same, but enhances efficiency.

A single summary for each cluster is appealing, since typical web search queries are *short* and often *ambiguous* in meaning [Shen 06], and a summary per query might not achieve conciseness while being comprehensive in capturing the contents presented in multiple retrieved documents. For example, if the search query is "tiger", the retrieved documents can be various in terms of their contents, which might include the Mac Operating system, the animal tiger, the golf player Tiger Woods, and fish, etc. Thus, a single summary created for the entire retrieved set of documents may be inaccurate and/or incomprehensive in handling the various topics.

On the other hand, Query Suggestion (QS), a query-creation interactive tool provided by many popular web search engines, such as Google, Yahoo!, and Bing, facilitates the web search by recommending concise queries that improves the precision and recall of the search engine [Vectomova 06]. Any search conducted by *Q-WISE* users is supported by its QS module which provides a guide to the users for formulating/completing a keyword query *Q* using suggested keywords (extracted from previous users' queries) as potential keywords in *Q*.

To determine the *effectiveness* and *efficiency* of our query-based web informative summarization engine, we have conducted two different performance evaluations. We first evaluate the effectiveness of *Q-WISE*'s clustering algorithm using three well-known datasets and compare it to other state-of-the-art clustering approaches. We also evaluate the quality of *Q-WISE*-generated summaries using three DUC datasets and compare *Q-WISE*-generated summaries against those created by thirty other state-of-the-art (query-based) multi-document

summarization tools. Moreover, we compare *Q-WISE* and Google in terms of the time required to locate desired information using the corresponding system. Furthermore, we conduct several controlled experiments to analyze (i) the query suggestions recommended by *Q-WISE* in response to a user query, (ii) the quality of *Q-WISE*-created cluster labels, and (iii) the quality of a *Q-WISE*-generated summary in terms of grammar, anti-redundancy, referential clarity, coverage, and structure and coherence.

Experimental results show that *Q-WISE* is *highly effective* and *efficient* in generating a concise and comprehensive summary of the documents retrieved in response to a *Q-WISE* user's query in each cluster and is ranked among the highest in its accuracy in multi-document summarization. The results also show that *Q-WISE*'s clustering approach achieves an *accuracy* in the ninetieth percentile. Moreover, the entire process from query submission till summary generation takes less than *four* seconds on average. Experiments conducted on the QS module of *Q-WISE* show that it is comparable to Google, Yahoo!, and Bing in terms of time required to generate recommended query suggestions. In addition, useful query suggestions, as determined by the controlled experiments, are ranked higher by *Q-WISE* than by Yahoo! and Bing.

Q-WISE is a contribution to the web and information retrieval community, since it (i) creates summaries, one for each potential topic derived from a user query, that is missing in a traditional web search engine, (ii) provides the user with an unbiased information source on a particular topic, since the creation of a summary is fully automated, without any editorial touch or subjective human intervention, (iii) enhances web searches by eliminating redundant retrieved information and helping the user locate desired information in time comparable to commercial web search engines, and (iv) establishes a new source for answering questions, since a summary,

which contains the most significant information from different documents, is likely to contain the answer to a user's question.

The remaining chapters in this thesis are organized as follows. In Chapter 2, we discuss works related to query suggestion, clustering, and multi-document summarization. In Chapter 3, we describe in detail the design of the query suggestion, clustering, and summarization modules of *Q-WISE*. In Chapter 4, we present the experimental results, which verify the *effectiveness* and/or *efficiency* of *Q-WISE*-generated query suggestions, cluster labels, clusters of retrieved documents on the same topic, and summaries of document clusters. In Chapter 5, we give a conclusion and include directions for future work.

Chapter 2

Related work

Q-WISE clusters and summarizes clustered documents retrieved by existing web search engines for a query constructed by its user with or without using its query suggestion tool. In this section, we discuss existing work related to query suggestion (in Section 2.1), clustering (in Section 2.2), and summarization (in Section 2.3).

2.1 Query suggestion

Recent studies on query suggestion focus on combining co-occurrence of keywords and a handcrafted thesaurus. Cao et al. [Cao 05] develop a language model that captures word relationships by utilizing WordNet, which is a hand-crafted thesaurus, and word co-occurrence computed using co-occurrence samples. Even though Liu et al. [Liu 04] achieve an improved performance on query suggestion using WordNet, words and their measures in WordNet are subjective and, unlike user query logs, do not capture (i) relationships among keywords from the user's perspective and (ii) updated keyword relationships through time. Ruch et al. [Ruch 06] present an argumentative feedback approach in which suggested query terms are selected from sentences classified into one of the four disjunct argumentative categories that have been regularly observed in scientific reports. Due to the increased time complexity of the approach in [Ruch 06] and being tailored for scientific reports, the feedback strategy is not scalable to the Web.

Cao et al. [Cao 08] introduce a query suggestion approach based on the contexts of queries recently issued by a user. Context-based query suggestion, however, requires very large query logs, since keywords suggested for a user query must appear in a list of related queries varying in size. Boldi et al. **[**Boldi 08] propose the *Query Flow Graph* (QFG), a directed graph in

which nodes are queries and any edge from node q_i to q_j indicates the probability of q_j being a suggestion for q_i. The authors of [Boldi 09, Bonchi 09] utilize a QFG for creating suggested queries using a random walk with restart model. Baraglia et al. [Baraglia 10] modify QFG to allow graphs to be updated to enhance their efficiency. These log-based methods [Boldi 08, Boldi 09, Bonchi 09] are adequate for queries that are created frequently, since accurate statistics are available on these queries. The statistics for infrequent queries, however, are based on only a few instances, which can lead to poor suggestions. Moreover, log-based methods rely on *training* to compute the edge weights, which is not required by *Q-WISE*.

Widely-used web search engines, such as Google, Yahoo!, and Bing, assist users with query suggestion. We have observed that (i) Google has the fastest query suggestion module and (ii) suggested keywords for a user's query that are recommended by existing web search engines do not seem to differ significantly from one to the other. The query suggestion module of *Q-WISE* is comparable to Google's in terms of computational time and the usefulness of its suggested queries.

2.2 Clustering

Clustering of web search results was first introduced in the Scatter-Gather system [Hearst 96]. Hereafter, a variety of clustering paradigms have been proposed, which include the singular value decomposition [Osinski 06], concept lattices [Chen 10], spectral clustering [Cheng 05], and graph theory [Xide 10]. Suffix Tree Clustering (SFC) [Crabtree 05, Ruixu 08], which uses recurring phrases to determine the similarity of web documents for clustering purpose, was later enhanced by [Chim 08], who minimize the size of a suffix tree. SnakeT [Ferragina 08] also uses an approach similar to the frequent phrases (known as frequent itemset) to extract meaningful labels and builds the cluster labels using a bottom-up hierarchy construction process. The

approaches presented in [Crabtree 05, Ferragina 08, Ruixu 08] perform clustering on web search results using the entire retrieved documents. Web search results, however, contain *non-relevant* terms in their navigational aids, such as keywords in advertisements and links to other pages, which could yield unreliable similarity measures of documents. *Q-WISE* uses the *titles* and *snippets* of documents retrieved by web search engines in clustering instead, which avoid *non-relevant* terms found in documents. In addition, cluster labels created by *Q-WISE* are generated using suffix arrays, which provide a space-conservative substitute to suffix trees commonly used for clustering web search results.

Besides phrase matching, graph-based and graph-partitioning methods are also available for clustering similar documents. Xide et al. [Xide 10] expand the set of retrieved snippets by including all the in- and out-linking pages to improve clustering precision. Since web search engines do not provide cheap access to the web graph, the link-retrieval efficiency is an issue. Microsoft [Zeng 04] develops a system that extracts (contiguous) sentences of variable length via regression. Regression, however, requires a training phase, which is non-feasible to apply to the heterogeneous web. In the context of *Q-WISE*, no training is required and clustering is done on-the-fly.

Kang and Kim [14] solve the ambiguity of short user queries by classifying queries for web document retrieval into one of three types of tasks: topic relevance task, homepage finding task, or a service finding task. Depending on the task, a different information emphasis is presented to the user. Although their method is successful in identifying whether a user query is referencing a service, a website, or a general topic, it fails to identify multiple interpretations and/or subtopics of a given topic query, which is still a main issue when dealing with ambiguous short queries.

2.3 Document summarization

Summarization methods can be classified into abstractive summarization (i.e., rewriting the summary, which usually requires information fusion) and extractive summarization (i.e., extracting keywords or sentences from documents to create a summary), and *Q-WISE* adopts the extractive summarization approach. The centroid-based (i.e., sentence-scoring and -ranking) method [Radev 04] is one of the most popular extractive summarization methods due to its simplicity and effectiveness. MEAD (www.summarization.com/mead) is centroid-based and scores sentences using sentence-level and inter-sentence features, such as TF/IDF. NeATS [Lin 02] is a multi-document summarizer based on SUMMARIST (http://128.9.208.230/must/), a single-document summarizer. MEAD and NeATS consider the sentence space but ignore the document-side knowledge, i.e., topics embedded in the documents. Sentence position, term frequency, topic signature, and term clustering have also been considered for selecting important content from documents for summarization purpose. In the context of *Q-WISE*, topics of documents are analyzed for creating a summary.

The authors of [Bhandari 08, Hennig 09] score sentences based on the representation of each sentence in the latent topic space provided by a trained Probabilistic Latent Semantic Analysis (PLSA) model. The authors pick the topics of the set of documents *S* to be summarized with the highest posterior probabilities and select sentences from *S* with the highest likelihood within a single topic to create a summary. Arora et al. [Arora 08] employ Latent Dirichlet Allocation to create multi-document summaries by selecting sentences from the topic with the largest likelihood. As compared with the summarization approach of *Q-WISE*, the systems mentioned above do not perform any *redundancy checking* and do not achieve *high coverage*, since they focus on sentences addressing the same topic.

Graph-based extractive summarization [Erkan 04, Mihalcea 04, Lin 09] uses weights of edges between sentences (represented as nodes) to capture their similarity. The PageRank algorithm [Altman 05], which is graph-based, determines the sentences that are the most salient in a collection of documents and closest to a given topic. Although effective in capturing the significant sentences in a document, graph-based methods do not account for multiple topics within a document. Leskovec et al. [Leskovec 04], who construct a document graph using subject-verb-object triples, semantic normalization, and co-reference resolution, consider node degree, PageRank, and Hubs to generate statistics for the nodes, which represent sentences, as attribute values in a machine learning algorithm to rank the sentences. Amini and Usunier [Amini 2009] present a transductive approach that learns the ranking function over sentences in retrieved documents with only a few labeled instances. Their approach outperforms classification models in sentence ranking. In the context of *Q-WISE*, no labeled instances or human-produced summaries are required, since no training is involved in its summarization, which minimizes the overhead and avoids the system scalability problem.

Chapter 3

Project description

During the process of generating query-based multi-document summaries, *Q-WISE* (i) assists the user in creating a query *Q* using a query suggestion dialog box, (ii) gathers the top 33 documents retrieved by each of the three major web search engines, Google, Yahoo, and Bing, in response to *Q*, since a collection of a hundred documents is an ideal set for generating good clusters and their corresponding summaries [Dunlavy 07], (iii) clusters the retrieved documents and assigns meaningful labels to the clusters, (iv) ranks the sentences in each cluster *C*, (v) groups the sentences in *C* using the Hierarchical Agglomerative Clustering (HAC) approach and the word-correlation matrix [Koberstein 06] as the similarity measure for HAC[2], and (vi) selects the highly ranked sentences that are distinct in content from the sentence clusters created in Step (v). Detailed explanation on each step is given below.

3.1 Overall view of the system

Since *Q-WISE* itself is not a search engine, the first step involves retrieving documents from major selected search engines in response to the input query and preprocessing these documents. Hereafter, *Q-WISE* produces phrases (labels) that capture the different (sub)topics in retrieved documents. The documents are then assigned to their corresponding topics identified by the corresponding labels. Finally, Q-WISE generates a short summary for each of the produced clusters.

The process of generating clusters and their corresponding summaries by *Q-WISE* can be decomposed into the following processing steps (as shown in Figure 1): (i) document retrieval, (ii) choosing cluster labels, (iii) ranking cluster labels, (iv) document clustering, (v)

[2] At the *document clustering phase*, documents are clustered according to their topics, whereas at the *sentence clustering phase*, sentences are clustered based on the subtopics of the documents in a cluster.

preprocessing of retrieved documents, (vi) co-reference resolution, (vii) sentence ranking, (viii) adding the temporal dimension, (ix) solving anti-redundancy and coverage, and (x) summary generation. Moreover, prior to these steps, *Q-WISE* helps the user formulate a query using a query suggestion module.

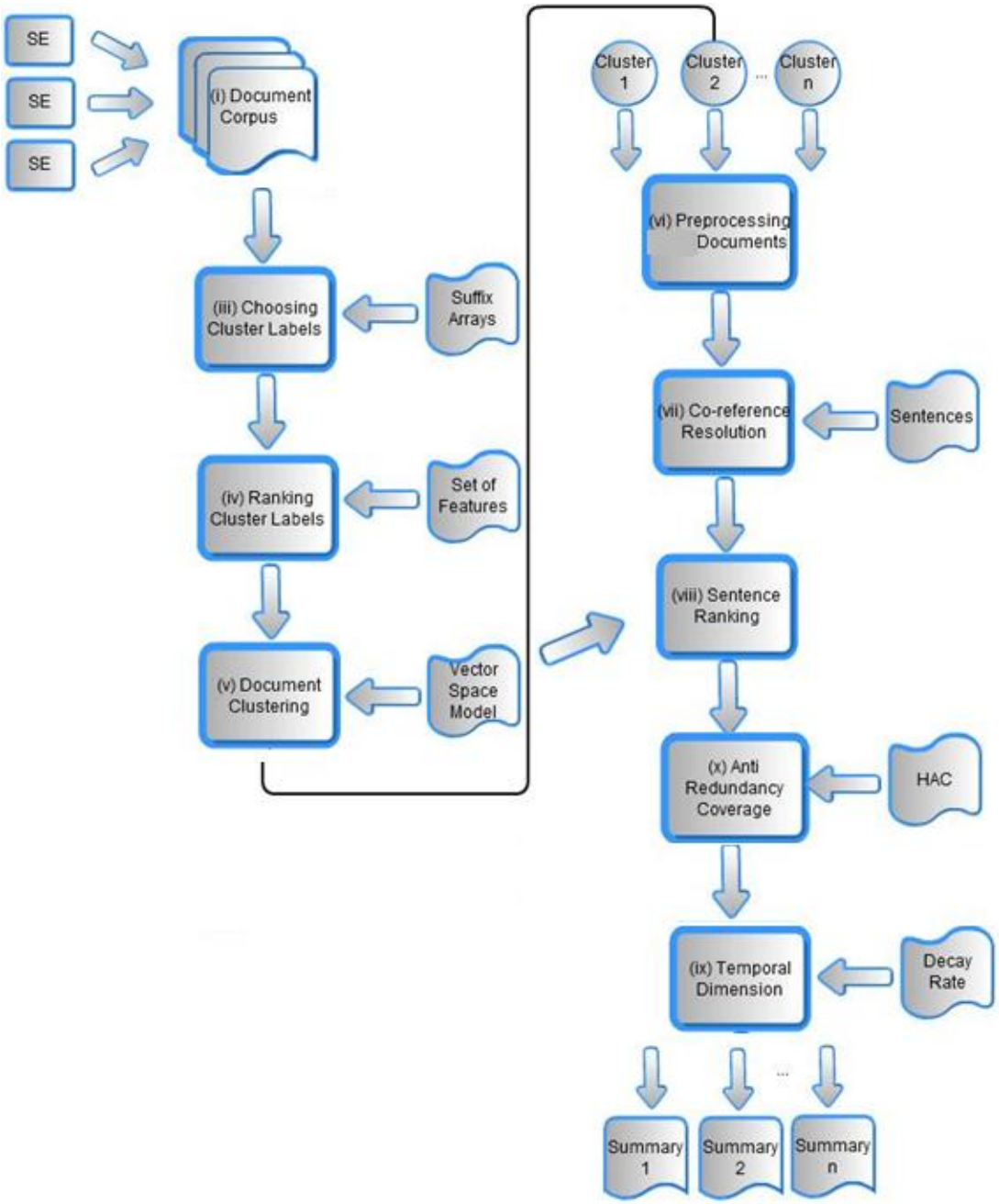


Figure 1. An overview of the process of *Q-WISE*

3.2 Query suggestion design

Web search engine users often provide imprecise specifications of their information needs in keyword queries, either because they are in a hurry, do not understand the search process well, or use inappropriate keywords. These scenarios might explain why web search engine users frequently create short queries[3], which are incomplete or ambiguous. Hence, providing a user interface that can assist the users in constructing keyword queries that capture their information needs is an important design issue of web search, which can significantly enhance the precision of retrieved results as a side effect [Anick 03].

One of the promising approaches for assisting the user in query construction is *query suggestion*. Query suggestion can be categorized into automatic and interactive. *Interactive query suggestion* displays keywords to be included in a query being created from where the user can choose one of the suggested queries or submit his own to the search engine. *Automatic query suggestion*, on the other hand, processes the user's query *Q* without providing suggested keywords to the user while the user is entering a query. Instead, *Q* is expanded internally using related keywords before it is processed by the search engine. Both approaches require a "query log" and a mechanism for deriving and ranking the suggested/expanded keywords. *Q-WISE* is equipped with an interactive query suggestion component, utilizing the AOL and Harold B. Lee Library query logs (discussed in details in Section 3.2.1) and a feature-based algorithm (see details in Section 3.2.2) that incorporates a trie structure for deriving and ranking suggested query words (see Section 3.2.3).

Major web search engines, such as Yahoo! and Google, offer query suggestion. However, their query suggestion mechanisms are based on morphological information of queries, i.e., co-

[3] The AOL query log shows that 84% of submitted queries are either a single-keyword or 2-keyword queries.

occurrence of a query word with other query words [Mei 08]. Although such query suggestion modules are useful in guiding the user through the process of constructing a query, occasionally, the suggested queries may not match the semantic of the query that the user intends to create. For example, an intended web search for "Harry Shum" would yield the suggested query "Harry Potter" after the first keyword has been entered, although the two are not related. *Q-WISE* only suggests query keywords if they are in the same query or the same AOL session within 10 minutes, which has been proven to be a reliable strategy [Fonseca 05].

3.2.1 The AOL/HBLL query logs

Q-WISE relies on the AOL and Harold B. Lee Library (HBLL) query logs to generate suggested queries. The logs of AOL (http://gregsadetsky.com/aol-data/) and HBLL (lib.byu.edu), respectively include queries created by millions of AOL users over a three months period between March 1 2006 and May 31 2006, and HBLL over two years between 2005 and 2007, and the AOL logs are available for public use. A query log includes a number of query sessions, each of which captures a period of sustained user activities on the corresponding search engine. Each AOL/HBLL session differs in length and includes a (i) user ID, (ii) the query text, (iii) date and time of search, and (iv) optionally clicked documents. (Figure 2 shows a query session from the AOL query log.) A *user ID*, which is an anonymous identifier of the user who performs the search, determines the boundary of each session (as each user ID is associated with one session), the *query text* are the words in a user query and multiple queries may be created under the same session, the *date and time of search* can be used to determine whether keywords in two or more queries created by the same user are within 10 minutes, the time period that dictates whether two queries should be treated as related, and *clicked documents* are retrieved documents that the user has clicked on and are represented and ranked by their titles by the corresponding search engine.

Words in queries and/or documents are either stopwords or non-stopwords. *Stopwords* are commonly-occurring keywords, which include prepositions, articles, and pronouns, etc., which carry little meaning and often do not represent the content of a document. From now on, unless stated otherwise, whenever we refer to "(key)words", we mean "non-stopwords".

The AOL query logs contain 50 million queries, while the HBLL contain 20 million queries. Out of these queries, about 30 million are unique and the rest are duplicates. The unique queries are examined and suggested keywords are extracted, whereas the duplicates are used as one of the features (discussed in Section 3.2.3) to determine the ranking of a suggested query.

User ID	Query	Date	Time	Clicked Documents
1326	back to the future	2006-04-01	17:59:28	http://www.imdb.com
1326	adr wheels	2006-03-28	12:53:39	
⋮				

Figure 2. An AOL query session

3.2.2 Processing the query logs

Q-WISE parses the AOL and HBLL query logs and retains the keywords in query texts in a trie data structure [Knuth 73], which is done once, and the constructed trie is 51 megabyte in size. Using the trie, candidate keywords suggested for a user query can be found and ranked dynamically. To suggest potential query keywords, *Q-WISE* locates a trie branch *b* containing the (letters in the) keywords that have been entered during the query creation process and extracts the subtree rooted at the last node of *b*. Hereafter, the extracted suggestions are ranked using a set of features (as defined in Section 3.2.3).

Q-WISE parses the query logs to extract query keywords while at the same time retains the information of *related* keywords in the same session, which were submitted by the same user within 10 minutes in the same session as discussed earlier. Hereafter, *Q-WISE* constructs the trie *T* using the extracted keywords in which each node *x* is labeled as a letter in an extracted

keyword in the given order, and each node in *T* is categorized as either "complete" or "incomplete". A *complete* node is the last node of a path on *T* representing an (a sequence of, respectively) extracted query keyword (keywords, respectively). If node *c* is a complete node, then T_c (the subtree of *T* rooted at *c*) contains all the suggested keyword(s) represented by the nodes in the path(s) leading from and excluding *c*. The possible number of suggestions of a particular keyword *K* is *n*, where n is the number of nodes in the (subtrees rooted at) T_c, which is determined by the number of complete nodes in T_c, since *K* can be combined with any number of the *n* suggested keywords. An *incomplete* node, on the other hand, is the first or a subsequent node of a path on *T*, which are labeled by the letters in keywords. If node *c* is an incomplete node, then all subsequent nodes of *c* up till the first complete node, if it exists, are the possible suggestions of the letter(s) and/or keyword(s) represented by the nodes in the path leading to and including *c*. The number of possible suggestions for a particular node *x* is *n*, where *n* is the number of *x*'s child nodes.

Example. Figure 3 shows a sample trie of a few queries in the query log. When the user enters the letters "TI" in a query, all branches rooted at the child nodes of node I are retrieved, which include "Time", "Ticket", and "Tiger", since node I is an incomplete node. If the user enters "TIG", then the keyword "Tiger" is displayed. If the user has entered "Tiger", the subtree rooted at node R, which is a complete node, is processed and the keywords "Airlines", "OS", "OS Mac", "OS Buy", and "Woods" are appended to "Tiger" and showed to the user.

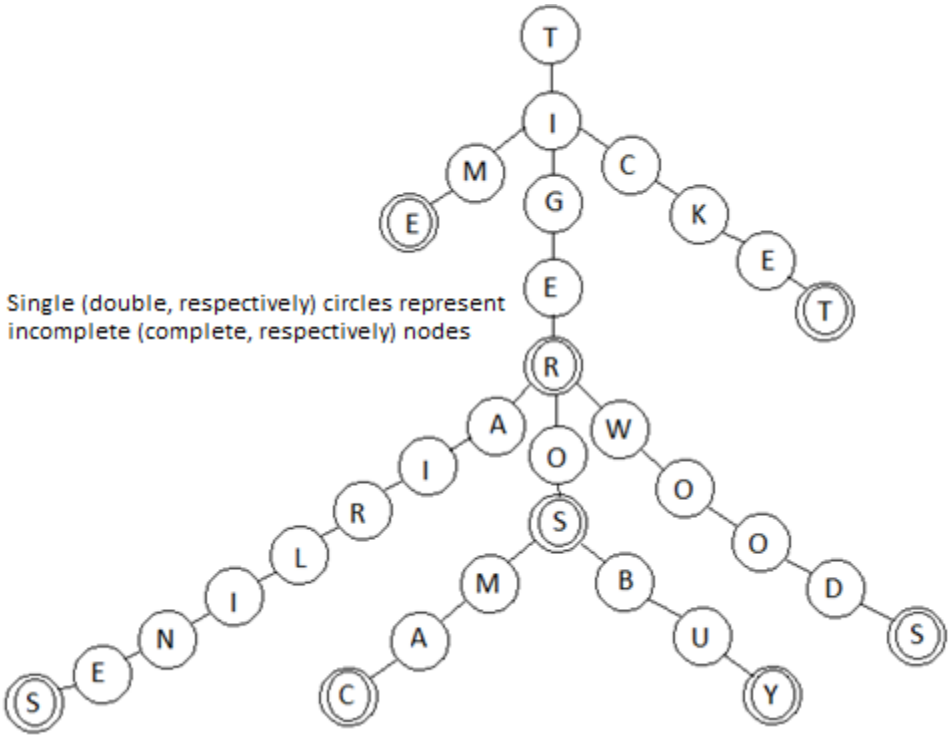


Figure 3. A sample trie

3.2.3 Ranking possible suggestions

Since the number of potential suggested keywords can be large, *Q-WISE* ranks each suggested query keyword based on (i) the *frequency of occurrence* of the keywords in the query logs, (ii) their *similarity* with the user's submitted keywords based on the word-correlation factors[4] [Koberstein 06], and (iii) the *number of times* the initial user query is *modified*, within 10 minutes, to the suggested query. The word-correlation factors, which are computed using the co-occurrence of keywords in a huge set of Wikipedia documents, determine the suggested keywords which are similar to the keywords in a user's query.

Given that SQ is a suggested query, *Q-WISE* computes a *ranking* score for SQ, denoted *SuggRank*(*SQ*), which reflects the degree of *closeness* of *SQ* (in terms of the information

[4] This measure cannot be employed until at least one keyword is entered by the user.

content) to the letters/keywords that have been entered while a user query is being constructed. *SuggRank* is defined as

$$SuggRank(SQ) = \frac{freq(SQ) + WCF(SQ,Q) + Mod(SQ,Q)}{1 - Min(\ freq(SQ), WCF(SQ,Q), Mod(SQ,Q))} \quad (1)$$

which is the Stanford Certainty Factor [Luger 05] on *freq(SQ)*, *WCF(SQ, Q)*, and *Mod(SQ, Q)*, where *Q* are the letter(s)/keyword(s) the user has entered when *SuggRank* is being computed, *freq(SQ)* is the *frequency* of *SQ* in the query logs, *WCF(SQ, Q)* is computed as the *sum* of the word-correlation factors [Koberstein 06] between keywords in *SQ* and *Q*, and *Mod(SQ, Q)* is the *number of times Q* is *modified* to *SQ* in the same session in the query logs within 10-minutes. The Stanford Certainty Factor of a value *V* is a measure that integrates different assessments, i.e., scores or weights to yield an estimation of the *strength* of *V*. The *Min* function is used when *all* elements in the formula are required (AND), whereas the *Max* function is used when some of the elements are optional (OR). Since $freq(SQ), WCF(SQ, QL), and\ Mod(SQ, Q)$ are in different numerical scales, prior to computing the *SuggRank* of *SQ*, *Q-WISE* normalizes them using a logarithmic scale so that they are in the same range.

Q-WISE selects the top 10 suggestions, which follows the 10 results per page approach employed by most major search engines, such as Google and Yahoo. Figure 4 shows a snapshot of *Q-WISE*'s query suggestion on the query keyword "Tiger" in comparison with Google. Notice that in the figure Google's suggestions are mostly focused on "Tiger Woods", the golf player, and there is no mention of any animal or airlines named 'Tiger'. This is likely because Google considers the number of times keywords in a query are submitted to the search engine as a feature in ranking, and "Tiger Woods" is a major topic in news. While the suggested query keywords are good for users using Google to look for Tiger Woods, it is not so for its users who are interested in an animal, an airline, or other information associated with the keyword 'tiger'.

Figure 4. Query suggestions generated by *Q-WISE* and Google based on the keyword "Tiger", respectively

3.3 Clustering

According to [Selberg 99], the *web search problem* is defined as finding the set of documents on the Web relevant to a given user query. However, these days the problem is more complicated than simply finding relevant documents due to (i) information overload caused by the increasing information available on the Web, which has become a highly dynamic collection of documents and (ii) the large number of potential documents relevant to a user query. To solve the problem, existing web search engines, such as Google, Yahoo!, and Bing, adopt a ranking approach. In response to a user query, a search engine retrieves a ranked list of documents; the higher a retrieved document is on the list, the higher its relevance to the query is. Many algorithms for computing the degree of relevance of retrieved documents have been proposed [Brashler 09, Liu 08, Shekhar 10]; however, these algorithms work well only with queries that are *precise* and *narrow* [Li 09, Osinski 06]. If the query is too *general* or *broad*, it is difficult, if not impossible, for a search engine to identify precisely the specific set of documents that satisfy the user's information needs. The side effect is that the user is required to sort through a list of documents to locate the relevant ones that (s)he is particularly interested in. Such a search has been

identified as a *low precision search* [Zamir 99]. The low precision search problem is caused by the fact that web search queries are typically *short* and often *ambiguous* [Shen 06]. As reported in [Weiss 01], which is also presented in a survey on iProspect (http://www.iprospect.com /premiumPDFs/keyword_length_study.pdf), as well as in our analysis of the queries in the AOL/HBLL query logs, majority of the queries submitted to web search engines are general 1-3 words in length, which make up about 89% of the total queries in AOL and HBLL query logs.

Consider as an example the results (included in the first page) retrieved by Google (on October 11, 2010) for the ambiguous query "jaguar", which include *jaguar the car*, the *animal*, the *sports team*, and the *software* (see Figure 5a). Even if "Jaguar" is revised into a more specific query such as "jaguar team picture", its results are still diverse in content, which include documents that address the *Jacksonville football team*, a *boat race team*, a *car team*, and a *music team* (see Figure 5b). Moreover, if a user is interested in downloading the jaguar software, a query such as "download jaguar" yields documents that discuss downloading jaguar *brochure*, jaguar *screensaver*, jaguar *wallpaper*, jaguar *music*, and jaguar *software* (see Figure 5c). Low precision searches are inevitable, and methods to enhance the searches are needed [Li 09].

Many methods have been introduced in solving the low precision search problem, which include filtering [Bernard 08], relevance feedback [Jung 07], and a human-made directory [Yahoo!]. *Filtering* involves minimizing the number of retrieved documents by methods varying from applying simple pruning techniques to advanced Artificial Intelligence algorithms. Although filtering techniques limit the total number of retrieved results, they still cannot solve the low precision search problem. *Relevance feedback* refines a user's query by adding chosen keywords in retrieved documents labeled as relevant by the user. The feedback strategy, however, requires the user's involvement throughout the retrieval process and thus is not fully

automated. A human-made directory, which is implemented by Yahoo!, includes among ranked documents retrieved by the search engine other documents from a human-made directory that are relevant documents to the particular query labeled in advance and is constructed by a number of human experts. Since the Web is highly dynamic, many of the documents in the directory can easily become out-dated or may not be available anymore which could have been removed by their owners.

Clustering, on the other hand, is a promising approach which identifies labels, groups, and assigns similar search results to different categories according to their subject areas. Generated clusters simplify the user's search process by allowing the user to quickly locate the specific subset of retrieved documents that satisfy the user's specific information need, which is a solution to the low precision search problem. Providing clustered search results would be much more appealing and useful to a user than a ranked list of retrieved documents that intermix with results based on different interpretations of a given query.

Figure 5. The first page of results generated by Google for the queries, "jaguar", "jaguar team picture", and "download jaguar", respectively

As the first step in the clustering process, *Q-WISE* submits the user's query to three major web search engines, Google, Yahoo!, and Bing, and retrieves the title, snippet (which is a short summary of the corresponding document), and URL of each one of the top 33 documents returned by each of the three search engines. During the clustering process, *Q-WISE* (i) generates a set of (cluster) labels, which are non-stopwords in the titles and snippets of retrieved

documents, (ii) selects (ranks, respectively) a subset of the labels generated in Step (i) using a set of features introduced in Section 3.3.1 (Section 3.3.2, respectively), (iii) applies the Vector Space Model (VSM[5]) to calculate the similarity between each label and each retrieved document (as presented in Section 3.3.3), and (iv) assigns the top-N (≥ 1) retrieved documents, ranked by VSM using the cosine similarity measure between documents and the labels, to their corresponding cluster identified by a label, where N is computed as the *weight* of the label divided by the sum of the weights of all the labels chosen for the clusters (as defined in Section 3.3.2).

3.3.1 Choosing cluster labels

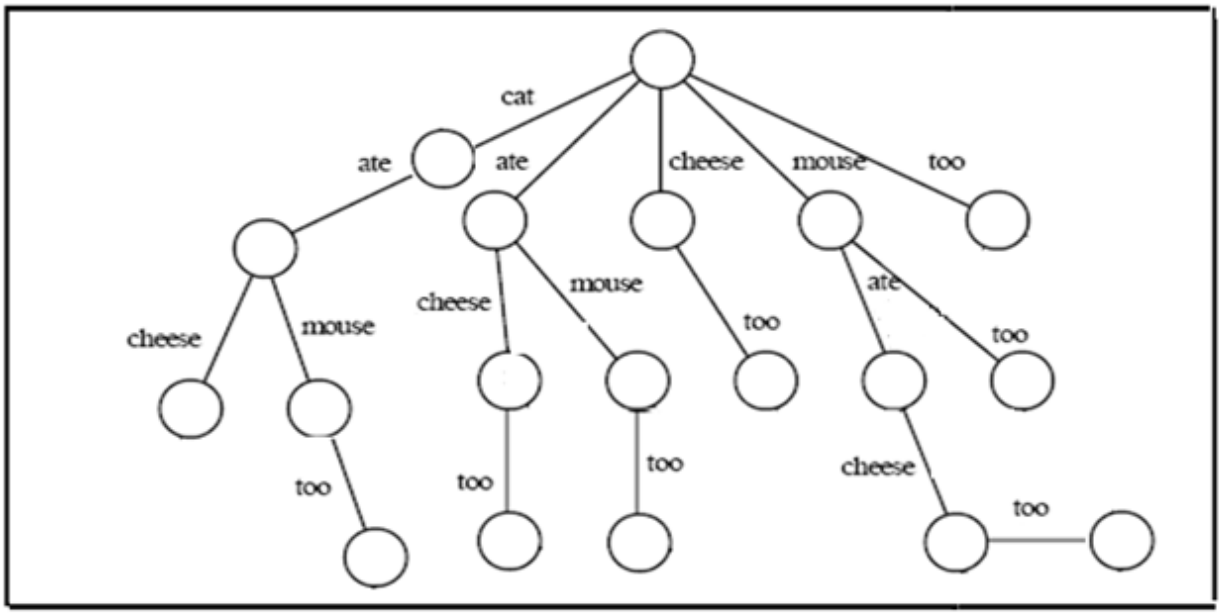
In a typical text clustering algorithm, cluster labeling follows the document clustering step and the entire document text is used for clustering. Web-based clustering, on the other hand, is different as it involves different challenges. Web-based clustering must be (i) *fast* and document titles and snippets can be processed faster than using the entire documents in extracting and ranking cluster labels, (ii) *flexible* and *fully automated* as web contents change constantly and user feedback must be avoided, and (iii) *user-oriented*, i.e., eases the process of finding the required information by producing cluster labels that are meaningful and informative to the user, which is not a concern in document clustering due to the different type of end users. The users of traditional clustering tools are typically skilled professionals, whereas users of web-based clustering are often not experts, and thus are less tolerant to errors. *Q-WISE* first creates cluster labels prior to performing the clustering step using the *titles* and *snippets* of documents retrieved by Google, Yahoo!, and Bing.

Candidate cluster labels are created by *Q-WISE* and extracted from the retrieved document titles and snippets by utilizing the *suffix tree* data structure. Labels are *ranked*

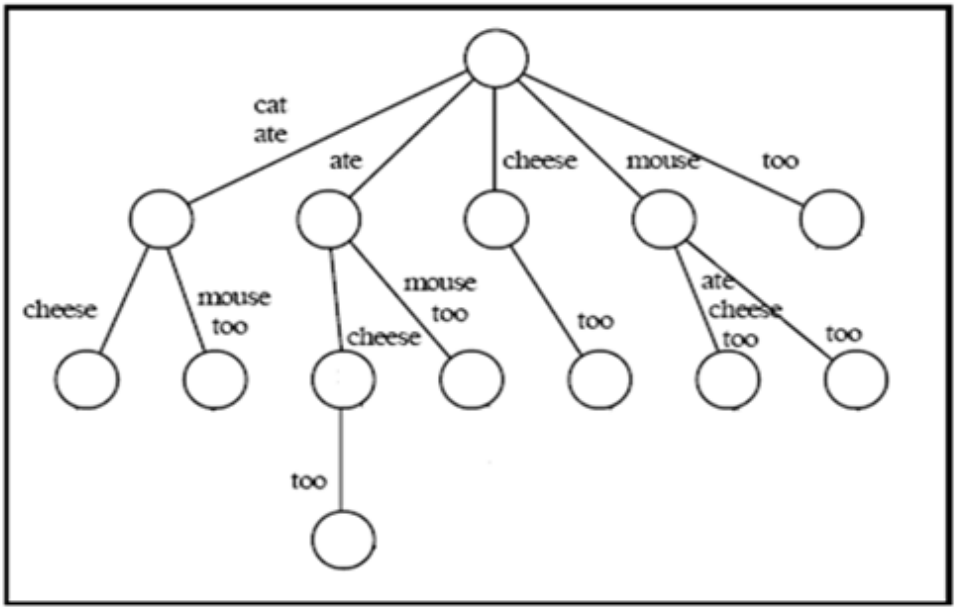
[5] VSM calculates the similarity between a query and each retrieved document using the well-known *term frequency* (TF) and *inverse document frequency* (IDF) measures.

according to a set of features which determines the score of each label that reflects its accuracy in capturing a particular subject area addressed in a subset of retrieved documents. Retrieved documents in each subset are assigned to their corresponding candidate label according to the score of the label (as presented in Section 3.3.2).

A *suffix tree* is a data structure that retains all the n-grams in a list of words that yield a string, which in our case is a sentence in a document snippet or title. Strings can be inserted into the tree incrementally in time *linear* to the number of words in each string, which has a runtime of *O(n)*, where *n* is the total number of words in the combined document snippets and titles in our case. Given a string *S* such that *S* is the concatenation of all the sentences in the snippets and titles of retrieved documents, a suffix tree of *S* is a rooted, directed tree in which the root node has a child node for each distinct word in *S* (see Figure 6a for an example), and edges between node *A* and another node *B* are *combined* if each node on the path *P* from A to B (including A) has only a single child and none of the labeled edges (excluding the edge leading to *B*) on *P* is the last word of any sentence in *S* (see Figure 6b). Each edge is labeled with a non-empty substring of *S* such that subsequent edges represent subsequent keywords in *S*, i.e., a suffix tree is a compact trie containing all the suffixes of *S*. The label (value) of a node *N*, which is used when traversing the suffix tree for processing, is the concatenation of the edge labels on the path from the root to *N*. Figure 6 shows a sample suffix tree of the strings “cat ate cheese”, “mouse ate cheese too”, and “cat ate mouse too” as shown in [Zamir 98].



(a) The initial tree



(b) The Final Tree

Figure 6. The initial and final suffix tree of the strings "cat ate cheese", "mouse ate cheese too", and "cat ate mouse too"

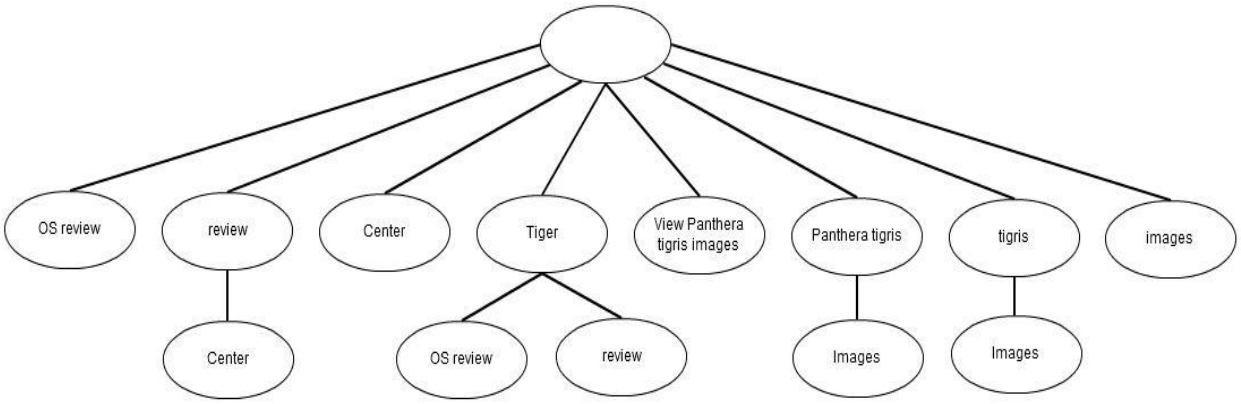
A suffix tree is constructed each time a query is processed by *Q-WISE*, which can be used for identifying *candidate* cluster labels *fast*, such that each node on a suffix tree is a candidate cluster given that the node (cluster) has more than one document assigned to it as suggested by

[Gulla 07]. Figure 7 below shows an example of how candidate cluster labels are extracted from a suffix tree.

Query: Tiger

Document 1: Tiger OS reviews. Review center
Document 2: Panthera tigris
Document 3: Tiger reviews
Document 4: View Panthera tigris images

Suffix Tree:



Candidate Clusters: Review, Tiger, Panthera tigris, tigris.

Figure 7. Candidate cluster labels generated for the query "Tiger"

Since the number of extracted suffixes can be large and not all the suffixes capture the subject area shared by a subset of retrieved documents, selected (cluster) labels must *satisfy* the following constraints and are non-numerical keywords:

1- Labels do not cross sentence boundaries, since sentence markers indicate a topical shift.

2- Labels are *complete* which are not included as substrings in other labels.

3- Labels do not begin or end with a stopword, since stripping leading and trailing stopwords from a phrase is likely to increase its readability [Zamir 99].

4- Labels do not end in the Saxon genitive form, which is a traditional term for the apostrophe-s.

3.3.2 Ranking cluster labels

Selected cluster labels (as created in Section 3.3.1) might (i) differ in the number of documents assigned to the labels for which their contents are captured by the labels and (ii) still be *ineffective* compared with others in capturing a subject area of a subset of retrieved documents. Hence, *Q-WISE* ranks the selected cluster labels according to various features, such as *frequency*, *stability*, and *significance* of the selected labels, to ensure that chosen labels which capture the contents of a larger portion of retrieved documents compared to other chosen labels and are more informative than other chosen labels are ranked higher on the list and are assigned a larger number of documents subsequently.

Let *L* be a cluster label and *C* be the set of retrieved documents. Cluster labels are *ranked* using the following measures:

1- *The number of Title Words* (*NTW*) in *L*, which is the number of keywords in the titles of documents in *C* that are also in *L*. The more title keywords in *L*, the higher its ranking score of *L* is, since the title of a document *D* usually reflects the *subject area* of *D*. Snippets, on the other hand, are not considered in this measure, since they represent sentences extracted from *D* that usually may not reflect the topic of *D* [Osinski 06].

2- *The frequency of Label* (*FoL*) *L*, which is the *frequency of occurrence* of *L* in the titles and snippets of documents in *C*. The more frequently a label occurs in the titles and snippets of a subset of documents *S* in *C*, the more likely it represents the *content* of *S* [Osinski 06].

3- *The label Stability* of *L*, which measures the mutual information (dependence), denoted *MI*, of *L* [Zhang 01]. *Dependency* identifies cluster labels that characterize the contents of documents in one cluster in contrast to other clusters. The higher the *MI* of *L* is, the more

dependent L is as a cluster label. Assume that *L* = "$c_1 c_2 \ldots c_n$", where c_i ($1 \leq i \leq n$) is a (stop)word in *L*, the *stability* of *L* is defined in [Zhang 01] as

$$MI(L) = \frac{f(L)}{f(L_L) + f(L_R) - f(L)}, \quad (2)$$

where $L_L =$ "$c_1 \ldots c_{n-1}$", $L_R =$ "$c_2 \ldots c_n$", and $f(L_L), f(L_R),$ and $f(L)$ are the frequencies of occurrence of $L_L, L_R,$ and L, respectively in the titles and snippets of documents in *C*.

4- *The label Significance* (*LS*) of *L*, which indicates the significant factor of a cluster label [Zhang 01]. The *longer* a cluster label is, the *higher* its *significance* is, since longer cluster lables are more *informative* of the subject area covered in the corresponding subset of documents. The *significance* of *L* is defined in [Zhang 01] as

$$LS(L) = f(L) \text{ x } g(|L|) \quad (3)$$

where *f*(*L*) is the *frequency* of occurrence of *L* in the titles and snippets of documents in *C*, |*L*| is the number of (stop)words of *L*, *g*(*x*) is a function such that *g*(1) = 0, *g*(*x*) = log_2 *x*, when $2 \leq x \leq 8$, and g(*x*) = 3, when *x* > 8.

Q-WISE computes a *ranking* score for *L*, denoted *LabWeight*(*L*), which reflects the *significance* of *L* in capturing the *contents* of documents in *C*. *LabWeight* is defined as

$$LabWeight(L) = \frac{NTW(L) + FoL(L) + MI(L) + LS(L)}{1 - Min(NTW(L), FoL(L), MI(L), LS(L))} \quad (4)$$

which is the Stanford Certainty Factor [Luger 05] on *NTW*(*L*), *FoL*(*L*), *MI*(*L*), and *LS*(*L*). Since $NTW(L), FoL(L), MI(L), and\ LS(L)$ are in different numerical scales, prior to computing the *LabWeight* of *L*, *Q-WISE* normalizes them using a logarithmic scale so that they are in the same range. Figure 8 shows a set of ranked labels created from the retrieved documents in response to the query "Eagle".

3.3.3 Document clustering

The Vector Space Model (VSM), which compares textual data by using algebraic vectors in a multidimensional space, is adopted by *Q-WISE* for assigning retrieved documents to their clusters identified by the corresponding labels. The linear algebra operations calculate the similarities among a given set of documents using VSM, which are effective and efficient. *Q-WISE* considers the retrieved document title, document snippet, and the cluster labels, such that each cluster label is treated as a separate document during the comparison process, and the degree of similarity between each label and the retrieved set of documents is computed. Every unique keyword in the retrieved documents to be analyzed yields a dimension in the VSM, and each document is represented as a vector spanning all these dimensions. For example, if vector v represents document j in a k-dimensional space, then component t of vector v, where $t \epsilon 1 \ldots k$, denotes the degree of similarity between document j and t in (the document representing) the label. The degree of similarity between retrieved documents and labels can be captured in a $k \times d$ matrix, known as the *term-document matrix*, where k is the number of unique terms in all the documents and d is the number of documents in the collection of 99 retrieved documents plus the chosen cluster labels. Element a_{ij} of the term-document matrix is the similarity value between term i of the label and document j of the set of retrieved documents. There are many methods for calculating a_{ij}, and the most common one, which is also adopted by *Q-WISE*, is the *cosine similarity measure* between any two documents using the *vector dot product* formula.

After selecting and ranking cluster labels, *Q-WISE* assigns a number of documents N to each cluster label, which yields a list of ranked documents assigned to each cluster. *Q-WISE* assigns the top-N (≥ 1) documents ranked by VSM to each cluster label L, where N is computed

as the *weight*, i.e., *LabWeight*, of *L* divided by the sum of the weights, i.e., *LabWeights*, of all the chosen labels rounded to the nearest whole number, as shown in equation 5.

$$N = ROUND\left(\frac{weight(L)}{\sum_{1}^{\#L} weight(L)}\right) \quad (5)$$

This method proves to be effective, since it (i) allows *overlapping* of documents between clusters, which is sometimes required as some documents may include contents of multiple subject areas of documents in different clusters, (ii) runs *fast* which takes less than 2 seconds on 99 documents and yields a number of labels between 10 to 20, and (iii) assigns *highly ranked* cluster labels more documents, since the distribution of different subject areas presented in the retrieved set of documents is not uniform, and highly ranked labels capture the subject areas of more documents than others.

3.3.4 User interface

Q-WISE's interface is very simple, following the design of the user interface of popular web search engines being used these days. The interface consists of a query text box and a search button. After a search is processed by *Q-WISE*, a list of ranked clusters for the query is displayed. Clicking on a cluster label *L* shows the titles and snippets of the documents contained in the cluster of *L*, along with the summary of the documents in the cluster (discussed in Section 3.4), if desired. Figure 8 shows a sample list of ranked clusters for the query "Eagle" and Figure 13 shows the generated summary for the label "Eagle Scout".

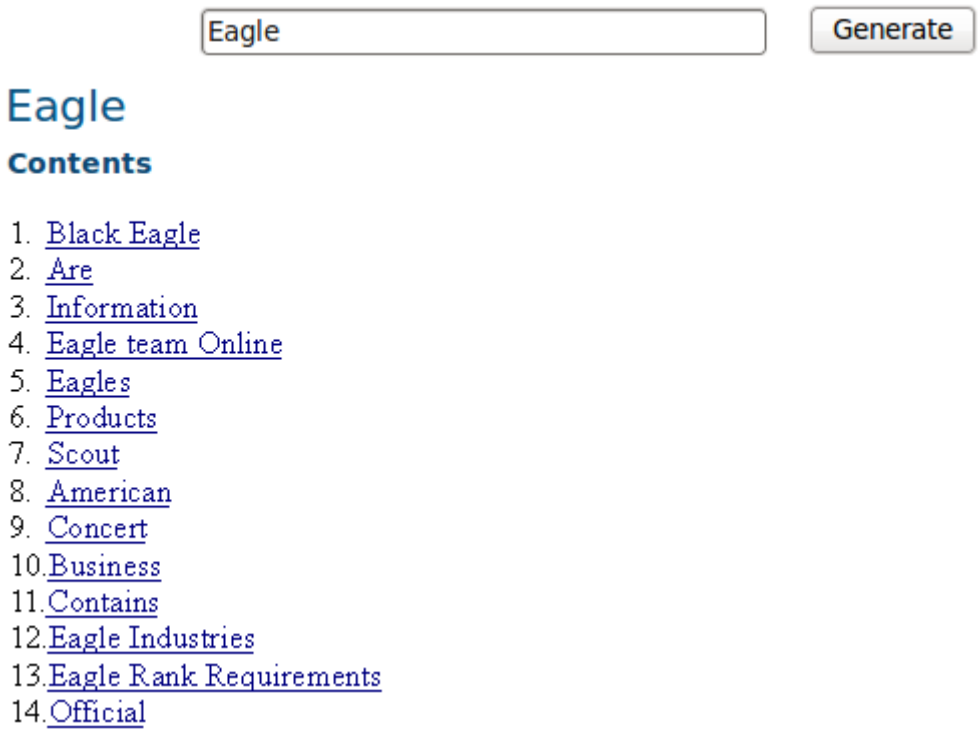


Figure 8. Query "Eagle" run through *Q-WISE*

3.4 Summarization

According to the report prepared by ComScore (http://searchenginewatch.com/), 213 million web queries were submitted on a daily basis in March 2006 in the U.S.A alone, which yielded 148,000 searches per minute. Existing web search engines, as stated earlier, retrieve a ranked list of topically-related documents in response to a keyword query. Many of these documents, however, include overlapped information and some of them may not meet the user's information need. To minimize the user's time and efforts in scanning through the contents of retrieved documents to determine the useful ones, existing web search engines display each retrieved document *D* with a *snippet*, which serves as a short summary of *D*. These snippets, however, are not always useful to the user, since they (i) often include very similar information and (ii) are created using phrases in retrieved documents solely based on the number of query keywords appeared in them, which may not capture the main content of the corresponding documents.

Consider the top-5 results retrieved by Google (on October 19, 2010) for the query “First walk on the moon", as shown in Figure 9. The titles and snippets of these results show the same information, i.e., Neil Armstrong was the first man to walk on the moon and the date of that incident. If the user is interested in more/other (specific) information about the *first walk on the moon*, such as the shuttle used in space, astronauts that accompanied Neil Armstrong, length of the journey, etc., the user must scan through the retrieved documents one by one, since there is no clue (indication) as to which retrieved documents might include the additional information. Even though the snippets of the top-5 results retrieved by Google for the query “Health problems related to computers”, as shown in Figure 10, convey different information, none of the snippets provide information addressing the “problems” specified in the query, since the retrieved snippets merely contain query keywords, i.e., “problems”, “health”, “computers”, and “related”, without including intuitive, useful information discussing the “real problems” in a substantial way. Moreover, the snippets do not reflect which documents contain more specific information about the “problems”, i.e., if the user wants to see the “problems” but is also interested in possible solutions, he/she has no clue as to which of these documents contain that information.

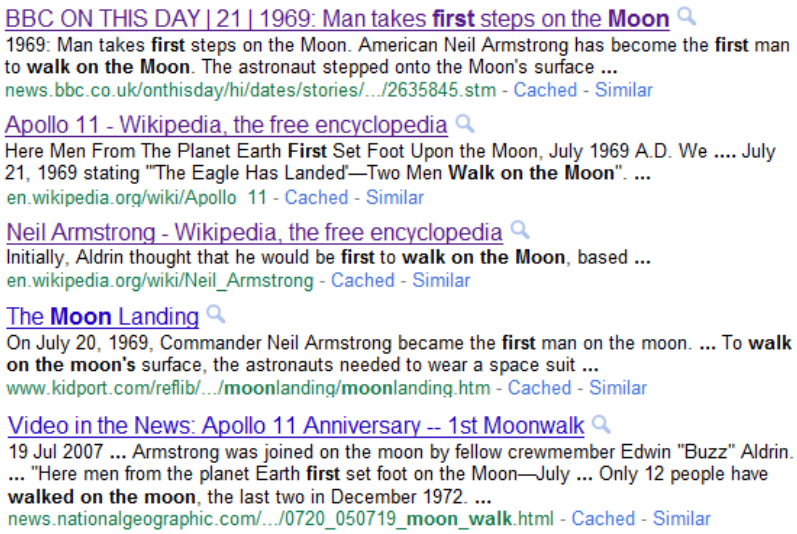


Figure 9. The top-5 results retrieved by Google (on October 19, 2010) for the query “First walk on the moon”

Figure 10. The top-5 results generated by Google (on October 19, 2010) for the query "Health problems related to computers"

Summarization is a promising approach in dealing with the issue of ineffective snippets and information overload, since it provides a summary (abstract) that includes the key concepts covered in a (set of) document(s). An ideal text summary of a (given set of) document(s) *S* (i) includes unique but excludes extraneous, redundant information presented in (various documents in) *S*, (ii) must be coherent and comprehensible, which can be achieved using natural language processing to handle issues such as *co-reference* and the *temporal dimension* (to be introduced in Sections 3.4.2 and 3.4.5, respectively), and (iii) must be of appropriate length, since a *very brief* summary is likely to exclude some important information in *S* and a *very detailed* one is likely to repeat the same information in *S*. Figures 11 and 12 show the summaries generated by *Q-WISE* for the top-5 results retrieved by Google for the queries "First walk on the moon" and "Health problems related to computers", respectively. *Q-WISE* includes five sentences in each summary, which are compatible in size with the top-5 snippets retrieved by Google with one sentence for

each retrieved document (i.e., 5 sentences in total). The summary for “First walk on the moon” includes the information provided in the snippets, in addition to other relevant information, such as the space shuttle, the time of arrival, the astronauts that accompanied Neil Armstrong, etc., excluded in the snippets of the top-5 results for the query. The summary for “Health problems related to computers” contains information relevant to the query, such as symptoms, corrective measures, and the actual problems, which are missing in the corresponding set of snippets.

Neil Armstrong was the first to step onto the Moon's surface, in the Sea of Tranquility, at 0256 GMT, nearly 20 minutes after first opening the hatch on the Eagle landing craft.

Neil Armstrong and Edwin Aldrin spent a total of 21 hours on the Moon, two-and-a-half of them outside the landing module.

The astronauts also unveiled a plaque bearing President Nixon's signature and an inscription reading: "Here men from the planet Earth first set foot upon the Moon July 1969 AD.

As he put his left foot down first Armstrong declared: "That's one small step for man, one giant leap for mankind".

Neil Armstrong was joined by colleague Edwin "Buzz" Aldrin at 0315 GMT and the two collected data and performed various exercises - including jumping across the landscape - before planting the Stars and Stripes flag at 0341 GMT.

Figure 11. The summary generated by *Q-WISE* for the query “First walk on the moon”

Working with computers can lead to various health-related problems, such as eye trouble, headache, pain in the neck and shoulders, arms, or hands.

Wear splints while you work to keep your wrists from bending too high or low, and use a keyboard tray so the keyboard and mouse are below your elbows and your wrists are level.

Corrective Measures: The correct thing to do is to place the computer table between lights and not directly under them.

Symptoms include pain, fatigue, irritation of muscles and tendons, a tingling or numb feeling, or loss of strength.

Causes: Wrong type of chair or desk, Right chair an desk but wrong posture, Sitting on the edge of the chair, sitting with all the weight on one buttock by sitting cross legged.

Figure 12. The summary generated by *Q-WISE* for the query “Health problems related to computers”

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 "the 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. The following six design issues have been addressed and emphasized in the design of a (query-based) multi-document summarization approach as compared to the design of a single-document summarization method [Ou 07].

1. The *redundant information* in a set of topically-related articles is much higher than its counterpart in an article, since an article usually discusses the main idea and other related information. A multi-document summarization approach is expected to eliminate sentences that convey the same piece of information.
2. Information presented in a set of articles may invoke a *temporal dimension*, which is typical in a stream of news reports on an unfolding event, and the most-recent developed information often override earlier ones. A multi-document summarization approach orders sentences in a given set of documents partially based on their publication dates.
3. The *length* of a summary is typically smaller for a collection of dozens/hundreds of topically-related documents than for concatenated single-document summaries, one for each of the documents in the same collection. The Document Understanding Conference (DUC) suggests a summary length of 250 words for a collection of documents, which is adopted by *Q-WISE*. Given such a small compression, users should be given the option to trace the original documents using their corresponding sentences in the summary for detailed information.

4. The *co-reference problem* (presented in Section 4.3.2) presents an even greater challenge for multi-document summarization than for single-document summarization for two main reasons. First, as stated earlier, a summarization approach must identify whether two references in two different sentences address the same object, which is a more complicated issue when the sentences come from different documents, since an object might be identified in various ways in different documents. Second, a multi-document summary may contain sentences extracted from several documents, which may include a pronoun without its preceding referent. If the referent is not identified, it is likely to (i) leave the reader with insufficient information to understand the summary, and/or (ii) give the reader a false indication of what the referent is.
5. Achieving good *coverage* in multi-document summarization is difficult, since topically-related articles from where information are selected for creating a summary can include a 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, since the former (i) provides an overview of various subtopics, if they exist, of a particular subject, (ii) gives the user more information about the subject while eliminating common information across many documents, and (iii) identifies a subject or research topic that evolves over time.

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 *SD* to be included in the summary so that they are readable and grammatically correct. *Q-WISE* adopts the extractive summarization strategy at the sentence level.

Q-WISE creates a summary for each cluster *C* of retrieved documents by (i) downloading and preprocessing the 99 documents retrieved from Google, Yahoo!, and Bing (discussed in Section 3.4.1), (ii) identifying and associating all (pro)nouns in the retrieved documents with their referents (discussed in Section 3.4.2), (iii) 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 the set of features defined in Section 3.4.3, (iv) choosing the top-*M* ($\geq$ 1) sentences (based on their *RS* scores) from the documents in *C*, such that $\sum_{i=1}^{M-1} L_i < 9 \text{ x } 250$ and $\sum_{i=1}^{M} L_i \geq 9 \text{ x } 250$, where L_i is the length in words of sentence *i* in *C*, (v) clustering the *M* sentences using the HAC algorithm based on the word-correlation factors [Kobersten 06] (as discussed in Section 3.4.4), (vi) selecting the top-*N* sentences (based on their *RS* scores) from each sentence cluster created in Step (v), such that $\sum_{i=1}^{N-1} L_i < 250$ and $\sum_{i=1}^{N} L_i \geq 250$, and, if desired, (vii) re-weighting the selected sentences based on their temporal dimensions to capture the flow of events (as discussed in Section 3.4.5). If the number of sentences *N* to be selected for a summary is *less* than the number of created sentence clusters of *C*, then the *N* sentences (one from each of the *N* clusters) with the highest *RS* score are chosen.

We use 9 x 250 in our equation, since Schlesinger et al. [Schlesinger 08] claim that 9 x *W* words are required to generate a summary *Sum*, where *W* is the number of words in *Sum*, since 9

x *W* provides sufficient, distinct contents for summarization. As discussed earlier, a *Q-WISE*-generated summary is about 250 words.

3.4.1 Document preprocessing

The set of 99 documents retrieved from Google, Yahoo!, and Bing, are first downloaded and preprocessed, where each retrieved document is in HTML format. We consider HTML pages only for creating multi-document summaries, since (i) other formats are very complex to process and require their own parsers and additional overhead time and (ii) over 99% of the documents retrieved by Google, Yahoo!, and Bing are HTML documents. Each document *D* is then parsed to remove surplus data, which include links to other documents, advertisements, and non-textual data, such as images and videos, and retain 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, such as periods, question marks, and exclamation points. The exclamation point and question mark are less ambiguous as end-of-sentence indicators. However, since 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.”, along with regular expressions for detecting decimals, email addresses, and ellipse, are used to ensure reliable identification of sentence boundaries. Hereafter, each sentence is parsed into a sequence of word tokens using the Conexor Parser (http://www.connexor.eu/technology/machinese/demo/). For each word token, its *Document_ID*, *Sentence_ID*, *word form* (the real form used in the text), *stem* (generated using the Porter stemming algorithm [Croft 10]), and *creation date* of the corresponding document are stored. The *Document* and *Sentence ID*s identify the document from where the sentences are extracted, the *stem* of a word is used in

different *sentence* and *document similarity* formulas, and the *date* is used for re-weighting the sentences in a summary based on their *temporal dimension*.

3.4.2 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. For example, given a sentence *S* in [Watson 10] "Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Lionel Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment", a co-reference resolution system partitions *S* such that all (pro)noun phrases in sentences referring to the same real-world entity are grouped together. Hence, in processing *S*, a co-reference resolution system yields three groups <Queen Elizabeth, her>, <husband, King George VI, the King, his>, and <Lionel Logue, therapist>. In summarization, it is required to replace a (pro)noun in a sentence with its referencing entity, since sentences in the summary can lose their original orders and yield a false indication of what the (pro)noun refers to. *Q-WISE* adopts an open source package for performing *co-reference resolution* [Watson 10] in solving the co-reference problem.

3.4.3 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* of each sentence, *Q-WISE* utilizes the following *features*:

1- *Title Frequency* (*TiF*) is the number of words in *S* that appear in the *cluster label* of *C*.

2- As the 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* [Luhn 58], denoted *SF*, of *S* based on significant words as

$$SF(S) = \frac{|significant - words|^2}{|S|} \quad (6)$$

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

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

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 low- and high-frequency cutoff values, respectively.

3- 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-WISE* computes *Sim*(S_i) using (i) the *word-correlation factors* (*wcf*) [Koberstein 06] of every word in S_i and words in each remaining sentence S_j in *C* and (ii) the *Odds ratio* = $\frac{p}{1-p}$ [Judea 88].

$$Sim(S_i) = \frac{\sum_{j=1, i \neq j}^{|C|} \sum_{k=1}^{n} \sum_{l=1}^{m} wcf(w_k, w_l)}{1 - \sum_{j=1, i \neq j}^{|C|} \sum_{k=1}^{n} \sum_{l=1}^{m} wcf(w_k, w_l)} \quad (8)$$

where |*C*| is the number of sentences in *C*, *n* (*m*, respectively) is the number of words in S_i (S_j, respectively), and w_k (w_l, respectively) is a word in S_i (S_j, respectively).

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

$$LSS(S) = sim(L, S) = \frac{\sum_{i=1}^{N} w_{i,S} \text{ x } w_{i,L}}{\sqrt{\sum_{i=1}^{N} w_{i,S}^2} \text{ x } \sqrt{\sum_{i=1}^{N} w_{i,L}^2}} \quad (9)$$

where $w_{i,S} = tf(i, S)$ x $idf(i)$, $w_{i,L} = tf(i, L)$ x $idf(i)$, $w_{i,S}$ ($w_{i,L}$, respectively) is the weight of a word *i* in *S* (*L*, respectively), and *N* is the total number of distinct keywords in

C. The *higher* the *LSS* value of *S*, the *higher* is the degree of *S* in reflecting the content of *C*, since *L* captures the contents of documents in *C*.

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

$$NE(S) = \frac{\sum_{i=1}^{|E|} f(E_i)}{f(E)} \tag{10}$$

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 contain any [Osinski 06].

6- A penalty is given to each short sentence (with less than 15 words) or long sentence (with more than 30 words) [Schiffman 02], since *short* sentences often require some introduction or 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-WISE* assigns a *Sentence Length* (*SL*) value of -1 to *S* if |*S*| < 15 or |*S*| > 30, where |*S*| is the number of (stop)words in *S*, and 0 otherwise.

7- 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 58]. Hence, *Q-WISE* assigns a *Sentence Position* (*SP*) value of 1 to *S*, if *S* is the *first* sentence of the *first* paragraph or the *last* sentence of the *last* paragraph in any document in *C*.

Q-WISE computes a *ranking score* for *S*, denoted *RS(S)*, which reflects the *relative degree of significance* of *S* in capturing the contents of the documents in *C* and is defined as

$$RS(S) = \frac{TiF(S) + SF(S) + Sim(S) + LSS(S) + NE(S) + SL(S) + SP(S)}{1 - Min(TiF(S), SF(S), Sim(S), LSS(S), NE(S), SL(S), SP(S))} \quad (11)$$

Equation 11 computes *RS*(*S*) using the Stanford Certainty Factor [Luger 05]. Since $TiF(S), SF(S), Sim(S), LSS(S), NE(S), SL(S)$, and $SP(S)$ are in different numerical scales, prior to computing *RS*, we normalize them using a logarithmic scale so that they are in the same range.

3.4.4 Solving the anti-redundancy and coverage problems

Before selecting sentences for creating the summary of a document cluster *C*, *Q-WISE* clusters the top-*M* ($\geq$ 1) ranked sentences (based on their *RS* scores) in *C*, where *M* is *nine* times the length of the desired summary, as discussed earlier, using the Hierarchical Agglomerative Clustering (HAC) algorithm. The HAC algorithm initially assigns each sentence to a singleton cluster, and then repeatedly merges clusters until a specified termination criterion is satisfied [Manning 99]. Since the HAC algorithm relies on a *similarity metric* among sentences in any two clusters for merging clusters, *Q-WISE* uses the *sentence-similarity measure* (*Sim*), as defined in Equation 8, to compute the similarity among the sentences in two (intermediate) clusters. To determine the termination criterion for HAC, *Q-WISE* applies the algorithm in [Alguliev 08], which implements Neural Nets [Grossberg 88], to determine the *optimal* number of subtopics covered in a (set of) document(s), which dictates the ideal number of sentence clusters in *C* to be generated by HAC.

Given that *L* is the desired length of a summary *Sum*, *Q-WISE* selects sentences from each sentence cluster *CT* created by HAC to be included in *Sum* such that (i) the first sentence *S* to be chosen from *CT* has the highest *RS* value in *C* and (ii) the selection stops when the length of selected sentences exceeds *L* and the length of the summary up till the second-to-last sentence is less than *L*. After the first round of selection, *Q-WISE* chooses the next sentence S' from *CT* with the *lowest similarity score*, denoted *LSS*, relative to *S*, which is computed as the *sum* of the

word-correlation factors [Koberstein 06] between each word in S' and *S*. By performing this clustering process, *Q-WISE* ensures that selected sentences are *distinct* in contents, which avoids *redundancy* and maximizes *coverage* in terms of the information included in the summary.

3.4.5 Adding the temporal dimension

The information captured in a set of documents on a particular topic might be dynamically changing 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-WISE* accounts for the *temporal dimension* in a set of documents by re-weighting each sentence 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*), as defined below.

$$RS(S) = TD(S) * RS(S) \quad (12)$$

where *S* is a sentence in a document cluster *C*, *TD*(*S*) is a time-based weight of *S*. The *earlier* a document in *C* which include *S* is published, the *smaller* the *TD*(*S*) is. Since *exponential average* is extensively used in time-series prediction, *Q-WISE* uses the *decay rate formula* in computing *TD*(*S*), which decreases the sentence weight exponentially according to time [Yu 05].

$$TD(S) = DecayRate^{\frac{y-t}{24}} \quad (13)$$

where *y* is the current time (i.e., day, hour, and minute), *t* is the publication time[6] of the document containing *S*, (*y* - *t*) is the time gap in hours, and *DecayRate* is a variable that is experimentally set to 0.5.

We have chosen not to treat the temporal dimension as a *feature* to compute *RS* and include it as a separate *weighting factor* so that it is an option to be included (excluded, respectively) in determining the ranking of *S* in *C*. This option is appropriate, since a given set of

[6] 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.

documents may not discuss events that override one another, i.e., old information are just as important as new ones.

3.4.6 Samples of generated summaries

Figure 13 shows the summary generated for the query "Eagle Scout", which is created for the document cluster labeled "Eagle Scout" as shown in Figure 8. The summary (i) includes *distinct sentences* that present different information such that sentences with *older* dates are ranked towards the bottom, (ii) covers most *subtopics* associated with eagle scouts, which include *badges*, *eagle scout ranks*, *age limit*, *service projects*, and *medals*, (iii) does not include any sentences with *unidentified (pro)nouns*, and (iv) is of *appropriate length*, i.e., 256 words, as defined by DUC.

Some Eagle Scouts have returned their badges to protest the BSA's discriminatory policies.

Eagle Scout may be earned by a Boy Scout or Varsity Scout who has been a Life Scout for at least six months, earns a minimum of 21 merit badges, demonstrates Scout Spirit, and demonstrates leadership in the troop, team, crew or ship.

Adult leaders who earned the rank of Eagle Scout as a youth may wear the square knot on their uniform above the left shirt pocket.

Eldred was the first of three generations of Eagle Scouts; his son and grandson hold the rank as well.

Any Venturer who achieved the First Class rank as a Boy Scout in a troop or Varsity Scout in a team may continue working for the Star, Life, and Eagle Scout ranks and Eagle Palms while registered as a Venturer up to his 18th birthday.

In 1952, age limits were set so that adults over 18 years of age could no longer earn Eagle Scout and the service project requirement was slightly expanded to "do your best to help in your home, school, church or synagogue, and community."

The design of the Eagle Scout medal had not been finalized by the National Council, so the medal was not awarded until Labor Day, September 2, 1912.

The cloth badge was introduced for Eagle Scouts attending the 2nd World Scout Jamboree in Denmark in 1924 with a design based on the hat pin.
Information needed to fill out the Eagle Scout Rank Award Application.

Figure 13. A Summary generated by *Q-WISE* for the cluster (labeled) "Eagle Scout"

Chapter 4

Experimental results

In this section, we assess the overall performance of *Q-WISE*. We first describe the datasets (in Section 4.1) used for the empirical study and detail the statistical approach (in Section 4.2) that determines the ideal number of appraisers and queries required for the evaluation of *Q-WISE* on its *query suggestion* module, the usefulness of its generated *cluster labels,* and the quality of its *cluster summaries* (in Section 4.3). The performance analysis, which is based on a number of evaluation measures (introduced in Section 4.4) and the assessment of the appraisers, addresses (i) the effectiveness of our *trie* approach in extracting useful keywords for query suggestion (in Section 4.5.1), (ii) the accuracy of the (ranking on) cluster labels created by *Q-WISE* in capturing the content of clustered documents (in Section 4.5.2), (iii) the efficiency and effectiveness of *Q-WISE*'s document clustering approach (in Section 4.5.3), and (iv) the ability of *Q-WISE* to generate a summary that achieves a high degree of coverage, coherence, anti-redundancy, temporal dimension, and grammatical correctness (in Section 4.5.4). We have also measured the query processing time of *Q-WISE* in clustering retrieved documents and generating summaries (in Section 4.5.5) and compared the anticipated time to locate desired information on *Q-WISE* and Google, respectively as compiled by Facebook users[7] (in Section 4.5.6).

4.1 The datasets

In this section, we present the datasets used for (i) evaluating the effectiveness and efficiency of *Q-WISE*'s clustering approach (in Section 4.1.1), and (ii) measuring the quality of *Q-WISE*-created summaries, each of which captures a specific topic of a given query (in Section 4.1.2).

[7] Facebook users who served as independent appraisers provided an objective and unbiased evaluation on items (i), (ii), and (iv) listed above.

4.1.1 Data for evaluating the clustering approach

Results generated by a clustering algorithm are often evaluated against an "ideal" set of clusters. This *distortion measure* compares the difference between a set of generated clusters and a *ground truth set*, which is a *reference* set of clusters, on the same set of documents. We have evaluated the clustering results of *Q-WISE* using three datasets, denoted CS_1, CS_2, and CS_3 as shown in Table 1, which are constructed from various data sources widely used for document clustering evaluation. CS_3 is the 20 NewsGroup (http://people.csail.mit.edu/jrennie/20News groups/) dataset (20NG), which consists of 19,997 articles retrieved from the Usenet newsgroup collection that are clustered into 20 different categories. DMOZ (www.dmoz.org), on the other hand, is the largest, most comprehensive human-edited directory of the Web, which organizes web pages into their corresponding categories. We randomly selected 20 topic categories from DMOZ and extracted a total of 20,000 documents (the same size as the 20NG dataset) from the categories to create the multi-topic dataset CS_1. We chose another 20 categories from DMOZ along with a different set of 20,000 documents to obtain another dataset CS_2. The categories and documents in the three datasets are disjoint.

Dataset	CS_1	CS_2	CS_3
Number of articles (documents)	20,000	20,000	19,997
Number of topics	20	20	20
Data source	DMOZ	DMOZ	NewsGroup (20NG)

Table 1. Datasets used for evaluating the effectiveness and efficiency of *Q-WISE*'s clustering approach

4.1.2 Data for evaluating the summarization approach

Generic multi-document summarization analysis has been one of the fundamental tasks of DUC 2005, DUC 2006, and DUC 2007, each of which is an open benchmark dataset created and archived by the Document Understanding Conference, DUC (nlpir.nist.gov/projects/duc/). We used all three datasets for evaluating *Q-WISE*-generated summaries. Table 2 shows the properties

of the three datasets, where TDT (http://projects.ldc.upenn.edu/TDT/) and AQUAINT (http://www.ldc.upenn.edu/Catalog/docs/LDC2002 T31/) are corpora from where the DUC datasets are extracted.

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

Table 2. DUC datasets used for evaluating the quality of *Q-WISE*-created summaries

NIST assessors, who organized DUC and created each dataset as shown in Table 2, 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 250 words), well-organized, and fluent summary that captures the key concepts presented in *C* on *T*, which can be treated as a keyword query. The summary is compared with the *reference summary* of *C*, which was created by NIST assessors, to analyze its quality.

4.2 Number of appraisers and queries used for the controlled experiments

Prior to conducting the performance evaluation of *Q-WISE*, we determine the ideal number of appraisers (in Section 4.2.1) and test queries (in Section 4.2.2.) to be included in our empirical study so that the evaluation is reliable and objective.

4.2.1 The number of appraisers

In statistics, two types of errors, Type I and Type II, are defined [Jones 03]. Type I errors, also known as α 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 in fact false. We apply the following formula in [Jones 03] to determine the ideal number of appraisers, *n*, which is dictated by the probabilities of

occurrence of Types I and II errors in *Q-WISE*'s query suggestion/clustering/summarization modules:

$$n = \frac{\left(Z_{\frac{\propto}{2}} + Z_{\beta}\right)^2 * 2\sigma^2}{\Delta^2} + \frac{\left(Z_{\frac{\propto}{2}}\right)^2}{2} \quad (14)$$

where

- Δ is the *minimal expected difference* to compare *Q-WISE* with Google, which is set to 1 in our study;
- σ^2 is the *variance* of data (i.e., suggested keywords, cluster labels, and generated summaries), which is 3.85 for *query suggestion*, 6.00 for *clustering*, and 3.82 for *summarization* in our study;
- α denotes the probability of making a Type I error, which is 0.05 in our study;
- β denotes the probability of making a Type II error, which is 0.20 in our study. Based on the value of β, we can determine the probability of a false null hypothesis to be correctly rejected, i.e., 1 - β [Greene 00];
- *Z* is the value assigned to the standard *normal distribution* of suggested keywords, cluster labels, or generated summaries. According to the standard normal distribution, when α = 0.05, $Z_{\frac{\propto}{2}}$ = 1.96, whereas when β = 0.20, Z_{β} = 0.84.

To determine the values of Δ and σ^2 for evaluating the query suggestion, clustering, and summarization modules of *Q-WISE*, we conducted three individual experiments, one for each module, in November 2010, using a randomly sampled 100 test queries for each experiment from the AOL log. We chose only 100 queries, since the *minimal expected difference* and *variance*, which are computed on a *simple random sample*, do not change with a larger sample set of queries.

For query suggestion (clustering, respectively), we manually evaluated the *usefulness* of keywords suggested (cluster labels created, respectively) by *Q-WISE* for each test query, where A *useful* keyword suggestion (cluster label, respectively) should be closely related to the information need expressed in a query *Q*, which can be regarded as a subtopic of *Q*. For summarization, we manually assessed the *quality* of each cluster summary created by *Q-WISE* for a test query.

In computing the variance[8], i.e., σ^2, for query suggestion (clustering, respectively), we calculated the *mean* among the number of keywords (cluster labels, respectively) suggested (created, respectively) by *Q-WISE* that are relevant to a query *Q* and averaged the sum of the square difference between the mean and the actual number of (relevant) suggested keywords (cluster labels, respectively) created for each one of the 100 test queries. We obtained 3.85 (6.00, respectively), which is the value of σ^2 for query suggestion (clustering, respectively). In computing the variance for summarization, we averaged the sum of the square difference between the mean of the number of *useful* summaries[9] and the number of summaries generated by *Q-WISE* for each of the 100 test queries, which yielded 3.82 as the value of σ^2.

Based on the analysis conducted on query suggestion, clustering, and summarization for each one of the 100 test queries, we assigned the *minimal expected difference* for comparing *Q-WISE* and Google to be 1. $\Delta = 1$ implies that we expect *Q-WISE* to be at least as good as Google in terms of query suggestions, providing useful cluster labels (compatible with document titles provided by Google), and creating high-quality summaries (compatible with document snippets created by Google).

[8] *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 [Urdan 05].
[9] A summary is considered *useful* if it is of high quality (4 or 5 on a 5-point scale) as defined by DUC.

The values of α and β are set to be 0.05 and 0.20, respectively, which imply that we have 95% *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%. According to [Kazmier 03] and as stated in [Hinton 04], 0.05 is the commonly-used value for α, whereas 0.80 is a conventional value for (1 - β), and a test with $\beta \leq 0.20$ is considered to be statistically powerful.

Based on the values assigned to the variables in Equation 14, the ideal number of appraisers required in our study is 96 for *clustering* and 62 for both *query suggestion* and *summarization*, which are calculated as follows:

$$n\,(Query\ Suggestion) = \frac{(1.96 + 0.84)^2 * 2 * 3.85}{1^2} + \frac{1.96^2}{2} \cong 62$$

$$n\,(Clustering) = \frac{(1.96 + 0.84)^2 * 2 * 6}{1^2} + \frac{1.96^2}{2} \cong 96$$

$$n\,(Summarization) = \frac{(1.96 + 0.84)^2 * 2 * 3.82}{1^2} + \frac{1.96^2}{2} \cong 62$$

Note that the values of α, β, σ^2, and Δ directly influence the size of *n*. Furthermore, the results collected from the *n* appraisers in the study are expected to be comparable with the results that are obtained by the actual population [Jones 03], i.e., web users who query web search engines.

4.2.2 The number of test queries

In determining the ideal number of test queries to be included in the controlled experiments, we rely on two different variables: the *average attention span* of an adult and the *average number of search queries* that a person often creates in one session when using a web search engine. As mentioned in [Rozakis 02 and Schell 96], the average attention span of an adult is between twenty to thirty minutes. Furthermore, Jansen et al. [Jansen 00], who have evaluated web users' behavior especially on (i) the amount of time web users spend on a web search engine, (ii) the

average size of users' queries, and (iii) the average number of queries submitted by a user, estimate that the average number of queries created by each user on a web search engine in one session is approximately 2.8. In our empirical study, each appraiser was asked to evaluate *Q-WISE* using *nine* queries (three for each on query suggestion, clustering, and summarization). We consider *nine* to be an ideal number of queries for the performance analysis of *Q-WISE*, since evaluating *nine* queries takes approximately *thirty* to *forty* minutes, which falls in the time span of an adult. We randomly selected 186 queries (62 appraisers x 3 queries) from the AOL query log for query suggestion and summarization evaluation, respectively, 288 queries for evaluating the clustering approach of *Q-WISE* (96 appraisers x 3 queries), and 474 queries (= 288 + 186) for comparing the performance of *Q-WISE* and Google in terms of time required to locate desired information.

4.3 Performance evaluation of *Q-WISE*

We have developed various applications on Facebook so that Facebook appraisers can evaluate *Q-WISE*. We chose appraisers from Facebook, since it is a social network with users diverse in nationalities, ages, genders, and cultures, who can provide unbiased evaluations.

The Facebook appraisers are asked to evaluate the (i) *query suggestion module* of *Q-WISE* using the evaluation method proposed by [Efthimiadis 06] (see details in Section 4.3.1), (ii) *usefulness* of *Q-WISE*-generated *cluster labels* in capturing the various topics covered by *Q* (in Section 4.3.2), (iii) *quality* of each *cluster summary* determined by various factors (presented in Section 4.3.3), and (iv) time it takes an appraiser to locate desired information on *Q-WISE*, which is compared with Google (in Section 4.3.4). A separate Facebook application is created for each item (i) – (iv) listed above.

4.3.1 Evaluating query suggestions

To evaluate keywords suggested by *Q-WISE* for a test query *Q* and their corresponding rankings, we follow the evaluation strategy suggested by [Efthimiadis 06] who claims that users are the *best judge* in analyzing the performance of an interactive query suggestion system. Our evaluation assesses the keywords suggested for *Q* by three web search engines, Google, Yahoo!, and Bing, over the ranked, suggested keywords provided by *Q-WISE*. To accomplish this task, appraisers are given a set of suggested keywords recommended by the three search engines for *Q* and asked to select the top-25 most useful ones [Efthimiadis 06]. After selecting the useful suggested keywords, each appraiser ranks the top-5 most useful ones to *Q*. We compare the 25 selected and top-5 ranked keywords against the keywords suggested by *Q-WISE* and their corresponding rankings, respectively. Hereafter, appraisers' preferences are averaged to yield the measure on *Q-WISE*'s query suggestion module.

Given below are the detailed steps applied to evaluate *Q-WISE*'s query suggestion module.

1. Obtain a set of suggested keywords for each one of the 186 test queries *Q* from Google, Yahoo!, and Bing, and each appraiser who examines *Q* and its corresponding set of recommended keywords identifies 25 useful suggestions and provides the top-5 rankings.
2. Process *Q* through *Q-WISE* and obtain for *Q* a ranked list of suggested keywords.
3. Divide the ranked list of suggested keywords created in Step 2 into
    a. 2 halves (the *top half* and the *bottom half*).
    b. 3 parts (*top third*, *middle third*, and *bottom third*).
4. Match the appraiser's choices of suggested keywords created in Step 1 (by the three web search engines) for *Q* to each of the ranked lists generated in Step 3 for *Q*. For each list *L*,

compute the *percentage* of keywords in *L* that are also chosen by the appraiser, which yields a distribution of the appraiser's recommended keywords over *L*.

5. Match the appraiser's choice of the top-5 suggested keywords *KS* for *Q* to the ranked list created in Step 2 for *Q*. For the keywords in *KS*, *add* the difference in their rank positions with the ones provided by *Q-WISE*. If any of the keywords *K* in *KS* is not a keyword suggested by *Q-WISE*, then the ranking position of *K* is set to $N + 1$, where *N* is the number of suggested keywords created by *Q-WISE* for *Q*.
6. Use the well-known *Wilcoxon test* to compute the statistical significance on *Q-WISE*-suggested keywords.

The percentages computed in Step 4 are averaged, which yield an estimation on the distribution of useful keywords on the list of suggested keywords recommended by *Q-WISE*. Figure 14 shows an example of evaluating *Q-WISE*'s suggested keywords using the appraiser's preference on suggestion recommended by three web search engines for query "Tiger Woods".

Query: Tiger Woods

Set of suggested keywords from Google, Yahoo, and Bing (36 total): Baby, divorce, divorce settlement, cigar guy, online, net worth, photo, update, ryder cup, wiki, 2011, affair, daughter photo, house, girlfriend, news, elin nordegren, accident, pga tour, scandal, jokes, wife, voicemail, apology, game, gossip, biography, rehab, rumors, video, quits golf, wedding, workout, foundation, 2005 cheats, ps3

Appraiser-selected 25 most useful suggestion: Divorce, divorce settlement, net worth, photo, ryder cup, wiki, affair, house, girlfriend, news, accident, pga tour, scandal, wife, apology, game, gossip, biography, rehab, rumors, video, quits golf, wedding, foundation, ps3

Appraiser-ranked top-5 suggestions: divorce, wife, golf, game, biography

Ranked list of suggested keywords created by *Q-WISE*: divorce, affair, photos, wife, house, jokes, transgressions, crash photos, golf, yacht, girlfriends, cheats, news, misstress, scandal, biography, games, real name, cd key code, race, accident, pga tour 2010, polo, online game, family, kids, foundation, wedding, nike, website, commercial, neck injury, logo, boat, bio, conference, jets, schedule

Top Half: divorce, …, cd key code
Bottom Half: race, …, schedule
Top third: divorce, …, news
Middle third: misstress, …, kids
Bottom Third: foundation, …, schedule

Percantage of appraiser-selected 25 most useful suggestions in top half = 9/19 = 47%
Percantage of appraiser-selected 25 most useful suggestions in bottom half = 4/19 = 21%
Percantage of appraiser-selected 25 most useful suggestions in top third = 8/13 = 62%
Percantage of appraiser-selected 25 most useful suggestions in middle third = 5/13 = 39%
Percantage of appraiser-selected 25 most useful suggestions in bottom third = 1/12 = 8%

Figure 14. An example of evaluating query keywords suggested by *Q-WISE*

The *higher* the percentage on the top-half (top third, respectively) is, the *more effective Q-WISE*'s query suggestion module is. The percentages computed for *Q-WISE* are compared against the corresponding values achieved by other state-of-the-art query suggestion tools, including Bing and Yahoo! (See experimental results in Section 4.5.1).

Steps 5 and 6 measure the effectiveness of *Q-WISE* in *ranking* suggested keywords. The sum of the difference in the top-5 ranking positions (provided by the appraisers versus the ones by *Q-WISE*) of suggested keywords for a test query *Q* computed in Step 5 yield the input to the

Wilcoxon test in Step 6, which is a non-parametric test based on the differences between the ranking scores.

$$WT = \sum_{i=1}^{5} R_i \quad (15)$$

where R_i $(1 \leq i \leq 5)$ is the *signed-rank difference* between two ranked suggestions. We have chosen the Wilcoxon test, since it considers ranking of matched pairs and is freely available (http://faculty.vassar.edu/lowry/wilcoxon.html).

The *lower WT* is, the *closer* is the rankings of *Q-WISE*'s suggested keywords to an appraiser's rankings. The value of *WT* for *Q-WISE* is compared against the corresponding value achieved by other state-of-the-art query suggestion tools, including Google and Yahoo! (in Section 4.5.1) through the Facebook appraisers.

Example 1. Assume that the top-5 ranked suggested keywords provided by an appraiser for the test query "Tiger" are "animal", "Woods", "OS", "airways", and "clothes", respectively. Further assume that the rankings of these suggested keywords as recommended by *Q-WISE* are 3, 2, 5, 7, and 18[10], respectively, then

$$WT = (3-1) + (2-2) + (5-3) + (7-4) + (18-5) = 20.$$

4.3.2 Evaluating cluster labels

For each set of cluster labels created by *Q-WISE* in response to each one of the 288 randomly chosen test queries *Q*, the appraisers are required to measure the *usefulness* of cluster labels as defined in [Osinski 06]. A cluster label is *useful* if it includes *concise* and *meaningful* keywords with respect to *Q*, whereas a *useless* label is either *ambiguous* or includes *senseless* description. The Facebook appraisers mark each cluster label as *useful* or *useless*.

[10] Assuming that "clothes" is not a keyword suggested by *Q-WISE* for the query *Q* "Tiger", its ranking position is set to 18 (17 + 1), where 17 is supposed to be the total number of suggestions made by *Q-WISE* for *Q*.

Given a set of cluster labels *CL* created by *Q-WISE*, the quality of *CL* is measured as the *fraction* of all the cluster labels in *CL* that the appraisers judge as *useful*, which is defined as

$$Qual_{Val} = \frac{Usefulness}{Total_{Cls}} \quad (16)$$

where *Usefulness* is the total number of cluster labels judged as useful and *Total_Cls* is the total number of created cluster labels in *CL*. *Qual_Val* ranges from 0.0 (all cluster labels are *useless*) to 1.0 (all cluster labels are *useful*). The *higher Qual_Val* is, the *more useful Q-WISE*-created cluster labels are. The value of *Qual_Val* achieved by *Q-WISE* is compared against the corresponding value achieved by other state-of-the-art web search clustering tools (in Section 4.5.2).

4.3.3 Evaluating summaries

Using the DUC 2002, DUC 2003, and DUC 2005 datasets and an evaluation guideline, which is a set of *quality questions* developed in 2001 by Chin-Yew Lin at ISI [Lin 02], a summary created by a summarization system can be evaluated. These questions address the quality of *grammaticality*, *non-redundancy*, *referential clarity*, *structure* and *coherency*, and *responsiveness* on a generated summary.

1. *Grammaticality*: A high-quality summary should not exhibit any *datelines*, i.e., a sentence in the summary that contains only a date but no text, *system-internal formatting* (formatting such as html tags), *capitalization errors*, and obviously *ungrammatical sentences* (such as fragments and missing components) that jeopardize the readability of the text.

The grammatical questions specified in DUC 2002 and 2003 for evaluating the quality of the *grammar* in a generated summary are listed below [DUC-QUALITY 2003].

1. How many gross capitalization errors are there?
2. How many sentences have incorrect word order?

3. How many times does the subject fail to agree in number with the verb?
4. How many of the sentences are missing important components (e.g., the subject, main verb, direct object, and modifier) that cause the sentence to be ungrammatical, unclear, or misleading?
5. How many times are unrelated fragments joined into one sentence?
6. How many times is an article missing or used incorrectly?
7. How many dangling conjunctions are there (such as “and”, “however”, etc.)?

2. *Non-redundancy*: Unnecessary repetition can take the form of (i) whole sentences that are repeated, (ii) replicated facts, or (iii) repeated use of a noun or noun phrase, e.g., using "Mike Jones" when the pronoun "he" would suffice, which should be excluded from a summary. The redundancy question, “How many instances of unnecessarily repeated information are there?“ from DUC 2002 and 2003 [DUC-QUALITY 2003], is used for evaluating the degree of *anti-redundancy* in a generated summary.

3. *Referential Clarity*: It should be easy to identify who or what the pronouns and noun phrases in a summary are referring to. If a person or an entity is mentioned, its role in the summary should be clear. A reference is treated as *unclear* if an entity is referenced but its identity or relation to the content of the summary is uncertain.

The referential clarity questions from DUC 2002 and 2003 for evaluating the *referential clarity* of a generated summary are given below [DUC-QUALITY 2003].

1. How many pronouns are there whose antecedents are *incorrect*, *unclear*, *missing* or *come only later*?
2. How many nouns are impossible to determine clearly who or what they refer to?
3. How many times should have a noun or noun phrase been replaced with a pronoun?

4. *Structure and Coherence*: A summary should be well-structured and well-organized, which should not be a heap of related information but should be constructed from sentences to yield a *coherent* body of information on a topic instead. The coherence question, "About how many sentences suggest a wrong cause-effect relationship, or just don't fit in topically with neighboring sentences?" from DUC 2002 and 2003 [DUC-QUALITY 2003], is used for evaluating the *structure* and *coherence* of a summary.

5. The DUC 2005, DUC 2006, and DUC 2007 datasets include an evaluation on the *responsiveness* of a multi-document summary. *Responsiveness* is measured according to the amount of information in a summary that actually *address* the information need expressed in a topic statement, which is created by DUC assessors.

All of the guidelines listed above are measured on a 5-point scale as suggested by DUC. We have posted the 186 queries extracted randomly from the AOL query logs, their respective summaries created by *Q-WISE*, and the evaluation questions listed above on Facebook for the appraisers to evaluate.

4.3.4 *Q-WISE* vs. Google

To assess the ability of *Q-WISE* in assisting the user locate the desired information *at least as fast* as, or *faster* than a web search engine, Facebook appraisers are asked to evaluate the *time* it takes to satisfy the information need expressed in a test query using *Q-WISE* and Google, respectively. Facebook appraisers (i) select an arbitrary test query *Q* from the 474 queries randomly chosen from the AOL query log for comparing *Q-WISE* and Google, and the system to evaluate (i.e., Google or *Q-WISE*), (ii) look for the retrieved (created, respectively) titles and snippets (cluster labels and summaries, respectively) that address the information need expressed in *Q*, and (iii) record the time required to locate the information need specified in *Q*. To avoid

any bias on the evaluation, the *sources* of the results retrieved in response to a test query are *not* specified, and they are displayed using a heading and a text span. In the case of Google, each *heading* (*text span*, respectively) is the title (snippet, respectively) of a retrieved document, whereas in the case of *Q-WISE*, each heading is a *cluster label* and its text span is the first *sentence* of the corresponding *summary* created by *Q-WISE*. Figure 15 (16, respectively) shows a (portion of the) sample results page for the query "Dental Health" created by *Q-WISE* (Google, respectively). We include only the first three text spans in the two pages.

Dental Hygienist

A dental hygienist is a licensed dental professional who specializes in preventive oral health, typically focusing on techniques in oral hygiene.

Dental Guide

The Dental Guide has been a well-established guide to UK dentistry for over 10 years and is your one shop for any dental information both for the consumer and dental health professionals.

Dental Health News

Recently, dentistry has taken a new role in our live by adding an adjective to its process: comfort

Dentistry

…

Dental care Tips

…

Oral Hygiene

…

American Dental Association

…

Dental Health Treatment

…

Child Dental Health

…

Dental Procedures

…

Figure 15. The (portion of the) sample results page for the query "Dental Health" generated by *Q-WISE* (i.e., System 1)

Figure 16. The (portion of the) Sample results page for the query "Dental Health" generated by Google (i.e., System 2)

4.4 Evaluation measures

In this section, we present the evaluation measures that quantify the performance of *Q-WISE* on generating *summaries* that satisfy the information need of a user and the effectiveness and efficiency of its *clustering* approach. As stated in Section 4.3.3, a Facebook appraiser evaluates the grammar, anti-redundancy, coherence, referential clarity, and responsiveness of a summary *S*, whereas the ROUGE score (introduced in Section 4.4.1) evaluates the amount of information covered in *S* that address the corresponding query (topic) in a substantial way. Moreover, an

appraiser evaluates the quality of *Q-WISE*-generated cluster labels, whereas the F-measure (introduced in Section 4.4.2) evaluates the quality of its generated clusters.

4.4.1 Summarization

We consider the ROUGE toolkit (version 1.5.5), which is widely adopted for *summary evaluation*, to analyze the summarization performance of *Q-WISE* on the DUC datasets. ROUGE measures the quality of a summary by counting the *overlapped* units between a generated summary *S* 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 *S* performs, which is defined as

$$ROUGE_n = \frac{\sum_{S \in RefSum} \sum_{n-gram \in S} Count_{match}\left(n_{gram}\right)}{\sum_{S \in RefSum} \sum_{n-gram \in S} Count\left(n_{gram}\right)} \quad (17)$$

where n (≥ 1) is the length of the (overlapped) n-gram to be counted, $Count_{match}$ (n-gram) is the number of *overlapped* n-grams in *S* and the set of reference summaries *RefSum*, and $Count$ (n-gram) is the number of n-grams in the set of reference summaries. More specifically, we will compute ROUGE-2 (unigram- and bigram-based co-occurrence statistics), ROUGE-SU4 (trigram and 4-gram-based co-occurrence statistics), and ROUGE-BE (all co-occurrence statistics given that matched keywords have the same part of speech tag) to analyze the performance of *Q-WISE*.

The DUC website includes the ROUGE-2, ROUGE-SU4, and ROUGE-BE scores of 30 multi-document summarization systems for each dataset, which we compare *Q-WISE* against.

4.4.2 Clustering

Clusters generated by *Q-WISE* are evaluated against the *ideal* clusters (created by humans) of a given dataset using the widely-known *F-measure*, which is based on precision and recall. *Precision* (*Recall*, respectively) is the *ratio* of the number of common documents in an *ideal*

cluster *I* and a *generated* cluster *C* to the total number of documents in *C* (*I*, respectively). The *higher* the *F-measure* is, the *better* the clustering method performs. We have compared our clustering results with other state-of-the-art tools in web document clustering (See Section 4.5.3 for details).

4.5 Performance evaluations

Having introduced the evaluation measures for assessing the performance of *Q-WISE*, we first analyze the *effectiveness* of its trie approach in recommending useful query keywords (in Section 4.5.1). Hereafter, we present the *degree of relevance* of cluster labels created by *Q-WISE* with respect to the contents of the corresponding clusters based on the assessments provided by the Facebook appraisers involved in our controlled experiment (in Section 4.5.2). We also evaluate the *effectiveness* and *efficiency* of *Q-WISE*'s clustering approach (in Section 4.5.3), and the quality of *Q-WISE*-created summaries (in Section 4.5.4). We measure the query processing time of *Q-WISE* in generating clusters and their corresponding summaries (in Section 4.5.5) and the anticipated time to locate desired information on *Q-WISE* and Google, respectively (in Section 4.5.6). In each section, we compare and contrast the performance of the evaluated component of *Q-WISE* with the corresponding state-of-the-art tools.

4.5.1 Results on query suggestion

We have collected the evaluations from 62 Facebook appraisers who examined the query keywords suggested by *Q-WISE* on each one of the 186 test queries. Based on the evaluations, we computed the (i) averaged percentage on the appearance (rankings, respectively) of suggested keywords for each test query on each of the five parts, i.e., top-half, bottom-half, top-third, middle-third, and bottom-third, and (ii) the Wilcoxon score (as defined in Section 4.3.1). These percentages were compared against the ones achieved by Yahoo!, Bing, and a baseline measure

using the AOL query log. The baseline measure determines suggested queries (i.e., query keywords) solely based on the frequency of the queries in the query log, i.e., the number of times an original user's query is modified to the suggested ones. The results are depicted in Figures 17 and 18, respectively.

We briefly introduce the *hitting time* algorithm, *query flow graphs*, and *query frequency* used by Bing, Yahoo!, and the baseline measure, respectively in creating a ranked list of query suggestions. Note that Google is not considered in this study, since its query suggestion algorithm is not available to the public and thus cannot be compared.

Bing: MSN search (now known as Bing) constructs a bipartite graph using a query log[11] and computes *hitting time* (i.e., click frequency) to determine query keywords to be recommended for a user's query. A bipartite graph $G = (V_1 \cup V_2, E)$ consists of a query set V_1, which is the set of queries extracted from the query log, and a URL set V_2, which is the set of URLs specified in the log. An edge in *E* from a query *i* to an URL *k* denotes that *k* was clicked by the user who submitted *i* to the search engine, and the edge is *weighted* by the *click frequency*, denoted *w*(*i, k*), which indicates the number of times *k* is clicked when *i* is the search query. During the query suggestion phase, a modified subgraph *SG* is constructed from *G* using the depth-first search approach on a user's query *Q* with queries in V_1 so that the search of suggested queries for *Q* stops when the number of nodes representing suggested queries on *SG* is larger than a predefined number of *n* queries (set by Bing). *SG* is a modified subgraph of *G*, since each URL node *k* in *SG*, which is also in the original *G*, is removed, but the connected edges of *k* are retained. Hereafter, Bing defines the *transition probability* between any two queries, *i* and *j*, linked by an edge on *SG*, which computes the degree of similarity between *i* and *j* as

[11] In comparing with *Q-WISE*, we use the same AOL query log on each of the three query suggestion systems, i.e., Bing, Yahoo!, and the baseline measure.

$$p_{ij} = \sum_{k \in V_2} \frac{w(i,k)}{d_i} * \frac{w(j,k)}{d_j} \tag{18}$$

where *k* is a URL in $G, d_i = \sum_{k \in V_2} w(i,k)$, and $d_j = \sum_{k \in V_2} w(j,k)$. For all the queries exhibited in *SG*, excluding the one being searched, Bing retrieves the queries with the top-*n largest* transition probabilities with respect to the user's query *i* as suggestions, where *n* is set to be 10 by Bing.

Yahoo!: The Yahoo! search engine recommends query keywords by using *query flow graphs*. A query flow graph is an aggregated representation of the latent querying behavior contained in a query log. Intuitively, in a query flow graph, a directed edge from query q_i to query q_j denotes that the two queries are part of the same search session. Any nodes on a path in a query flow graph may be seen as potential query suggestions, whose likelihood is given by the *strength* of the edges along the path. A directed edge (q_i, q_j) is labeled with (i) the *probability* that a user creates q_j after q_i in the same search session (i.e., q_j is a possible suggestion to q_i), and (ii) the type of the transition, known as the Query Reformulation Type (QRT). Query Reformulation Types indicate the relationship between two queries, *i* and *j*, linked by a directed edge in a query flow graph, which include *Equivalent Rephrasing* (i.e., *j* has the same meaning as *i* but is rephrased differently), *Parallel Move* (i.e., *i* and *j* are two subtopics of the user's query), *Generalization* (i.e., *j* is a generalization of *i*), *Specialization* (i.e., *j* is a special case of *i*), etc. In a query flow graph created by the Yahoo! Search engine, edges are directed and are annotated with two labels, a *weight* and the *QRT*, as given by a pre-trained model[12]. Weights are defined as

$$w(q,q') = r(q,q') \,/ \sum_{k:(q,k) \in E} r(q,k) \tag{19}$$

[12] A group of Yahoo! *editors* manually labeled the set of query pairs (q, q') in each search session using one of the reformulation types to create the training dataset.

where $r(q, q')$ denotes the number of times query q has been directly modified to query q' as shown in a query log, and the weight $w(q, q')$ represents the *probability* of query q and query q' being in the same querying session. A small excerpt of a query flow graph is shown in Figure 19, where *S* and *P* denote the *Specialization* and *Parallel Move* query reformulation types, respectively.

Yahoo! retrieves the top-*n* nodes (queries) based on their edge weights in a query flow graph, such that the chosen nodes, which are suggested queries, have the highest edge weights among all the nodes that are either directly connected to a user's query node or any indirect nodes that are connected to a node already retrieved as a suggestion, where *n* is set to be 10 by Yahoo!. As shown in Figure 19, given the user's search query "barcelona", the query suggestions retrieved in order are "Barcelona fc" (0.08), "Barcelona weather" (0.04), "Barcelona hotels" (0.02), "cheap Barcelona hotels" (0.07), and "luxury Barcelona hotels" (0.03).

Baseline: A query suggestion approach is often compared against a baseline measure, which ranks query suggestions strictly based on their *frequencies* of occurrence in a query log. The list of queries suggested by a baseline measure is created by retrieving all queries in the query log that include the user's query *Q* as a *substring*. The more *frequent* a query suggestion *S*, which includes *Q* as a substring in the query log is, the *higher S* is ranked.

Among the four systems involved in the evaluation, Figure 17 shows that *Q-WISE* achieves a relatively high percentage over the *top-half* and *top-third* and a relatively low percentage on the *bottom-half* and *bottom-third*, which imply that query keywords suggested by *Q-WISE* are useful, since they are often chosen by Facebook appraisers. Although Yahoo! has a slightly higher percentage than *Q-WISE* over the *top-half*, *Q-WISE* achieves a much higher percentage on the *top-third*, which suggest that useful keywords are often ranked higher by *Q-*

WISE than by Yahoo! (and Bing, in addition to the baseline measure, respectively). Among all the systems involved in the performance evaluation, *Q-WISE* achieves the *lowest* Wilcoxon test score, as shown in Figure 18, which indicates that the ranking of query keywords suggested by *Q-WISE* is the closest to the Facebook appraisers' provided rankings, and *Q-WISE*'s ranking strategy is the *most effective* compared with Yahoo!, Bing, and the baseline measure.

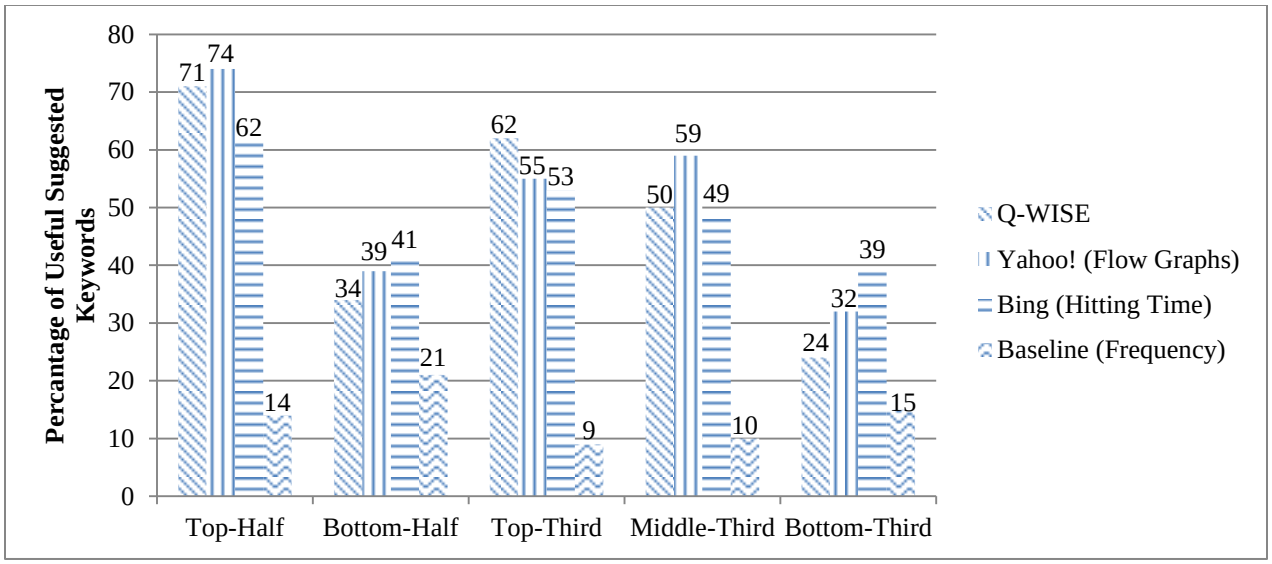


Figure 17. Averaged percentages of useful suggested keywords over the 5 parts on the ranked query suggestion lists created by *Q-WISE*, Yahoo!, Bing, and the baseline measure, respectively computed using Facebook appraisers' evaluations

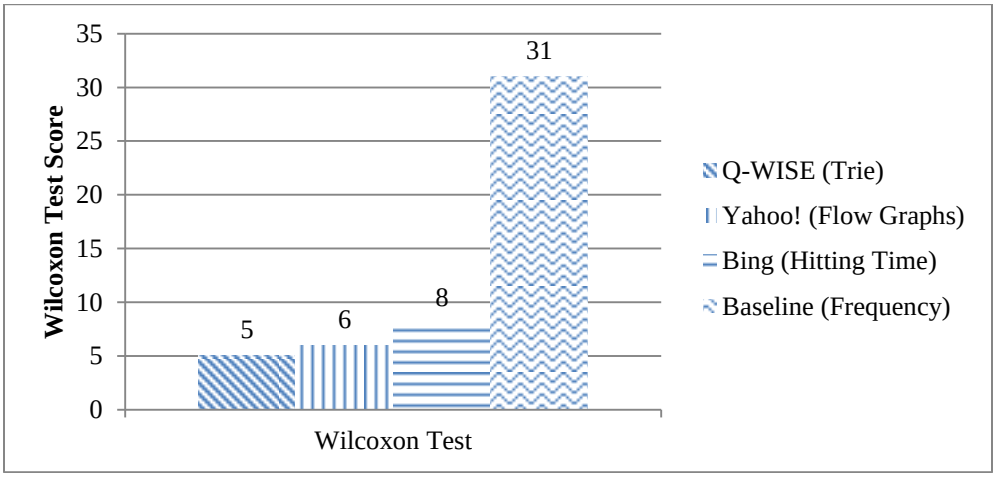


Figure 18. The Wilcoxon test scores for *Q-WISE*, Yahoo!, Bing, and the baseline measure, respectively based on the rankings compiled by Facebook appraisers

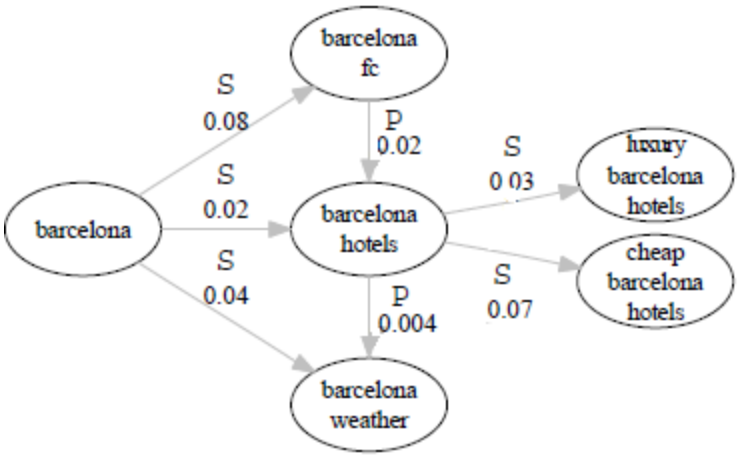


Figure 19. An excerpt of the query flow graph around the query "*barcelona hotels*", which is created by using the Yahoo! UK query log

4.5.2 Quality of *Q-WISE*-created cluster labels

We have collected the evaluations provided by the 96 Facebook appraisers who examined and assessed the usefulness of *Q-WISE*-created cluster labels for each one of the 288 test queries (as discussed in Section 4.3.2). Based on the assessments of the appraisers, 92.2% of *Q-WISE*-created cluster labels were considered useful. Figure 20 shows the average percentages of useful cluster labels determined by 20 out of the 96 Facebook appraisers involved in the evaluation. The percentages of useful cluster labels are almost always in the upper eightieth and ninetieth percentile. The few cases where the average percentages are below 80 are due to the *low quality* of retrieved titles and snippets from where the labels were constructed.

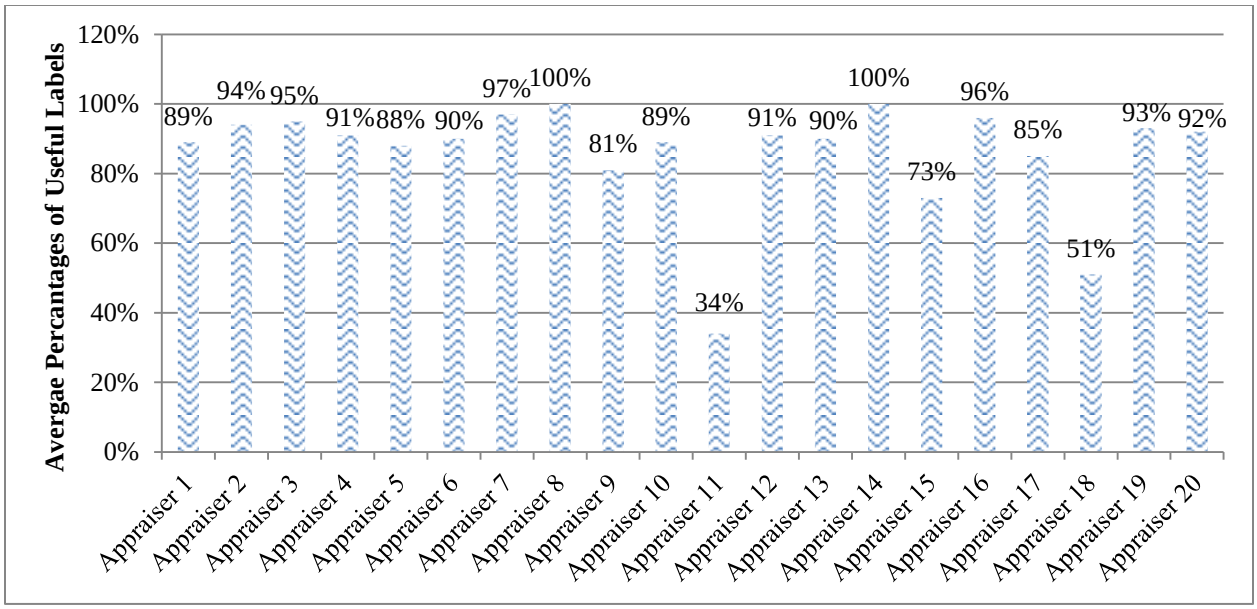


Figure 20. Average percentages of useful *Q-WISE*-created labels determined by 20 (out of the 96) Facebook appraisers

4.5.3 The effectiveness and efficiency of *Q-WISE*'s clustering approach

We have evaluated the effectiveness and efficiency of *Q-WISE*'s clustering approach using the three datasets *CS1*, *CS2*, and *CS3* defined in Section 4.1.1. *Effectiveness* was measured using the well-known *F-measure*, whereas *efficiency* was based on the *processing time* required to cluster each one of the three datasets. The *F-measure* and *processing time* are compared against the ones achieved by WhatsOnWeb [Giacomo 07] and Armil [Geraci 06] using the same three datasets. We chose the two clustering methods for comparisons, since (i) unlike other clustering approaches, their algorithms are publicly available and (ii) they outperform the Vivismo (www.vivismo.com) clustering engine, which is considered an industrial standard on the evaluation of clustering quality and user satisfaction[13]. The *F-measure* achieved by each of the three clustering approaches and their *processing time* are depicted in Figures 21 and 22, respectively.

[13] Vivismo won the annual "best meta-search award" offered by the online magazine SearchEngineWatch.com in 2001 and 2002.

We briefly introduce the clustering approaches of WhatsOnWeb and Armil on web search results, respectively below.

WhatsOnWeb: WhatsOnWeb [Giacomo 07] clusters web documents retrieved in response to a user query *Q* using a topology driven approach, which relies on a *snippet graph*. Intuitively, a snippet graph captures the URLs retrieved by a web search engine in response to *Q*, along with their relationships, which are determined by analyzing the *snippets* of the retrieved URLs. Each vertex of a snippet graph is a URL, and an edge connecting two URLs indicates that the corresponding snippets share some stemmed keywords. The document clusters are derived from a snippet graph by finding communities of vertices, which are sets of vertices that have high edge *weights* from a topological point of view, created by using a recursive divisive strategy based on graph connectivity.

The *snippet graph G* of a set of URLs *U* retrieved in response to *Q* is a labeled, weighted graph defined as follows:

- Each vertex v_u in *G* denotes a URL $u \in U$, and the *label* of v_u is the *title* of *u*.
- *T*he *label* of an edge $e = (v_{u1}, v_{u2})$ in *G*, where $u_1, u_2 \in U$, is the concatenation of all the common *sentences*, which are maximal subsequences of consecutive stemmed keywords, shared between the two snippets S_{u1} and S_{u2} of u_1 and u_2, respectively that are connected by *e*. The *score of a sentence* σ shared between S_{u1} and S_{u2} is defined as $w_\sigma = \sum_{k \in \sigma} w_k$, where w_k is the *degree of relevance* of the common keyword *k* in the two snippets with respect to the snippets computed using the standard vector space model, the *weight* of *e* is $w_e = \sum_{\sigma \in S_e} w_\sigma$, and S_e is the concatenation of all sentences shared by S_{u1} and S_{u2}.

WhatsOnWeb is one of the few web clustering engines based on graph theory. A drawback of WhatsOnWeb is the slow response time required for processing a query due to the inefficiency of generating clusters from a snippet graph.

Armil: To cluster documents retrieved in response to a search query, Armil [Geraci 06] maps the snippets of retrieved results into a vector space endowed with a distance function, which is treated as a metric. Hereafter, a modified furthest-point-first algorithm (M-FPF) is applied to generate the clusters of snippets. Given a set of points *N* in a vector space, each of which denotes a retrieved document, and an integer *k*, which denotes the number of clusters to be created, the FPF algorithm first selects an arbitrary point $n_1 \in N$ and picks the next point (up till the k^{th} point), which maximizes the distance of all the chosen points. Eventually, the k chosen points represent the k most dissimilar documents. Armil creates a cluster C_j $(1 \leq j \leq k)$ by including all the points (documents) closer to n_j than to any other points, where n_j is a chosen point from where C_j is created. The M-FPF algorithm generates the same clusters of the standard FPF algorithm, except that it avoids unnecessary distance computation to speed up the computation. In contractually to FPF, M-FPF relies only on pairwise distance calculations between snippets. *Q-WISE*, however, does not depend on a (i) distance function or (ii) pre-determined number of clusters for performing clustering, which could differ from one query to another and may affect the clustering quality if chosen inappropriately.

Among the three systems involved in the clustering performance evaluation, *Q-WISE* achieves the *highest F-measure*, as shown in Figure 21, which indicates that clusters generated by *Q-WISE* on each one of the three datasets used in the evaluation are the closest to the pre-determined *ideal* set of clusters (as defined in Section 4.1.1) than the other two. Figure 22 shows that *Q-WISE* is the *fastest* clustering method compared with its counterparts.

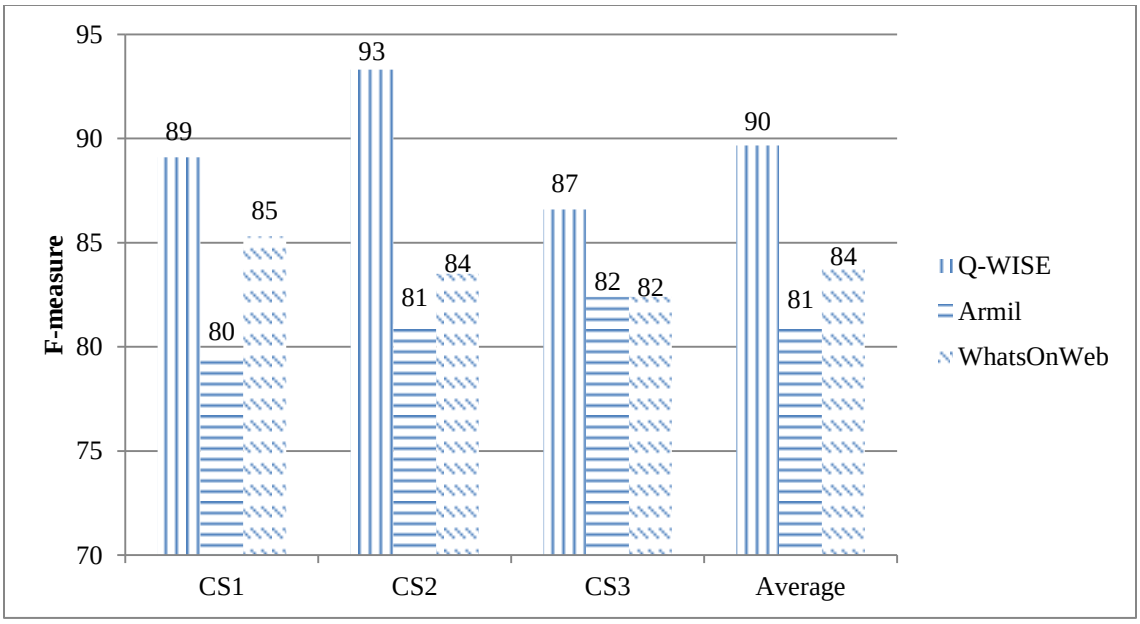


Figure 21. (Average) *F-measure*(*s*) over the three datasets used for evaluating the *effectiveness* of the clustering approach of *Q-WISE*, Armil, and WhatsOnWeb, respectively

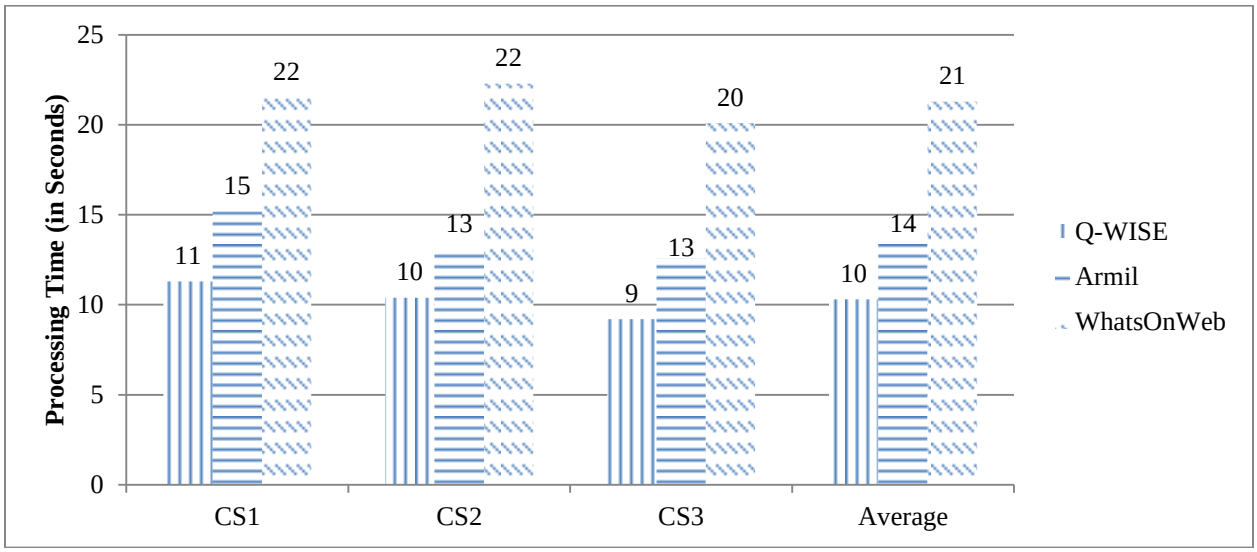


Figure 22. (Average) *processing time* for clustering each one of the three datasets *CS*1, *CS*2, and *CS*3 by *Q-WISE*, Armil, and WhatsOnWeb, respectively

4.5.4 Quality of *Q-WISE*-created summaries

We have collected the evaluations provided by 62 Facebook appraisers who reviewed the summary created by *Q-WISE* in response to each one of the 186 test queries. Based on the appraisers' evaluations, we computed the (i) quality of the grammar, (ii) degree of anti-

redundancy, (iii) referential clarity, (iv) structure and coherence, and (v) responsiveness of each *Q-WISE*-created summary (as defined in Section 4.3.3). Besides the appraisers' evaluations, the *quality* of a *Q-WISE*-created summary, which is measured using the information captured by the summary, was quantified using the documents in the three DUC datasets and the evaluation metrics introduced in Sections 4.1.2 and 4.4.1, respectively. The results are depicted in Figures 23 and 24, respectively. We have also compared the quality (scores) of *Q-WISE*-created summaries against thirty automatic multi-document summarization systems as shown in Tables 3 and 4, respectively[14].

As illustrated in Table 3, *Q-WISE* achieves the highest score on *non-redundancy*, the second highest on *referential clarity* and *responsiveness*, the fourth on *structure and coherence*, and the fifth on *Grammar*. The comparatively lower scores on grammar, in addition to structure and coherence, among the five quality measures are due to the fact that *Q-WISE*'s summarization approach is *extractive*, which is not sophisticated in connecting extracted sentences in a summary. The extractive summarization approach adopted by *Q-WISE*, however, is not a major drawback, since compared with the thirty summarizers, *Q-WISE* is ranked in the top five on each measure.

Table 4 shows that *Q-WISE* achieves the second (third, respectively) highest ROUGE-2 and ROUGE-SU4 (ROUGE-BE, respectively) score(s) among the thirty summarizers involved in the evaluation. This indicates that the information included in *Q-WISE*-created summaries are of high quality, i.e., *Q-WISE*'s summaries address a user query in a substantial way, compared with other low ranking summarization systems.

[14] The summarization performance results of the thirty systems are posted under the DUC website.

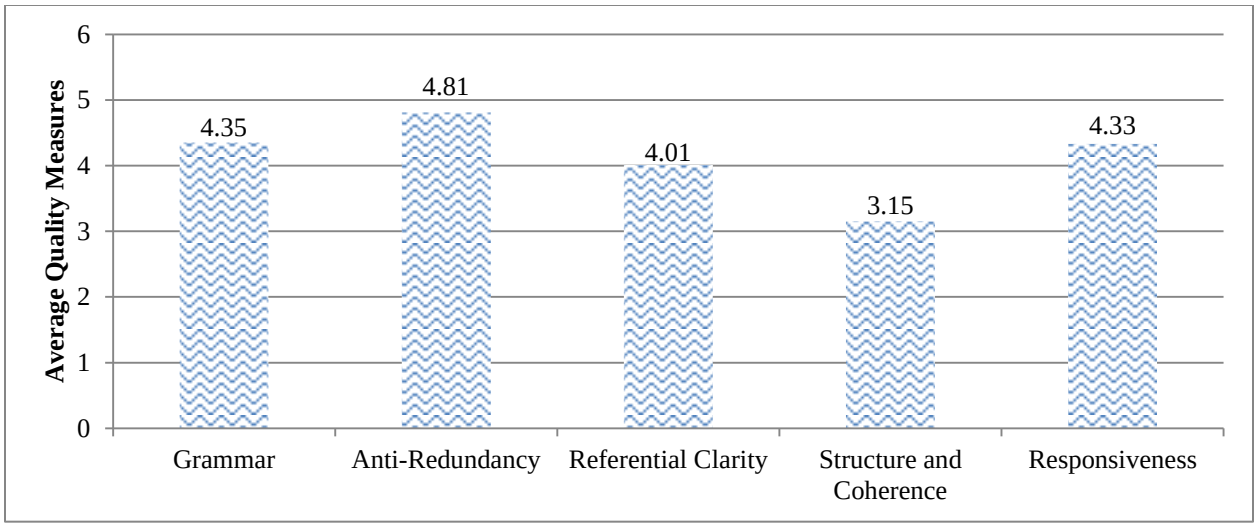


Figure 23. The average quality measures, on the scale of 1-5, of *Q-WISE*-generated summaries provided by Facebook appraisers

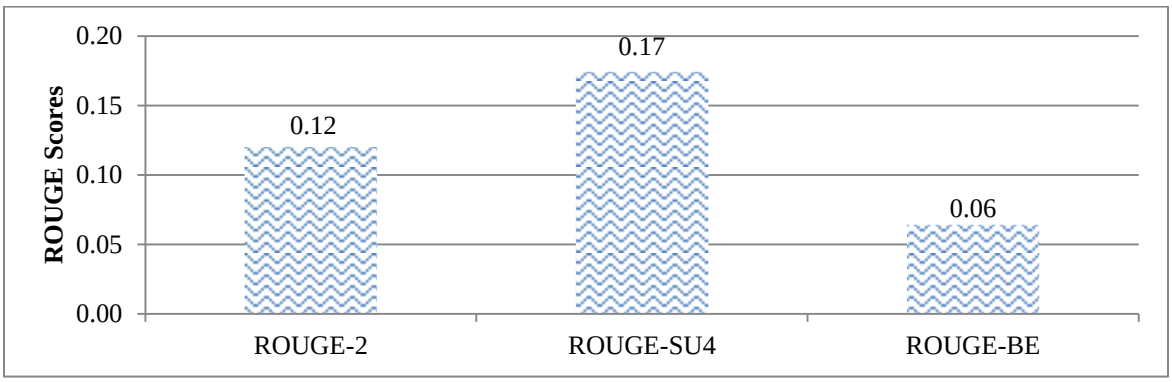


Figure 24. The ROUGE scores achieved by *Q-WISE* using the three DUC datasets

	Score	*Q-WISE* is Out-Performed by	*Q-WISE* Outperforms
Grammatically	4.35	5	25
Non-redundancy	4.81	1	29
Structure & Coherence	3.15	4	26
Referential Clarity	4.01	2	28
Responsiveness	4.33	2	28

Table 3. Comparisons between *Q-WISE* and the thirty summarizers participated in DUC in terms of the five quality measures

	Score	*Q-WISE* is Out-Performed by	*Q-WISE* Outperforms
ROUGE-2	0.12	2	28
ROUGE-SU4	0.17	2	28
ROUGE-BE	0.06	3	27

Table 4. Comparisons between the quality (based on the ROUGE scores) of *Q-WISE*-created summaries and the ones created by the thirty summarizers participated in DUC in response to each test query

4.5.5 Query processing time of *Q-WISE*

We have measured the *processing time* of the query suggestion, clustering, and summarization modules of *Q-WISE*, respectively using 474 queries randomly selected from the AOL query log. As shown in Figure 25, the processing time required to generate query suggestions for a user's input is on an average of 0.07 seconds, which indicates that *Q-WISE* generates query suggestions instantly while the user is formulating his/her query, and is comparable to Yahoo!, Google, and Bing. The processing time required to generate document clusters and their corresponding labels is on an average of 0.23 seconds. Last, but not least, *Q-WISE* generates a summary in less than 2 seconds on an average. Figures 26, 27, and 28 show the processing time for query suggestions, clustering, and summarization, respectively on 15 selected queries.

As reported in [Carpineto 09], fetching documents from a public search engine using its API takes up a significant portion of the total query processing time, which increases the time required to generate a summary using *Q-WISE*. Table 5 shows the average time required to fetch 200 documents from Yahoo!, Google, and Bing, respectively on 40 queries, as reported in [Carpinto 09].

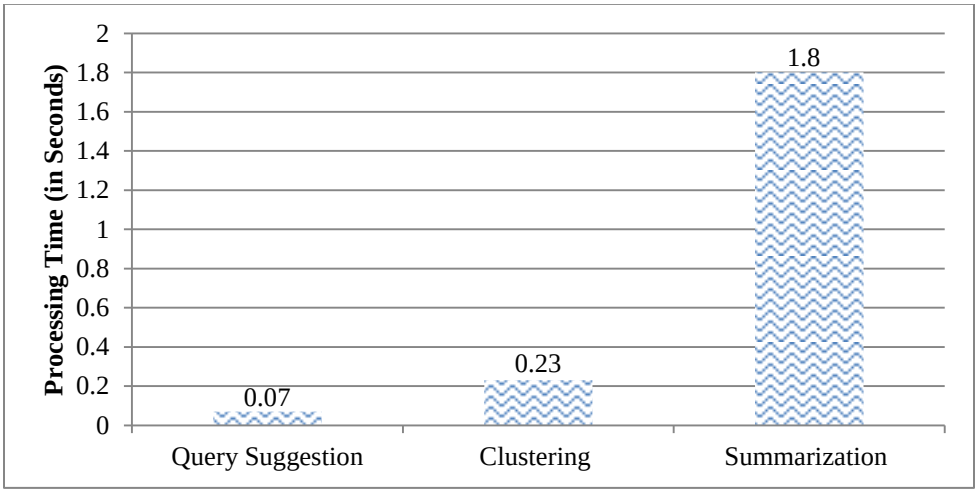


Figure 25. Average processing time of *Q-WISE*'s query suggestion, clustering, and summarization approaches

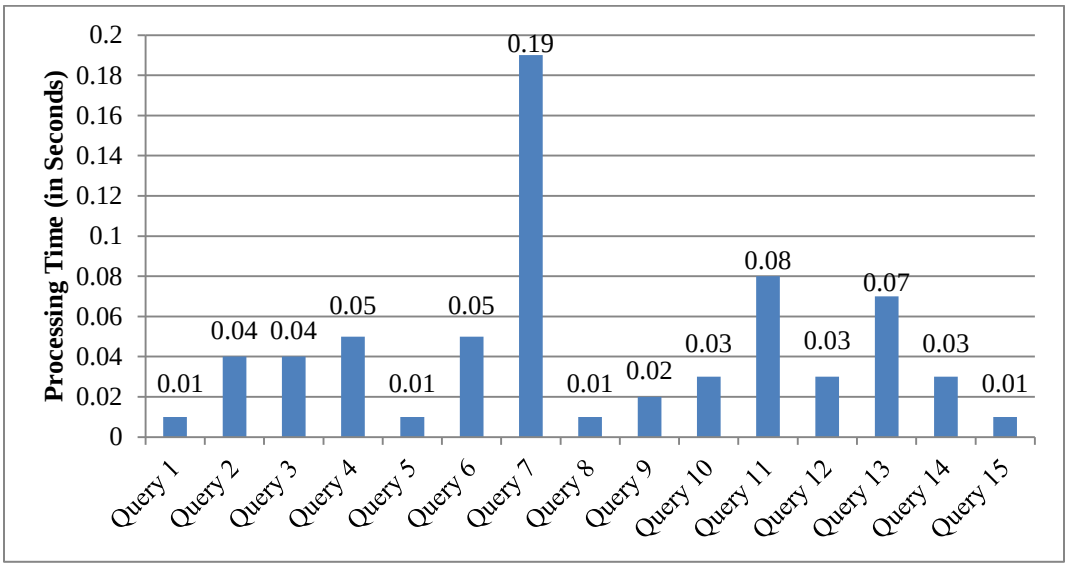


Figure 26. Processing time on query suggestions for 15 (out of 474) queries

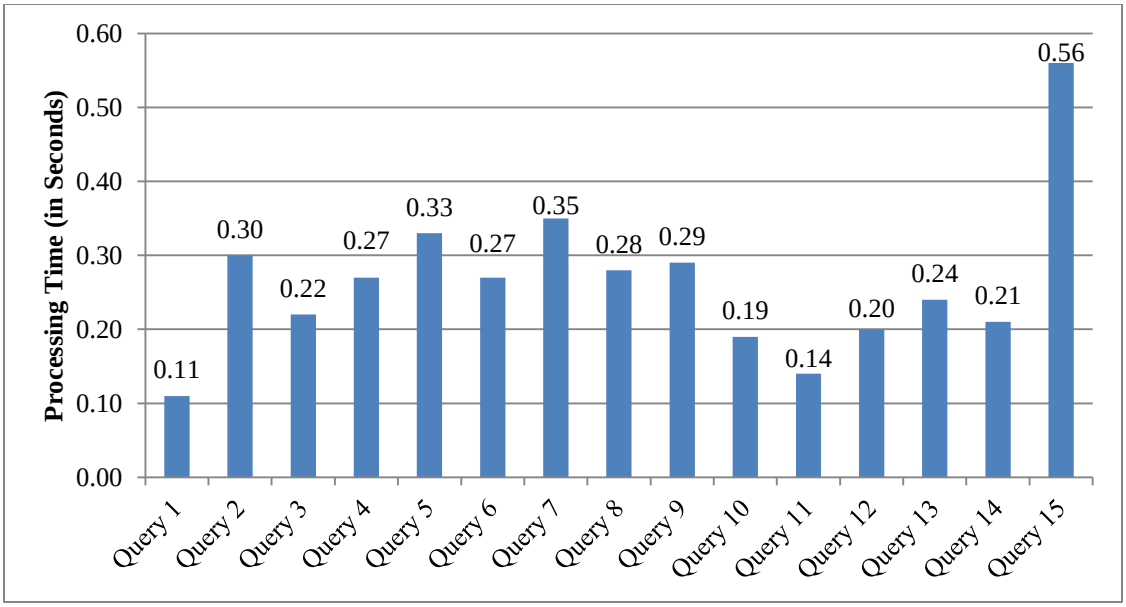


Figure 27. Processing time on clustering retrieved documents for 15 (out of 474) queries

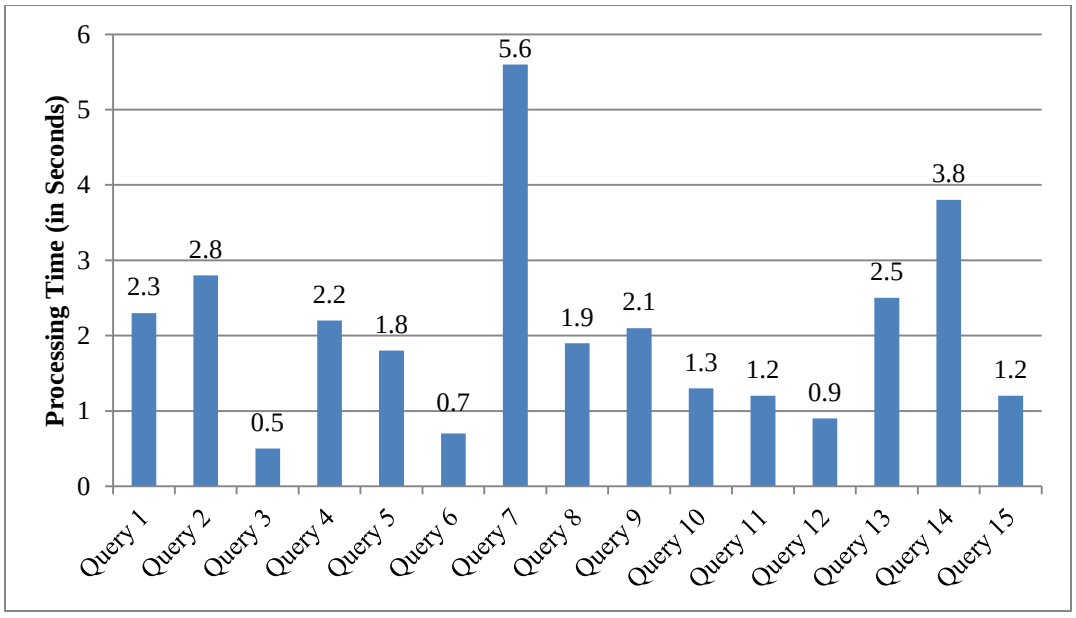


Figure 28. Processing time on summarizing clustered documents for 15 (out of 474) queries

	Average Delay	Standard Deviation
Yahoo!	2.12	0.65
Google	5.85	2.35
Bing	0.56	0.43

Table 5. Average processing time required by Yahoo!, Google, and Bing using their APIs, respectively to extract 200 search results for each one of the 40 queries

4.5.6 *Q-WISE* vs. Google

We have collected the evaluations provided by 158 Facebook appraisers who have compared the time in locating desired information retrieved by *Q-WISE* and Google, respectively on each one of the 474 test queries (as described in Section 4.3.4). The evaluations show that it takes a Facebook appraiser an average of 63 (72, respectively) seconds to locate the desired information on Google (*Q-WISE*, respectively). The difference is less than 10 seconds on an average. Figure 29 shows the average time required to find information on *Q-WISE* and Google, respectively, which are the processing time reported by 20 out of the 158 Facebook appraisers involved in the evaluation.

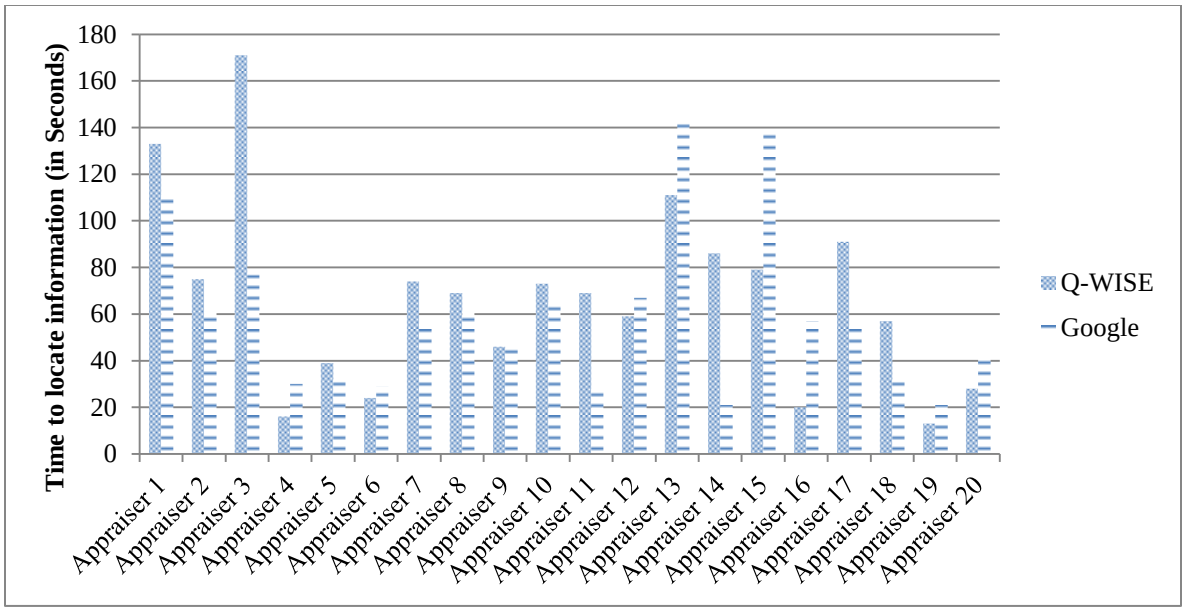


Figure 29. Average time required to locate desired information on *Q-WISE* and Google, respectively, which are reported by each one of the 20 (out of 158) Facebook appraisers

As shown in Figure 29, the time required to locate information on *Q-WISE* is comparable to Google (less than 10 seconds difference on average), which is one of the most popular, efficient, and effective commercial search engines. This does not mean that *Q-WISE* can replace or is better/worse than Google. However, *Q-WISE* offers a different search experience that some users might find more effective than traditional search. The same 158 Facebook appraisers who evaluated *Q-WISE* and Google were asked if they would consider replacing Google with *Q-WISE*. The given choices were "yes", "no", or "depends on the type of search". 127 appraisers (80% of appraisers) answered "depends on the type of search".

Q-WISE introduces a search method that helps users view different interpretations of their query and an abstract on their intended information need. This might not be useful to users trying to locate an article or a paper, but is very useful to users who are trying to obtain general information on a particular subject and/or a summary of the main points. Moreover, *Q-WISE* facilitates the search process by identifying groups of documents clustered based on different

interpretations of the user's query. While this may be a bad feature to users who are specific in their search and don't want to see different interpretation of the query, it is helpful to a majority of web search users who write short ambiguous queries.

As *Q-WISE* is fully automated and requires no human feedback, it cannot distinguish worthwhile from useless sentences. I have noticed in some cases, that a sentence in a *Q-WISE*-generated summary is just a list of words that do not form a useful sentence. Such sentences could generate from bibliographies, tables, images, tag clouds, etc. Q-WISE needs more work to preprocess retrieved documents more efficiently and avoid such sentences. Moreover, sentences in a *Q-WISE*-generated summary are based on text features, i.e., a sentence's position and length, which might affect the quality of the summary. More analysis needs to be made to determine what people regard as relevant and useful sentences for a particular topic. Finally, *Q-WISE* can only work for documents in English.

Chapter 5

Conclusions

Current web search engines, such as Google, Yahoo!, and Bing, offer users a mean to locate desired information available on the Web. In response to a user query, a search engine retrieves a list of ranked documents *S* and displays 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 identify precisely the set of documents that satisfy the user's intended request. In addition, since snippets are created using sentences/phrases in the corresponding retrieved documents solely based on the keywords appeared in the user query, they (i) may not always capture the main content of the corresponding documents and (ii) are similar in contents and thus are not useful in distinguishing the contents of the corresponding documents.

In order to improve existing web searches, we have developed a query-based, web informative summarization engine, denoted *Q-WISE*, which (i) assists its users in formulating their queries using a simple, yet effective trie-based query suggestion module, (ii) clusters retrieved documents based on the various topics covered in the documents which match the interpretations of the information needs specified in a user query, (iii) assigns a meaningful label to each cluster *C*, which identifies the topic/contents of the documents in *C*, and (iv) summarizes documents on each specified topic to assist the user quickly identify the results of interest. Experimental results using well-known datasets and our Facebook applications show that *Q-WISE* (i) recommends useful query suggestions, (ii) solves the ambiguity problem of search queries by considering the various interpretations of each query, (iii) provides an effective and efficient algorithm, which clusters retrieved documents related in contents based on the various

topics covered in the interpretations, and (iv) assists the user locate desired information in time comparable to commercial web search engines using created summaries.

Q-WISE is a significant contribution to the web search community, since it (i) is user-friendly, (ii) solves the ambiguous problem on interpreting the intended information need of a web user, (iii) clusters closely related documents that could be scattered all over the ranked result set, and (iv) provides an enhanced snippet, i.e., cluster summary of each document cluster which assists users to quickly identify information of interest.

Regarding future work, we plan to examine using an open source search engine, such as Nutch (http://nutch.apache.org/) or Lucene (http://lucene.apache.org/java/docs/index.html), instead of querying commercial web search engines, to retrieve the top-100 documents to generate topic-based clusters and summaries. A stand-alone search engine (i) can reduce the overhead required to fetch and download results from existing web search engines and (ii) avoids the access restrictions imposed by existing search engines API's, which allow only a limited number of queries per day and thus reduce the number of users who can use *Q-WISE* on a daily basis. We will also consider new features, such as noisy data, frequency of information in sentence, importance of document using authors and/or page traffic, etc., to enhance our summarization approach. The additional features will be analyzed for their accuracy in selecting informative sentences using multiple regression analysis.

References

[Alguliev 08] R. Alguliev and R. Alyguliev. Automatic Text Documents Summarization through Sentences Clustering. Automation and Information Sciences, Vol. 40, pp. 53-63. 2008.

[Altman 05] A. Altman and M. Tennenholtz. Ranking Systems: The PageRank Axioms. In Proceedings of the ACM Conference on Electronic Commerce (EC), pp. 1-8. 2005.

[Amini 09] M. Amini and N. Usunier. Incorporating Prior Knowledge into a Transductive Ranking Algorithm for Multi-Document Summarization. In Proceedings of the International Conference on Research and Development in Information Retrieval (SIGIR), pp 704-705. 2009.

[Anick 03] P. Anick. Using Terminological Feedback for Web Search Refinement: A Log-based Study. In Proceedings of the Annual International Conference on Research and Development in Informaion Retrieval (SIGIR), pp. 88-95. 2003.

[Arora 08] R. Arora and B. Ravindran. Latent Dirichlet Allocation Based Multi-Document Summarization. In Proceedings of the Second Workshop on Analytics for Noisy Unstructured Text Data (AND), pp. 91-97. 2008.

[Baraglia 10] R. Baraglia, C. Castillo, D. Donato, F. Nardini, R. Perego, and F. Silvestri. The Effects of Time on Query Flow Graph-based Models for Query Suggestion. In Proceedings of the International Conference on Adaptivity, Personalization and Fusion of Heterogeneous Information (RIAO). 2010.

[Baxendale 58] P. Baxendale. Machine-Made Index for Technical Literature – An Experiment. IBM Journal of Research and Development. 1958.

[Berger 99] A. Berger and J. Lafferty. Information Retrieval as Statistical Translation. In Proceedings of the Conference on Research and Development in Information Retireval (SIGIR), pp. 222-229. 1999.

[Bernard 08] C. Bernard. Filtering Search: A New Approach to Query-Answering. In Proceedings of Foundations of Computer Science, pp. 122-132. 2008.

[Bhandari 08] H. Bhandari, M. Shimbo, T. Ito, and Y. Matsumoto. Generic Text Summarization using Probabilistic Latent Semantic Indexing. In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), pp. 133-140. 2008.

[Boldi 08] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The Query Flow Graph: Model and Applications. In Proceedings of the Conference on Information and Knowledge Managment (CIKM), pp. 609-618. 2008.

[Boldi 09] P. Boldi, F. Bonchi, C. Castillo, D. Donato, and S. Vigna. Query Suggestions Using Query Flow Graphs. In Proceedings of the Conference on Web Search Click Data (WSCD), pp. 56-63. 2009.

[Bonchi 09] F. Bonchi, P. Boldi, C. Castillo, and S. Vigna. From ‘Dango’ to ‘Japanese Cakes’: Query Reformulation Models and Patterns. In Proceedings of the Conference on Web Intelligence (WI), pp. 183-190. 2009.

[Brashler 09] M. Braschler. Multilingual Information Retrieval Based on Document Alignment Techniques. Research and Advanced Technology for Digital Libraries, pp.183-197. 2009.

[Cao 08] H. Cao, D. Jiang, J. Pei, Q. He, A. Liao, E. Chen, and H. Li. Context-aware Query Suggestion by Mining Click-through and Session Data. In Proceedings of the Conference on Knowledge Discovery and Data Mining (KDD), pp. 875-883. 2008.

[Carpineto 09] C. Carpineto, S. Mizzaro, G. Romano, and M. Snidero. Mobile Information Retrieval with Search Results Clustering: Prototypes and Evaluations. The American Society for Information Science and Technology, Vol. 60, pp. 877-895. 2009.

[Chen 11] L. Chen. Using a New Relational Concept to Improve the Clustering Performance of Search Engines. Information Processing and Management, in Press. 2011.

[Cheng 05] D. Cheng, S. Vempala, R. Kannan, and G. Wang. A Divide and Merge Methodology for Clustering. In Proceedings of the Symposium on Principles of Database Systems (PODS), pp. 196-205. 2005.

[Chim 08] H. Chim and X. Deng. A New Suffix Tree Similarity Measure for Document Clustering. In Proceedings of the International Conference on World Wide Web (WWW), pp. 121-130. 2008.

[Crabtree 05] D. Crabtree, X. Gao, and P. Andreae. Improving Web Clustering by Cluster Selection. In Proceedings of the International Conference on Web Intelligence (WI), pp. 172-178. 2005.

[Croft 10] B. Croft, D. Metzler, and T. Strohman. Search Engines: Information Retrieval in Practice. 2010.

[Daume 05] H. Daume and D. Marcu. Induction of Word and Phrase Alignments for Automatic Document Summarization. Computational Linguistics, Vol. 31, Issue 4, pp. 505-530. 2005.

[Dunlavy 07] D. Dunlavy, D. O'Leary, J. Conroy, and J. Schlesinger. QCS: A System for Querying, Clustering, and Summarizing Documents. Information Processing and Management, Vol. 43, pp. 1588-1605. 2007.

[Efthimiadis 00] E. Efthimiadis. Interactive Query Expansion: A User-based Evaluation in a Relevance Feedback Environment. The American Society for Information Science, Vol. 51, Issue 11, pp. 989-1003. 2000.

[Erkan 04] G. Erkan and D. Radev. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Artificial Intelligence Research, Vol. 22, pp. 457-479. 2004.

[Ferragina 08] P. Ferragina and A. Guli. A Personalized Search Engine Based on Web-snippet Hierarchical Clustering. Software: Practice and Experience, pp. 189-225. 2008.

[Fonseca 05] B. Fonseca, P. Golgher, B. Possas, B. Ribeiro-Neto, and N. Ziviani. Concept-based Interactive Query Expansion. In Proceedings of the International Conference on Information and Knowledge Management (CIKM), pp. 696–703. 2005.

[Geraci 06] F. Geraci, M. Pellegrini, P. Pisati, and F. Sebastiani. A Scalable Algorithm for High Quality Clustering of Web Snippets. In Proceedings of the 21st Annual ACM Symposium on Applied Computing (SAC), pp. 1058-1062. 2006.

[Giacomo 07] E. Giacomo, W. Didimo, L. Grilli, and G. Liotta. Graph Visualization Techniques for Web Clustering Engines. IEEE Transactions on Visualization and Computer Graphics, Vol. 13, Issue 2, pp. 294-304. 2007.

[Goldstein 00] J. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-Document Summarization by Sentence Extraction. In Proceedings of the Geometrical Models of Natural Language Semantics Workshop on Automatic Summarization, pp. 40-48. 2000.

[Greene 00] W. Greene. Econometric Analysis. Prentice Hall. 2000.

[Grossberg 88] S. Grossberg. Nonlinear Neural Networks: Principles, Mechanisms, and Architectures. Neural Networks, Vol. 1, pp. 17-61. 1988.

[Gulla 07] J Gulla, H Borch, and J. Ingvaldsen. Contextualized Clustering in Exploratory Web Search. Emerging Technologies of Text Mining: Techniques and Applications, Chapter IX, pp. 184-207. 2007.

[Gyongyi 08] Z. Gyongyi and G. Koutrika. Questioning Yahoo! Answers. In Proceedings of the International World Wide Web Conference (WWW). 2008.

[Hearst 96] M. Hearst and J. Pedersen. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR), pp. 76-84. 1996.

[Hennig 09] L. Hennig. Topic-based Multi-Document Summarization with Probabilistic Latent Semantic Analysis. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP), pp. 144-149. 2009.

[Hinton 04] P. Hinton. Statistics Explained: A Guide for Social Science Students. Routledge. 2004.

[Jansen 00] B. Jansen, A. Spink, and T. Saracevic. Real Life, Real Users, and Real Needs: a Study and Analysis of User Queries on the Web. Information Processing and Management, Vol. 36, Issue 2, pp. 207-227. 2000.

[Jones 03] B. Jones and M. Kenward. Design and Analysis of Cross-Over Trials, Second Edition. Chapman and Hall. 2003.

[Judea 88] P. Judea. Probabilistic Reasoning in the Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann. 1988.

[Jung 07] S. Jung, J. Herlocker, and J.Webster. Click Data as Implicit Relevance Feedback in Web Search. Information Processing and Management, Vol. 43, Issue 3, pp. 791-807. 2007.

[Kang 03] H. Kang and G. Kim. Query Type Classification for Web Document Retrieval. In Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR), pp. 64-71. 2003.

[Kazmier 03] L. Kazmier. Schaum's Outline of Business Statistics. McGraw-Hill. 2003.

[Khoo 02] C. Khoo, S. Ou, and D. Goh. A Hierarchical Framework for Multi-document Summarization of Dissertation Abstracts. In Proceedings of the International Conference on Asian Digital Libraries (ICADL), pp. 99-110. 2002.

[Kibble 00] R. Kibble and D. Kees. Coreference Annotation: Whither? In Proceedings of the International Conference on Language Resources and Evaluation (LREC), pp. 1281-1286. 2000.

[Knuth 73] D. Knuth. The Art of Computer Programming: Fundamental Algorithms. The Art of Computer Programming Series, Vol. 1. 1997.

[Koberstein 06] J. Koberstein and Y.-K. Ng. Using Word Clusters to Detect Similar Web Documents. In Proceedings of the First International Conference on Knowledge Science, Engineering, and Management (KSEM), pp. 215-228. 2006.

[Leskovec 04] J. Leskovec, M. Grobelnik, and N. Milic-Frayling. Learning Sub-Structures of Document Semantic Graphs for Document Summarization. In Proceedings of Link Analysis: Dynamics and Static of Large Networks (LinkKDD), pp. 133-138. 2004.

[Li 09] H. Li, C. Sun, and K. Wang. Clustering Web Search Results using Conceptual Grouping. In Proceedings of the International Conference on Machine Learning and Cybernetics, pp. 1499-1503. 2009.

[Lin 02] C. Lin and E. Hovy. From Single to Multi-Document Summarization: A Prototype System and its Evaluation. In Proceedings of Associations for Computer Linguistics (ACL), pp. 457-464, 2002.

[Lin 09] H. Lin, J. Bilmes, and S. Xie. Graph-based Submodular Selection for Exctractive Summarization. In Proceedings of the Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 381-386. 2009.

[Liu 04] S. Liu, F. Liu, C. Yu, and W. Meng. An Effective Approach to Document Retrieval via Utilizing WordNet and Recognizing Phrases. In Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR), pp. 266-272. 2004.

[Liu 08] X. Liu and W. Croft. Evaluating Text Representations for Retrieval of the Best Group of Documents. In Proceedings of the Conference on Advances in Information Retrieval (IR), pp. 454-462. 2008.

[Lugar 05] G. Luger. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. London: Addison-Wesley Pearson Education. 2005.

[Luhn 58] H. Luhn. The Automatic Creation of Literature Abstracts. IBM Journal of Research and Development, Vol. 2, pp. 159–165. 1958.

[Manning 99] C. Manning and H. Schtze. Foundations of Statistical Natural Language Processing. MIT Press. 1999.

[Mei 08] Q. Mei, D. Zhou, and K. Church. Query Suggestion using Hitting Time. In Proceedings of the Conference on Information and Knowledge Management, pp. 469-478. 2008.

[Mihalcea 04] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 404-411. 2004.

[Osinski 06] S. Osinski. Improving Quality of Search Results Clustering with Approximate Matrix Factorisations. In Proceedings of the Conference on Information Retrieval Research (ECIR), pp. 167-78. 2006.

[Ou 07] S. Ou, C. Khoo, and D. Goh. Automatic Multi-document Summarizationfor Digital Libraries. A-LIEP, pp. 72-82. 2007.

[Radev 04] D. Radev, H. Jing, M. Stys, and D. Tam. Centroid-based Summarization of Multiple Documents. Information Processing and Management, Vol. 40, Issue 6, pp. 919-938. 2004.

[Rozakis 02] L. Rozakis. Test Taking Strategies and Study Skills for the Utterly Confused. McGraw Hill. 2002.

[Ruch 06] P. Ruch, I. Tbahriti, J. Gobeill, and A. Aronson. Argumentative Feedback: A Linguistically-Motivated Term Expansion for Information Retrieval. In Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING/ACL), pp. 675-682. 2006.

[Ruixu 08] L. Ruixu and J. Whang. A New Cluster Merging Algorithm of Suffix Tree Clustering. In Proceedings of the International Conference on Intelligent Information Processing (IIP), pp. 197-203. 2008.

[Schell 96] C. Schell and R. Rathe. Geri Ann: Designing Educational Programs for the Internet. Gerontology and Geriatrics Education (Journal of the Association for Gerontology in Higher Education), Vol. 16, Issue 4, pp. 15-25. 1996.

[Schiffman 02] B. Schiffman, A. Nenkova, and K. McKeown. Experiments in Multidocument Summarization. In Proceedings of the Human Language Technology Conference, pp. 52-58. 2002.

[Schlesinger 08] J. Schlesinger, D. Leary, and J. Conroy. Arabic/English Multi-document Summarization with CLASSY – The Past and the Future. In Proceedings of International Conference on Intelligent Text Processing and Computational Linguistics, pp. 568-581. 2008.

[Selberg 99] E. Selberg, S. Tanimoto, and O. Etzioni. Towards Comprehensive Web Search. Univeristy of Washington. 1999.

[Shekhar 10] S. Shekhar and R. Agrawal. An Architectural Framework of a Crawler for Retrieving Highly Relevant Web Documents by Filtering Replicated Web Collections. In Proceedings of the International Conference on Advances in Computer Engineering, pp. 29-30. 2010.

[Shen 06] D. Shen and R. Pan. Query Enrichment for Web-Query Classification. ACM Transactions on Information Systems, Vol. 24, No. 3, pp. 320-352. 2006.

[Spink 00] A. Spink and J. Xu. Selected Results from a Large Study of Web Searching: the Excite Study. Information Research. Available at: http://informationr.net/ir/6-1/paper90.html. 2000.

[Trout 10] J. Trout and S. Rikin. Repositioning: Marketing in an Era of Competition, Change, and Crisis. Available at: http://kenhoma.wordpress.com/2010/03/03/how-many-pages-are-added-to-the-web-each-day/. 2010.

[Urdan 05] T. Urdan. Statistics in Plain English, 2nd Edition. Lawrence Erlbaum. 2005.

[Vectomova 06] O. Vectomova and Y. Wang. A Study of the Effect of Term Proximity on Query Expansion. Information Science, Vol. 32, Issue 4, pp. 324–333. 2006.

[Wang 09] D. Wang, S. Zhu, T. Li, and Y. Gong. Multi-Document Summarization using Sentence-based Topic Models. In Proceedings of the International Joint Conference on Natural Language Processing (IJCNLP), pp. 297-300. 2009.

[Watson 10] http://www.markwatson.com/opensource/

[Weiss 01] Y. Weiss, A. Ng, and M. Jordan. On Spectral Clustering: Analysis and an Algorithm. In Proceedings of the Conference on Neural Information Processing Systems (NIPS), pp. 849-856. 2001.

[Xide 10] C. Xide, Y. Yu, J. Han, and B. Liu. Hierarchical Web-Page Clustering via In-Page and Cross-Page Link Structures. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), pp. 222-229. 2010.

[Yih 07] T. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-Document Summarization by Maximizing Informative Content-Words. In Proceedings of the International Joint Conference on Artificial Intelligence, pp. 307-314. 2007.

[Yu 05] P. Yu, X. Li, and B. Liu. Adding the Temporal Dimension to Search – a Case Study in Publication Search. In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI), pp. 543 – 549. 2005.

[Zamir 98] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR), pp. 46-54. 1998.

[Zamir 99] O. Zamir and O. Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results. Computer Networks, Vol. 31, Issue. 11-16, pp. 1361–1374. 1999.

[Zeng 04] H. Zeng, Q. He, Z. Chen, and W. Ma. Learning to Cluster Web Search Results. In Proceedings of the Conference on Research and Development in Information Retrieval (SIGIR), pp. 210-217. 2004.

[Zhang 01] D. Zhang and Y. Dong. Semantic, Hierarchical, Online Clustering of Web Search Results. In Proceedings of the International Workshop on Web information and Data Management (WIDM), pp. 69-78. 2001.