Structure-Based Query-Specific Document Summarization

Ramakrishna Varadarajan
School of Computing and Information Sciences
Florida International University
Miami, FL 33199
rvara001@cs.fiu.edu

Vagelis Hristidis
School of Computing and Information Sciences
Florida International University
Miami, FL 33199
vagelis@cs.fiu.edu

ABSTRACT

Summarization of text documents is increasingly important with the amount of data available on the Internet. The large majority of current approaches view documents as linear sequences of words and create query-independent summaries. However, ignoring the structure of the document degrades the quality of summaries. Furthermore, the popularity of web search engines requires query-specific summaries. We present a method to create query-specific summaries by adding structure to documents by extracting associations between their fragments.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]; H.3.1 [**Content Analysis and Indexing**]

General Terms: Algorithms, Performance, Experimentation

Keywords: query-specific summarization, keyword search, adding structure to documents, Steiner tree problem, user survey

1. OVERVIEW

As the number of documents available on users' desktops and the Internet increases, so does the need to provide high-quality summaries in order to allow the user to quickly locate the desired information. The Information Retrieval (IR) community has largely viewed text documents as linear sequences of words for the purpose of summarization. Although this model has proven quite successful in efficiently answering keyword queries, it is clearly not optimal since it ignores the inherent structure in documents.

Furthermore, most summarization techniques are query-independent and follow one of the following two extreme approaches: Either they simply extract relevant passages viewing the document as an unstructured set of passages, or they employ Natural Language Processing techniques. The former approach ignores the structural information of documents while the latter is too expensive for large scale datasets (e.g., the Web) and sensitive to the writing style of the documents.

In this work, we propose a method to add structure to text documents in order to allow effective query-specific summarization. That is, we view a document as a set of interconnected text fragments (passages). We focus on keyword queries since keyword search is the most popular information discovery method on documents, because of its power and ease of use.

Our technique has the following key steps: First, at the preprocessing stage, we add structure to every document, which can then be viewed as a labeled, weighted graph, called the *document graph*. Then, at query time, given a set of keywords, we perform keyword proximity search on the document graphs to discover how the keywords are associated in the document graphs. For each document its summary is the minimum spanning tree on the corresponding document graph that contains all the keywords (or equivalent based on a thesaurus).

The document graph is constructed as follows. First we parse the document and split it to text fragments using a delimiter (e.g., the new line character). Each text fragments becomes a node in the document graph. A weighted edge is added to the document graph between two nodes if they either correspond to adjacent text fragments in the text or if they are semantically related, and the weight of an edge denotes the degree of the relationship. There are many possible ways to define the degree of the relationship between two text fragments. In this work we consider two fragments to be related if they share common words (not stop words) and the degree of relationship is calculated by an adaptation of traditional IR term weighting formulas. We also consider a thesaurus to enhance the word matching capability of the system. To avoid dealing with a highly interconnected graph, which would lead to slower execution times and higher maintenance, cost, we only add edges with weight above a threshold. Also notice that the edge weights are query-independent, so they can be precomputed.

Example 1 Figure 2 shows the document graph for the document of Figure 1, which has been trimmed due to lack of space. The document is first split to text fragments *v0...v16,* which correspond to its paragraphs.(other delimiters are also possible). Notice that the edge between nodes *v8* and *v7* has the highest weight because there are many infrequent (and hence with higher *idf* value) words that are common between them, like "Donoghue" and "BrainGate".

At query time, the precomputed document graph of a document is processed as follows to create the best query-specific summary. First, each node of the document graph is assigned a score according to the relevance of the corresponding text fragment to the query. To do so we employ traditional IR ranking functions. Notice that a full-text index is used to accelerate this step. Then, we execute our keyword proximity algorithms, which are inspired by the techniques developed in previous work on proximity search on graphs [1,2], where approximation algorithms are presented for the Group Steiner Tree problem (which is equivalent[1] to the

[1] The proximity problem is slightly harder since the sets of nodes do not have to be disjoint.

proximity search problem). The best summary is the top-ranked spanning tree that contains all the keywords. The ranking considers both the node and the edge weights (which are query-dependent and independent respectively). Notice that the problem can be easily modified to allow summaries that do not contain all keywords, although this case is not further discussed.

(v0) Brain chip offers hope for paralyzed
(v1) A team of neuroscientists have successfully implanted a **chip** into the **brain** of a quadriplegic man, allowing him to control a computer.
(v2) ...
(v3) ...
(v4) Results of the pilot clinical study will be presented to the Society for Neuroscience annual conference in San Diego, California, on Sunday. Up to five more patients are to be recruited for further **research** into the safety and potential utility of the device.
(v5) ...
(v6) ...
(v7) John Donoghue, professor of neuroscience at Brown and a co-founder of Cyberkinetics in 2001, said that BrainGate could help paralyzed peopled control wheelchairs and communicate using email and Internet-based phone systems.
(v8) ...
(v9) ...
(v10) Donoghue's initial **research**, published in the science journal Nature in 2002, consisted of attaching an implant to a monkey's **brain** that enabled it to play a simple pinball computer game remotely.
(v11) The four-millimeter square **chip**, which is placed on the surface of the motor cortex area of the **brain**, contains 100 electrodes each thinner than a hair which detect neural electrical activity. The sensor is then connected to a computer via a small wire attached to a pedestal mounted on the skull.
(v12) ...
(v13) ...
(v14) Surgeon Gerhard Friehs, associate professor of clinical neurosciences at Brown Medical School, who implanted the device, described the results as "spectacular" and "almost unbelievable."
(v15) "Here we have a **research** participant who is capable of controlling his environment by thought alone -- something we have only found in science fiction so far," said Friehs.
(v16) ...

Figure 1: Sample news document from www.cnn.com.

Example 1 (cont'd) Table 1 shows the top-ranked spanning trees for the document graph of Figure 2 for the query "Brain chip research". The values shown above the nodes in Table 1 indicate the node scores with respect to the query. Notice that all results contain all query keywords. The top result is the best summary of the document of Figure 1 (the keywords of the query are shown in bold) for this query. Intuitively, this result is the best because it contains the minimum possible number of nodes and the edge that connects the two nodes is strong.

Also observe that Result #4 is ranked lower than Result #3 even though it has fewer nodes. The reason is that the nodes of Result #4 are connected through very commonly occurring words like "computer" and "brain" whereas in Result #3 they are connected through infrequent words like "Friehs". Notice that to compute the frequency of a keyword we consider all documents of the corpus.

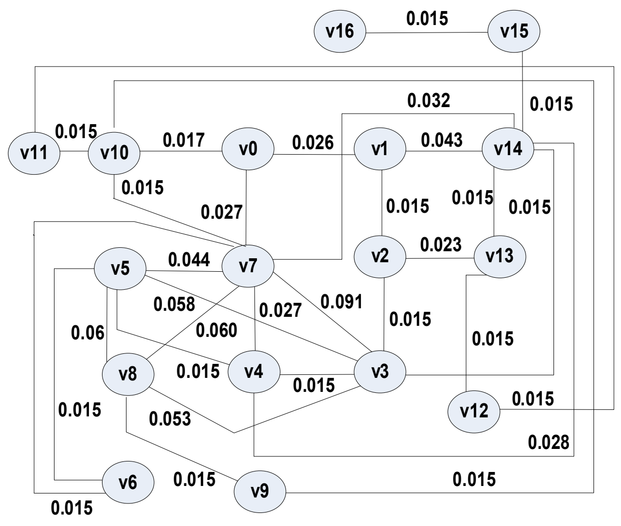


Figure 2: Document Graph for the document in Figure 1.

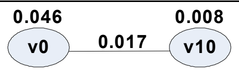
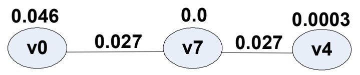
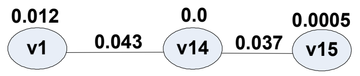
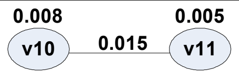
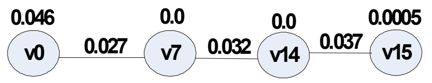
Rank	Score	Summary
1	67.74	
2	84.77	
3	87.64	
4	103.77	
5	167.41	

Table 1: Top-5 summaries for query "Brain Chip Research".

To evaluate the quality of the summarization of our approach, we conducted a survey. The dataset used in the survey consists of two news articles taken from the technology section of cnn.com, while the subjects are fifteen FIU students. The participants were asked to evaluate the quality of the summaries produced by our approach, compared to Google Desktop and MSN Desktop result snippets. Our approach was rated 78% (resp. 68%) higher than Google Desktop (resp. MSN Desktop).

REFERENCES

[1] G. Bhalotia, C. Nakhe, A. Hulgeri, S. Chakrabarti and S. Sudarshan: Keyword Searching and Browsing in Databases using BANKS. ICDE, 2002

[2] R. Goldman, N. Shivakumar, S. Venkatasubramanian and H. Garcia-Molina: Proximity Search in Databases. VLDB, 1998