Structure-preserving and query-biased document summarisation for web searching

F. Canan Pembe
Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey and Department of Computer Engineering, İstanbul Kültür University, Istanbul, Turkey, and
Tunga Güngör
Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey

Refereed article received
19 July 2008
Approved for publication
20 January 2009

Abstract

Purpose – The purpose of this paper is to develop a new summarisation approach, namely structure-preserving and query-biased summarisation, to improve the effectiveness of web searching. During web searching, one aid for users is the document summaries provided in the search results. However, the summaries provided by current search engines have limitations in directing users to relevant documents.

Design/methodology/approach – The proposed system consists of two stages: document structure analysis and summarisation. In the first stage, a rule-based approach is used to identify the sectional hierarchies of web documents. In the second stage, query-biased summaries are created, making use of document structure both in the summarisation process and in the output summaries.

Findings – In structural processing, about 70 per cent accuracy in identifying document sectional hierarchies is obtained. The summarisation method is tested on a task-based evaluation method using English and Turkish document collections. The results show that the proposed method is a significant improvement over both unstructured query-biased summaries and Google snippets in terms of f-measure.

Practical implications – The proposed summarisation system can be incorporated into search engines. The structural processing technique also has applications in other information systems, such as browsing, outlining and indexing documents.

Originality/value – In the literature on summarisation, the effects of query-biased techniques and document structure are considered in only a few works and are researched separately. The research reported here differs from traditional approaches by combining these two aspects in a coherent framework. The work is also the first automatic summarisation study for Turkish targeting web search.

Keywords Data structures, Document delivery, Markup languages, Search engines, Worldwide web

Paper type Research paper

Introduction

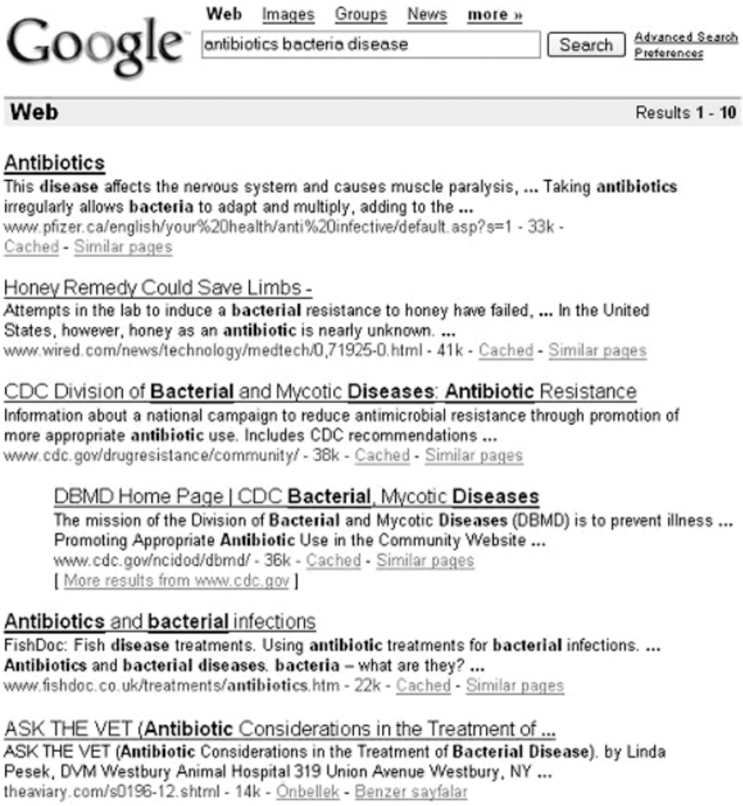
The drastic increase in documents available on the world wide web has resulted in the wide-spread problem of information overload (Mani and Maybury, 1999). People now have access to vast amounts of information; however, it is becoming increasingly difficult to locate useful information. Search engines usually return a large number of results in response to user queries. One study of European users showed that about 50 per cent of documents viewed by users are irrelevant (Jansen and Spink, 2005). Users

Online Information Review
Vol. 33 No. 4, 2009
pp. 696-719
© Emerald Group Publishing Limited
1468-4527
DOI 10.1108/14684520910985684

need to open several links to find the desired information, especially for specific and complex queries (e.g. best retirement countries) and for tasks such as background searching rather than queries with commonplace answers (e.g. capital city of Sweden).

In currently available search engines, such as Google and Altavista, each link in the results is associated with a short summary (e.g. a two-line extract) of its content. Although such extracts show some of the document fragments containing the query words, they fail to reveal their context within the document. As a result, the user either misses relevant results or spends time on irrelevant ones. Figure 1 shows the first six results of Google in response to the TREC-2004[1] query “antibiotics bacteria disease”. In that task, the aim of the user is to find documents that discuss how and why antibiotics become ineffective for some bacteria types. When we analyse the related documents, we see that only half of the extracts in the figure effectively direct the users.

At this point, automatic summarisation techniques gain importance. Although creating summaries as successful as human summaries is still a long-term research direction, summaries that are not perfect can be utilised to improve the effectiveness of other tasks such as information retrieval (Sparck-Jones, 1999). Automatic summarisation research has traditionally focused on creating general-purpose summaries. However, in an information retrieval paradigm, it has become important



antibiotics bacteria disease

Antibiotics
This **disease** affects the nervous system and causes muscle paralysis, ... Taking **antibiotics** irregularly allows **bacteria** to adapt and multiply, adding to the ...
www.pfizer.ca/english/your%20health/anti%20infective/default.asp?s=1 - 33k - Cached - Similar pages

Honey Remedy Could Save Limbs -
Attempts in the lab to induce a **bacterial** resistance to honey have failed, ... In the United States, however, honey as an **antibiotic** is nearly unknown. ...
www.wired.com/news/technology/medtech/0,71925-0.html - 41k - Cached - Similar pages

CDC Division of **Bacterial** and Mycotic **Diseases**: **Antibiotic** Resistance
Information about a national campaign to reduce antimicrobial resistance through promotion of more appropriate **antibiotic** use. Includes CDC recommendations ...
www.cdc.gov/drugresistance/community/ - 38k - Cached - Similar pages

DBMD Home Page | CDC **Bacterial**, Mycotic **Diseases**
The mission of the Division of **Bacterial** and Mycotic **Diseases** (DBMD) is to prevent illness ... Promoting Appropriate **Antibiotic** Use in the Community Website ...
www.cdc.gov/ncidod/dbmd/ - 36k - Cached - Similar pages
[More results from www.cdc.gov]

Antibiotics and **bacterial** infections
FishDoc: Fish **disease** treatments. Using **antibiotic** treatments for **bacterial** infections. ... **Antibiotics** and **bacterial diseases**. **bacteria** – what are they? ...
www.fishdoc.co.uk/treatments/**antibiotics**.htm - 22k - Cached - Similar pages

ASK THE VET (**Antibiotic** Considerations in the Treatment of ...
ASK THE VET (**Antibiotic** Considerations in the Treatment of **Bacterial Disease**). by Linda Pesek, DVM Westbury Animal Hospital 319 Union Avenue Westbury, NY ...
theaviary.com/s0196-12.shtml - 14k - Önbellek - Benzer sayfalar

Figure 1.
First few outputs of Google search engine for an example query

to bias towards user queries in order to be effective. Also, traditional approaches have usually considered a document as a flat sequence of sentences and ignored the inherent structure of documents during the summarisation process and in the output summaries. This aspect becomes especially important in the context of web documents, which typically show complex organisation of content, having sections and subsections with different topics and formatting.

In this paper, we propose a novel summarisation approach for web searching that utilises query-biased techniques and document structure both in the summarisation process and in the output summaries. Providing the context of searched terms in a way that preserves the structure of the document (i.e. the sectional hierarchy and heading structure) may help users to determine the relevance of the results better. To the best of our knowledge, the effects of the explicit document structure and query-biased techniques have not been investigated before in a web search context. We tested the system using a task-based evaluation method. The results show a significant improvement over both Google extracts and unstructured summaries of the same size.

Background and related work

Search engines and query types

The information retrieval discipline deals with the storage, retrieval and maintenance of information. The general objective of an information retrieval system is to minimise the time spent by a user in locating the needed information (Jackson and Moulinier, 2007; Kowalski and Maybury, 2002). Although a variant of classical information retrieval, information retrieval on the web shows significant differences when compared with traditional text retrieval systems. These differences mainly stem from a number of typical characteristics of the web such as its distributed architecture, the heterogeneity of the available information, and its size and growth rate (Chowdhury, 2006; Broder and Henzinger, 2002).

One of the major components of a web information retrieval system is the searching component (the search engine). A search engine allows the user to enter search terms (queries) that are run against a database and retrieves from its database web pages that match the search terms. We can identify several types of query forms supported by modern search engines (Chowdhury, 2006; Baeza-Yates and Ribeiro-Neto, 1999; Berry and Browne, 1999; Kowalski and Maybury, 2002). The basic and most widely used mechanism is Boolean search, where one or more keywords separated by (implicit or explicit) Boolean operators are entered. Phrase search is a variant of Boolean search in which the user requires a set of contiguous words to be found in the given order. A generalised form of this idea is proximity search in which a sequence of words or phrases is given together with a maximum distance allowed between them. Another mechanism is limiting the search with some restrictions on the search items or on the webpage properties. The most common types are range searching (specifying a range for a word) and field searching (restricting the search to the content of a field, such as the title, language or file type).

Some search engines support a number of more advanced query types (Berry and Browne, 1999; Kowalski and Maybury, 2002). Natural language search is a generalisation of Boolean search, where any reference to Boolean operators is eliminated and the user formulates the query as a question or a statement. The search algorithms underlying this model of searching need to be quite different from those of simple search models. In thesaurus search, the search terms supplied by the user are expanded to also include similar words or concepts. Finally, fuzzy search refers to the capability of the system to handle the misspellings and variations (e.g. the stemmed form) of the same words.

According to Ingwersen and Järvelin (2005), the information needs of users can be classified based on three dimensions: the intention or goal of the searcher, the kind of knowledge currently known by the searcher, and the quality of what is known. In the case of well-defined knowledge of the user, specific information sources are searched, whereas in ill-defined (muddled) cases the search process is rather exploratory. Based on a related work, a summarisation system can be evaluated based on four types of information need in web searching: search for a fact, search for a number of items, decision search, and background search (White *et al.*, 2003).

It is worth mentioning some statistical figures about search engines in order to better evaluate their capabilities. Researchers estimate that the content of the web is doubling each year. Search engines can index only a fraction of the web. A 2003 study reported that the largest search engine at that time (Google) indexed about 3.8 billion web pages, which corresponded to only 16 per cent of the web (Chowdhury, 2006). Due to the huge size of the web, response time is a critical factor for search engines. When a query is given by a user, a search engine normally scans all the pages indexed. However, in order not to have an adverse effect on the response time, usually a few of the first result screens are generated more quickly (e.g. by using some templates), exploiting the fact that most of the users do not pass beyond a few screens (Grossman and Frieder, 2004).

Web document analysis

In its general sense, document structure analysis aims at deconstructing a given document into its component regions and understanding their functional roles and relationships. Compared to the older field of printed document analysis (Niyogi and Srihari, 1995), the structural and semantic analysis of web documents is a relatively young field of research. Web documents are usually encoded in hypertext markup language (HTML) and can contain rich structural information. In web document analysis, document object model (DOM) trees of web documents are being utilised increasingly because they provide a more global view of the document structure (Chaudhuri, 2006). A DOM tree can be used to extract useful content from HTML documents by eliminating cluttered parts such as advertisements (Gupta *et al.*, 2005). In one study, HTML pages in a specific domain were converted into semantically-rich extensible markup language (XML) documents by restructuring the DOM trees of the documents (Chung *et al.*, 2002). There is some related work on extraction of the main title from web documents using DOM tree analysis and machine learning techniques (Xue *et al.*, 2007).

HTML documents can be deconstructed into coherent segments in a flat or hierarchical way. This has several applications including display of content in small-screen devices such as PDAs, and summarisation. In one work, HTML documents were divided into flat segments using machine-learning techniques based on DOM trees and formatting features (Feng *et al.*, 2005). In hierarchical structure analysis, the document is processed to obtain a hierarchy of segments and sub-segments. One approach to obtain the hierarchical structure is to group visually similar document parts (Yang and Zhang, 2001). Alternatively, the document can be partitioned iteratively into smaller blocks by detecting visual separators (Chen *et al.*, 2003). This analysis can also be performed by partitioning the document into semantic textual units and arranging the hierarchy based on emphasis differences between the units (Buyukkokten *et al.*, 2001). Other approaches to hierarchical structure identification include the application of a string matching algorithm on the DOM tree paths (Mukherjee *et al.*, 2003) and using presentation regularities (Vadrevu *et al.*, 2007). Our research differs from all these works in that we have concentrated on section headings and made use of these in building the hierarchy, whereas other works have not particularly concentrated on heading-based hierarchy.

Automatic summarisation and abstracting

The goal of automatic summarisation is the processing of a document and the presentation of its most important content to the user in a condensed form. There are basically two forms of summary: extracts and abstracts. An extract is a summary created by drawing sentences from the document itself, whereas an abstract is a summary at least some of whose material is not present in the document (Chowdhury, 2006; Mani, 2001). In view of this definition, abstraction is a much more difficult task than extraction, since it involves rephrasing a text and thus necessitates sophisticated mechanisms and knowledge sources. This is the reason that the vast majority of summarisation systems focus on extraction rather than abstraction.

Related to the type of user a summary is intended for, there are two basic approaches. Query-biased (user-focused) summaries are tailored to the information need of a particular user, whereas generic summaries aim at summarising a given document without focusing on a particular topic. Evaluation of a summary can take one of two forms. Intrinsic evaluation judges the quality of a summary directly based on some criteria such as informativeness, coverage and correctness. Extrinsic evaluation is related to the purpose of the summary and judges the quality of the summary based on how it affects the completion of some other task (Chowdhury, 2006; Mani, 2001; Moens, 2002).

The summarisation process consists of three main steps: analysis, transformation and synthesis. Normally, extraction involves only the analysis step, which is based on the idea of scoring sentences in the document with respect to some criteria and then selecting a number of high scoring sentences. There are four main methods used to measure the importance of a sentence (Chowdhury, 2006; Mani, 2001). The frequency method gives a score to a sentence depending on the number of occurrences of the sentence words in the document. The cue method favours sentences that contain one or more of the phrases in a predefined list of cue phrases. A sentence that includes a cue phrase (e.g. beginning with "In summary") is assumed to contain important information. In the title method, the scores of sentences that contain words that occur in the title of the document are increased. The location method gives precedence to sentences that appear in some particular sections of the document (e.g. the first sentence of the document), assuming that they are more indicative of the content than the others. All systems that prepare summaries by sentence extraction use some variants of these basic methods.

Most of the related works on automatic summarisation have aimed at creating generic summaries rather than query-biased ones, and have employed extraction methods based on sentence weighting (Guo and Stylios, 2005; Yeh *et al.*, 2005; Otterbacher *et al.*, 2006). In one work, the effects of several sentence-weighting schemes were investigated (Liang *et al.*, 2007). It was found that using a combination of weighting components improves the performance in comparison to any single component.

Query-biased summaries have been shown to be effective in information retrieval tasks (Tombros and Sanderson, 1998). WebDocSum is a retrieval interface providing query-biased summaries that are longer compared to the extracts provided by traditional search engines (White *et al.*, 2003). The summaries of the system are presented in a separate window to prevent a cluttered view of the search results. White *et al.* (2003) showed them to be more effective than the summaries of Google and AltaVista in a task-based evaluation. Both of these studies (Tombros and Sanderson, 1998; White *et al.*, 2003) were based on sentence extraction and basic statistical approaches such as title, location and query methods.

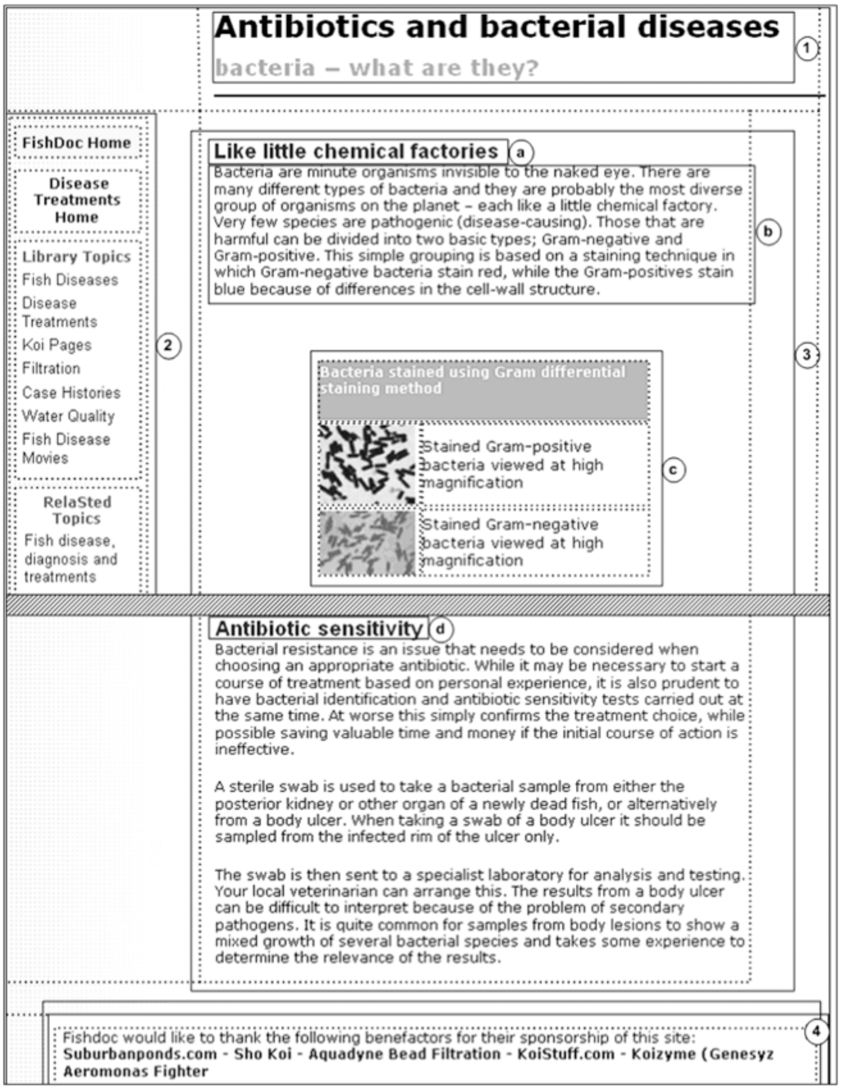
In the literature, there are discourse-level summarisation approaches based on content analysis (cohesion) or structural analysis (coherence). The former involves relationships between words and is related to how tightly the document is connected (Barzilay and Elhadad, 1999). The latter captures structural information by identifying macro-level relationships between sentences or clauses (Marcu, 1999). Based on several earlier psycholinguistic studies, Marcu (2000) stated that the structure of a text is essential in summarising the text. In another study, a structure-based and query-specific summarisation technique was proposed where the structure is obtained by connecting related document fragments (i.e. paragraphs) and then forming a document graph (Varadarajan and Hristidis, 2005).

There are few works on summarisation based on explicit document structure such as document sectional hierarchy. In one such study, the "table of content"-like structure of HTML documents was incorporated into generic summary outputs (Alam *et al.*, 2003). Yang and Wang (2008) developed the fractal summarisation method where generic summaries are created based on the hierarchical structure of a document, including chapters, sections and subsections. These studies focused on general-purpose summaries, not tailored to particular user queries or web search tasks. As far as we know, there has been no research on summarisation combining the explicit structure of web documents and query-biased techniques.

There exist some studies on summarisation of XML documents that are inherently structured. In one of these studies, query-biased summarisation was used as an aid for searching XML documents (Szlávik *et al.*, 2006). In another study, a machine learning approach was proposed for summarisation of XML documents based on structure and content (Amini *et al.*, 2007).

Research goals

Our main research goal was to create summaries that are more effective than the ones provided by traditional search engines. The targeted documents were general web documents with no domain restriction. Web documents are typically heterogeneous documents containing images, text in various formats, interactive forms, menus, etc. Their content may also be diversified with sections on different topics, advertisements, etc. A screenshot of some parts of a web document is given in Figure 2. The circles mark different parts of the document and will be used for reference.



Antibiotics and bacterial diseases

bacteria – what are they?

FishDoc Home

Disease Treatments Home

Library Topics
Fish Diseases
Disease Treatments
Koi Pages
Filtration
Case Histories
Water Quality
Fish Disease Movies

RelaSted Topics
Fish disease, diagnosis and treatments

Like little chemical factories

Bacteria are minute organisms invisible to the naked eye. There are many different types of bacteria and they are probably the most diverse group of organisms on the planet – each like a little chemical factory. Very few species are pathogenic (disease-causing). Those that are harmful can be divided into two basic types; Gram-negative and Gram-positive. This simple grouping is based on a staining technique in which Gram-negative bacteria stain red, while the Gram-positives stain blue because of differences in the cell-wall structure.

Bacteria stained using Gram differential staining method

Stained Gram-positive bacteria viewed at high magnification

Stained Gram-negative bacteria viewed at high magnification

Antibiotic sensitivity

Bacterial resistance is an issue that needs to be considered when choosing an appropriate antibiotic. While it may be necessary to start a course of treatment based on personal experience, it is also prudent to have bacterial identification and antibiotic sensitivity tests carried out at the same time. At worse this simply confirms the treatment choice, while possible saving valuable time and money if the initial course of action is ineffective.

A sterile swab is used to take a bacterial sample from either the posterior kidney or other organ of a newly dead fish, or alternatively from a body ulcer. When taking a swab of a body ulcer it should be sampled from the infected rim of the ulcer only.

The swab is then sent to a specialist laboratory for analysis and testing. Your local veterinarian can arrange this. The results from a body ulcer can be difficult to interpret because of the problem of secondary pathogens. It is quite common for samples from body lesions to show a mixed growth of several bacterial species and takes some experience to determine the relevance of the results.

Fishdoc would like to thank the following benefactors for their sponsorship of this site:
Suburbanponds.com - Sho Koi - Aquadyne Bead Filtration - KoiStuff.com - Koizyme (Genesyz Aeromonas Fighter

Figure 2.
Screenshot of an example web document

One issue related to the research goal is the structural and semantic analysis of web documents. This is a challenging task since web documents are generally prepared for visual access. In this paper, we address the particular problem of finding the sectional hierarchy of an HTML document, where a document can be considered as consisting of sections and sub-sections with corresponding headings and sub-headings. The document in Figure 2 corresponds to the fifth result in Figure 1. By looking at the Google snippet, it is hard to determine whether it is a relevant document. However, if the context of the searched terms was made explicit by using some structural clues and hierarchical information, we expect that users could more easily make a decision. This involves issues like identifying the headings and distinguishing the main content from secondary parts such as menus.

Another issue we consider is the use of this structural and semantic information during the summarisation process and in the output summaries. This involves the identification of the importance of each section and subsection. Important sections and corresponding headings should be more heavily represented in the summary. This is contrary to the traditional approaches where a document is considered as a flat sequence of sentences.

Methodology

Structural processing

In the proposed system, the aim of structural processing is to identify the sectional hierarchies of web documents, that is, sections and sub-sections together with the corresponding headings and sub-headings. This information will in turn be used during the summarisation phase. The proposed structural processing method involves three steps, which are detailed below.

DOM tree processing. A DOM tree is a hierarchical representation of an HTML document. However, it concerns the presentation of the document and usually does not correspond to a semantic hierarchy. Nevertheless, this representation may be partly utilised to obtain the semantic organisation of a document. In Figure 3, a part of the DOM tree corresponding to the document in Figure 2 is shown.

In HTML syntax, there are two main types of tag: "container tags" (e.g. < table > , < td > , < tr > , etc.) that can contain other HTML tags or text, and "format tags" (e.g. < b > , < font > , < h1 > , etc.) that are usually concerned with the formatting of text (Raggett *et al.*, 1999). In general, the organisation of content is achieved by the use of container tags. The DOM tree usually has a complex organisation with nested container and format tags. The depth of the tree may be quite large and it usually contains tags that do not correspond to textual content. A hierarchy that is closer to the sectional hierarchy (usually with much less depth) should be distilled from this tree.

The theory behind the DOM tree-processing step is that semantically related parts of an HTML document usually occur in the same or neighbouring container tags in the hierarchy. The approach we take is to convert the DOM tree into a simplified version with only containment relationships. The tree is restructured so that each leaf corresponds to exactly one sentence whereas in the original DOM tree, each leaf may correspond to part of a sentence or more than one sentence. The process involves the following steps and the output for the example DOM tree is shown in Figure 4.

(1) Prune nodes that do not contain text in the leaves beneath them and nodes that will not be used in summarisation, such as forms or drop-down menus.

(2) Split or merge leaf nodes so that each leaf node corresponds to exactly one sentence.

(3) Simplify the tree to obtain the containment hierarchy. The algorithm (see Figure 5) works in a breadth-first fashion starting from the root. Nodes that have only one child or nodes with format tags (such as < font >) are removed. For this purpose, their children are percolated up and the format tags are passed as features of them. In the resulting tree, document parts are grouped under block elements.

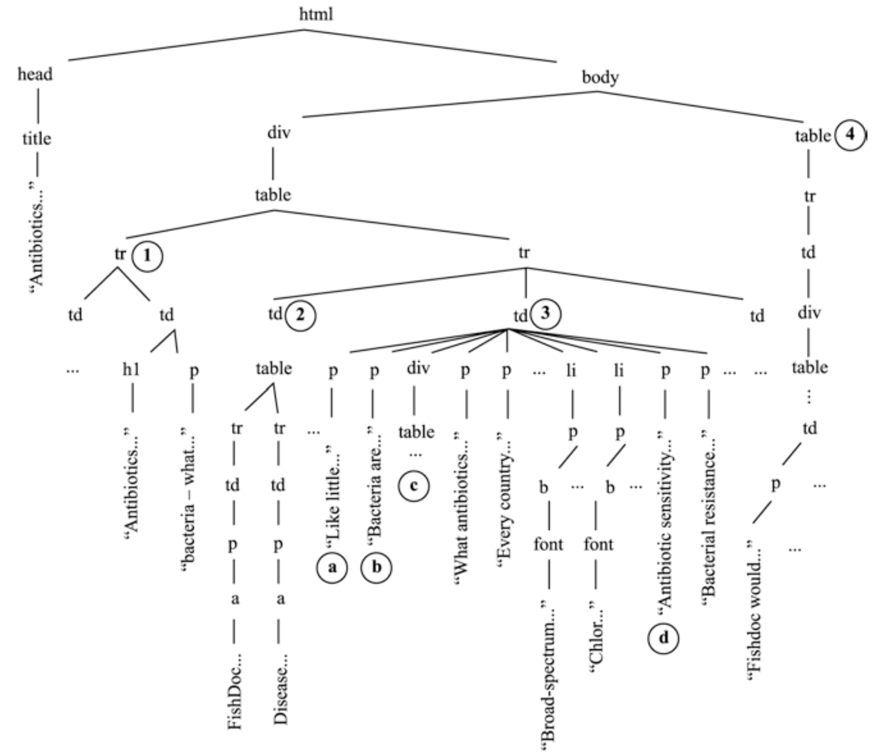


Figure 3.
Part of the DOM tree for the example web document

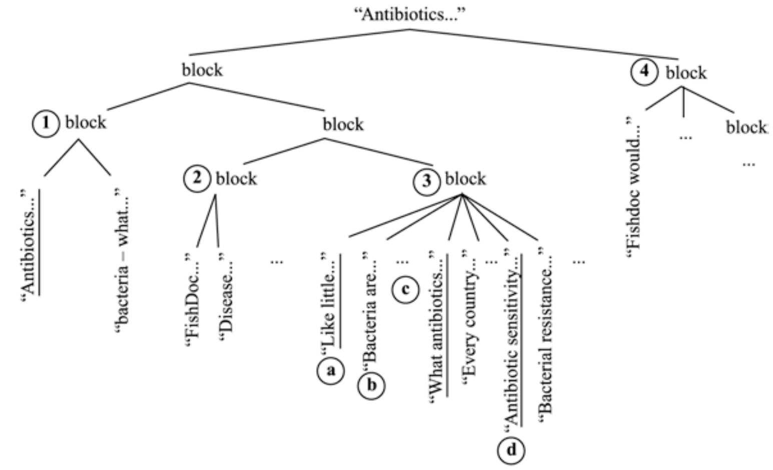
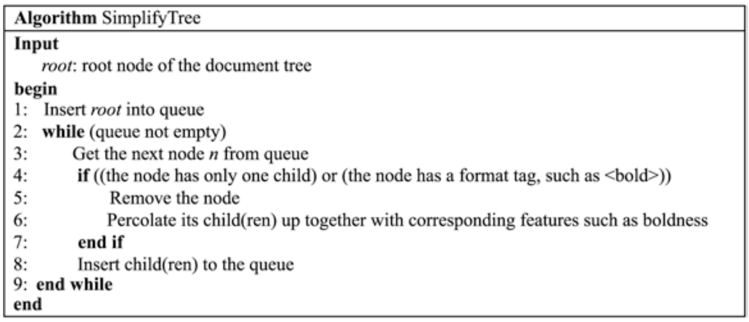


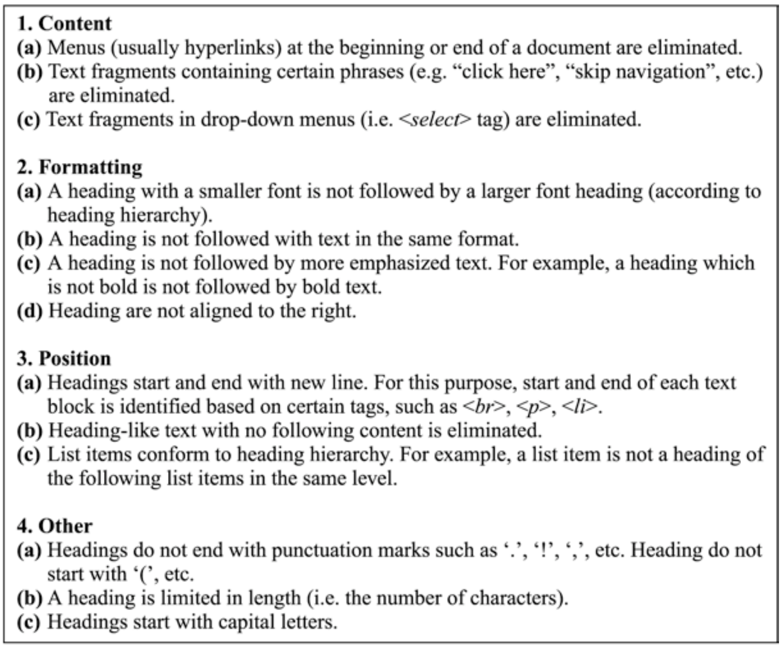
Figure 4.
The document tree obtained after DOM tree processing

Heading identification. The aim of this step is to automatically identify all the headings and subheadings in a given HTML document. Actually, there are heading tags (< h1 > through < h6 >) in HTML to format different levels of heading. However, these tags are rarely used for this purpose. Sometimes, they are even used for formatting non-heading text. We examined several web pages (English and Turkish) in order to find out the characteristics that distinguish headings from non-heading text. In most of the documents, the headings were formed by formatting them differently from their surrounding text. Thus, their content and the context in which they occurred should be examined. We took a heuristic-based approach to distinguish headings from non-heading text. The heuristics are listed in Figure 6. The headings obtained for the example document are shown underlined in Figure 4.



```
Algorithm SimplifyTree
Input
    root: root node of the document tree
begin
  Insert root into queue
  while (queue not empty)
      Get the next node n from queue
      if ((the node has only one child) or (the node has a format tag, such as <bold>))
          Remove the node
          Percolate its child(ren) up together with corresponding features such as boldness
      end if
      Insert child(ren) to the queue
  end while
end
```

Figure 5.
Document tree simplification algorithm



1. Content

(a) Menus (usually hyperlinks) at the beginning or end of a document are eliminated.

(b) Text fragments containing certain phrases (e.g. “click here”, “skip navigation”, etc.) are eliminated.

(c) Text fragments in drop-down menus (i.e. <*select*> tag) are eliminated.

2. Formatting

(a) A heading with a smaller font is not followed by a larger font heading (according to heading hierarchy).

(b) A heading is not followed with text in the same format.

(c) A heading is not followed by more emphasized text. For example, a heading which is not bold is not followed by bold text.

(d) Heading are not aligned to the right.

3. Position

(a) Headings start and end with new line. For this purpose, start and end of each text block is identified based on certain tags, such as <*br*>, <*p*>, <*li*>.

(b) Heading-like text with no following content is eliminated.

(c) List items conform to heading hierarchy. For example, a list item is not a heading of the following list items in the same level.

4. Other

(a) Headings do not end with punctuation marks such as ‘.’, ‘!’, ‘,’, etc. Heading do not start with ‘(’, etc.

(b) A heading is limited in length (i.e. the number of characters).

(c) Headings start with capital letters.

Figure 6.
Heuristics used for heading identification

Hierarchy restructuring. The last step is to restructure the tree by making use of the heading information already discovered. In the DOM tree processing step, each sentence (including headings) was identified accompanied by formatting features given in Table I. These feature-value pairs are used for differentiating between different levels of headings in the hierarchy. The idea is that in a semantic block of text, headings in the same level usually have the same features and the sectional hierarchy is achieved with distinct formatting for different levels of heading as in the following.

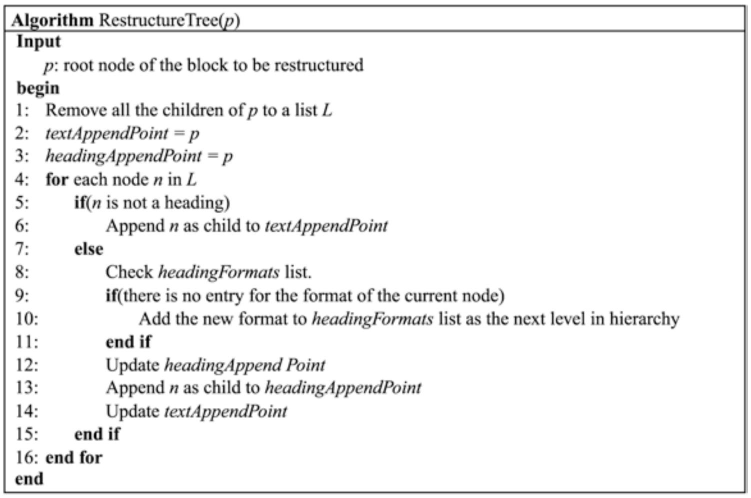
Antibiotics and bacterial diseases
HTML code: … < h1 > Antibiotics and bacterial diseases < /h1 > …
Features: {h1 = true}

FREQUENTLY ASKED QUESTIONS
HTML code: … < b > < font size = ”2” > FREQUENTLY ASKED QUESTIONS < /font > < /b > …
Features: {B = true, f_size = 2, allUpperCase = true}

The strategy we employ works bottom-up in the document tree and first smaller blocks of text (deeper in the hierarchy) are restructured. The algorithm to restructure a given block within the document tree is given in Figure 7. Given a particular node in the document tree, its children are considered one by one. If the considered child is not a heading, it is appended under the last heading node encountered. Otherwise, first it is checked whether it belongs to a heading level previously used in the block. If not, it is added as a new heading level. Then, it is appended to the appropriate position in the hierarchy. Figure 8 shows the tree obtained after the application of the restructuring step on the tree of Figure 4. In this tree, the root covers the entire document. The

Table I.
Features used for identifying the format of the text

Feature	Description	Data type
h1	< h1 > , level-1 heading	Boolean
h2	< h2 > , level-2 heading	Boolean
h3	< h3 > , level-3 heading	Boolean
h4	< h4 > , level-4 heading	Boolean
h5	< h5 > , level-5 heading	Boolean
h6	< h6 > , level-6 heading	Boolean
B	< b > , bold	Boolean
strong	< strong > , strong emphasis	Boolean
em	< em > , emphasis	Boolean
A	< a > , hyperlink	Boolean
U	< u > , underlined	Boolean
I	< i > , italic	Boolean
f_size	< font size = … > , font size	Integer
f_colour	< font colour = … > , font colour	String
f_face	< font face = … > , font face	String
allUpperCase	all the letters of the words are uppercase	Boolean
cssId	CSS id attribute if used	String
cssClass	CSS class attribute if used	String
alignment	align attribute	String
li	< li > , different levels of list elements	Integer



```
Algorithm RestructureTree(p)
Input
    p: root node of the block to be restructured
begin
    Remove all the children of p to a list L
    textAppendPoint = p
    headingAppendPoint = p
    for each node n in L
        if(n is not a heading)
            Append n as child to textAppendPoint
        else
            Check headingFormats list.
            if(there is no entry for the format of the current node)
                Add the new format to headingFormats list as the next level in hierarchy
            end if
            Update headingAppend Point
            Append n as child to headingAppendPoint
            Update textAppendPoint
        end if
    end for
end
```

Figure 7.
Hierarchy restructuring algorithm

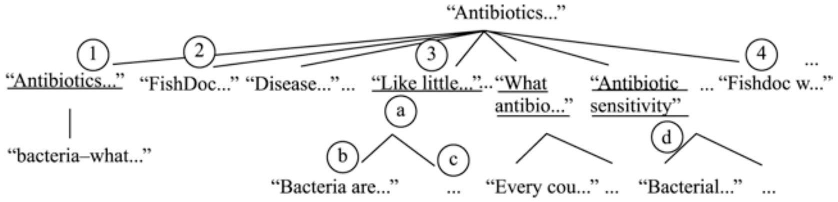


Figure 8.
Part of the document tree after restructuring

intermediate nodes contain section and sub-section headings and the leaves contain the underlying sentences.

Summary extraction

In the proposed system, we create indicative summaries using the method of sentence extraction. The structure of the original document and the context of the selected sentences are preserved and thus the user can judge the relevance of documents more precisely. The summarisation algorithm is run after the structural processing phase and utilises the structural properties of documents both during the summarisation process and in the output summaries. A query-biased approach is employed that is suitable for web searching. We generate the summaries using two levels of scoring as explained below.

Sentence scoring. For scoring the sentences, we employ a form of the basic sentence scoring metrics described earlier. Since we aim at preparing query-biased summaries in this work, we add a query component to the scoring mechanism. We use four types of scoring method. The heading method and the location method have been adapted in such a way that they utilise the output of the structural processing step. Whenever appropriate, stop words are eliminated and stemming is applied during the processes explained below.

- *Heading method.* Headings in a document usually include keywords related to the content and sentences containing heading words are likely to carry more information. Previous studies have taken only the main title of the documents into account or operated on predefined headings. In contrast, the algorithm in this research makes use of all the headings identified during structural processing. A heading score is assigned to each sentence as the number of heading words it contains.
- *Location method.* This method is based on the idea that sentences located at certain positions usually convey salient information. We have modified this approach to incorporate sectional information so that sentences that are the first sentence of a section or subsection are given a positive score.
- *Term frequency method.* Terms occurring frequently within a document usually convey important information and sentences with a large number of such words should be preferred. In the proposed system, each sentence is assigned a term frequency score by summing the frequencies of the constituting words.
- *Query method.* In the web search context, biasing summaries with respect to the user query is important (Tombros and Sanderson, 1998). When sentences containing query words are more heavily represented in the summaries, the users are expected to be able to judge the relevance of search results better. In the proposed system, each sentence is given a query score as the number of query words it includes.

The scores for each method are normalised by dividing them to the maximum score for that method in a given document. The overall sentence score is calculated as follows:

$$s_{\text{sentence}} = w_1 \times s_{\text{heading}} + w_2 \times s_{\text{location}} + w_3 \times s_{\text{tf}} + w_4 \times s_{\text{query}}$$

where s represents a normalised score and w corresponds to the related weight. We have experimented with different weight values and observed that giving similar weights to heading, location and term frequency metrics, and assigning a weight to the query metric a few times more than the others give rise to the best performance. This indicates that query terms provide the most important information, but the other sources of information also have an effect and should not be disregarded. Table II shows the score calculation for the following example sentence from the document in Figure 2 ($w_1 = w_2 = w_3 = 1$ and $w_4 = 3$).

Table II.
Application of scoring methods on an example sentence (numbers inside parentheses denote frequencies within the document)

Method	Applicable terms	Sentence score	Maximum score in document	Normalised sentence score
Heading	bacteria, bacteri, diseas	5	6	0.83
Location	–	0	1	0.00
Term frequency	bacteria(17), involv(1), bacteri(11), diseas(7), ulcer(5), fin(1), rot(1), acut(1), septicaemia(1), gill(1)	64	261	0.25
Query	bacteria, diseas	2	2	1.00
Overall				4.08

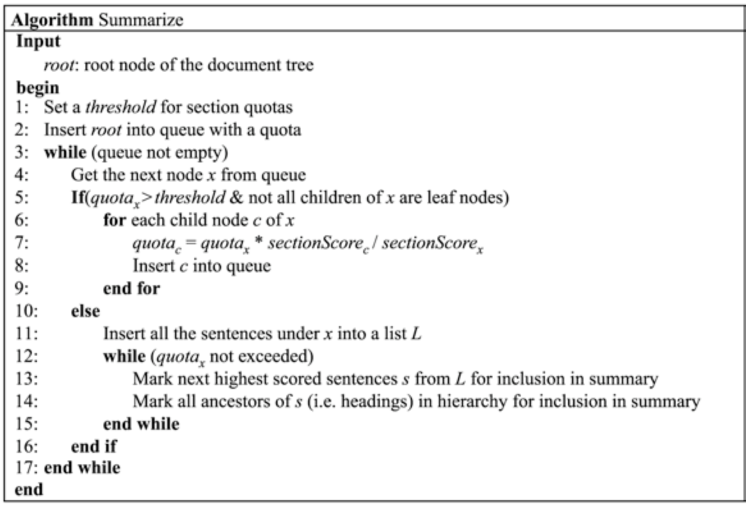
Query: antibiotics bacteria disease

Sentence: These are the bacteria that are usually involved with bacterial disease such as ulcers, fin rot, acute septicaemia and bacterial gill disease.

Section scoring. Traditional approaches to summarisation usually generate summaries by considering the document as a linear sequence of sentences. Some of them score the sentences by taking some structural information (e.g. heading and location information) into account. However, during the extraction phase, they still select the sentences without considering the sectional hierarchy of the document. In this paper, we follow a different strategy. We argue that sections inside a document have varying degrees of importance with respect to the user query. Hence, they should be represented to different extents in the summary. Moreover, rather than flat text summaries, we form structured summaries that include the context of sentences in terms of section headings.

In the proposed system, each section is given a section score as a measure of its importance. This score is computed as the sum of the sentence scores in that section and determines the number of sentences that section will be represented by in the summary (i.e. its quota). The summarisation algorithm (Figure 9) operates on the document tree obtained in the structural processing stage. The document root is initially given a quota corresponding to the maximum summary size (e.g. 25 sentences). During processing, the quota of a node (section) is shared among its children (sub-sections). A predetermined threshold (e.g. three sentences) is used to end this division for small values of quota. When the threshold is reached or when a section has no more subsections, the highest scored sentences within the section are selected one by one, together with the headings above them (corresponding to ancestor nodes) until the quota for that section is reached.

After the summaries are formed, the search results are displayed on the screen in a similar way to the search engines. The difference from traditional search engines is that when the user moves the mouse on a result, the corresponding summary is shown



```
Algorithm Summarize
Input
    root: root node of the document tree
begin
    Set a threshold for section quotas
    Insert root into queue with a quota
    while (queue not empty)
        Get the next node x from queue
        If($quota_x$ > threshold & not all children of x are leaf nodes)
            for each child node c of x
                $quota_c = quota_x * sectionScore_c / sectionScore_x$
                Insert c into queue
            end for
        else
            Insert all the sentences under x into a list L
            while ($quota_x$ not exceeded)
                Mark next highest scored sentences s from L for inclusion in summary
                Mark all ancestors of s (i.e. headings) in hierarchy for inclusion in summary
            end while
        end if
    end while
end
```

Figure 9.
Summarisation algorithm

in a separate window. As the user moves from one link to another, the previous summary window is cancelled and the new summary is displayed. In this way, it becomes possible to display several results on a page with a detailed view of each result.

Data collection and analysis

The system was developed using the open source GATE framework for text engineering (Maynard *et al.*, 2002) as the underlying development environment. GATE supports some common functionalities like the tokeniser and sentence splitter, and includes the Porter stemmer (Porter, 1980) as a plug-in. We implemented the DOM tree extractor, the HTML document structure analyser and the summarisation engine as new modules of this framework.

Analyses in the literature about user behaviour in forming queries have shown that users do not put much effort into formulating a query and they mostly use very simple Boolean types of queries (Ingwersen and Järvelin, 2005; Markey, 2007). It has been reported that 80 per cent of queries are formed as a sequence of words without any Boolean operator in between and that the average query length is 2.35 words (Broder and Henzinger, 2002). Another study has given the same average length and has estimated the average number of operators as 0.41 operators per query (Baeza-Yates and Ribeiro-Neto, 1999). In addition, it was found that 25 per cent of users use a single keyword. Chowdhury (2006) stated that only about 8 per cent of queries contain Boolean operators and only 9 per cent use some advanced features. All these results indicate the poor nature of end-user searches. In this research we followed the same approach in simulating user behaviour by using Boolean queries with a length of two to three words.

We created two different document collections for the experiments. The first included English web documents collected from the results of Google in response to ten different queries from the TREC-2004 Robust Test Set[1] (see Table III). For each query, a set of ten documents were randomly collected from the top 50 results returned by Google, corresponding to a total of 100 out of 500 documents. The second collection included Turkish web documents collected from the results of Google using TREC-like queries defined for Turkish (Can *et al.*, 2008) (see Table III). The collection included 50 documents randomly collected out of 250 documents. The experiments were performed

Table III.
Queries used in the experiments

English queries	Turkish queries
Q1. Hubble telescope achievements	Q1. *Tsunami* (Tsunami)
Q2. Best retirement country	
Q3. Literary/journalistic plagiarism	Q2. *Ekonomik kriz* (Economic crisis)
Q4. Mexican air pollution	
Q5. Antibiotics bacteria disease	Q3. *Türkiye'de meydana gelen depremler* (Earthquakes in Turkey)
Q6. Abuses of e-mail	
Q7. Declining birth rates	Q4. *Sanat ödülleri* (Art awards)
Q8. Human genetic code	
Q9. Mental illness drugs	Q5. *Bilisim egitimi ve projeleri* (IT education and projects)
Q10. Literacy rates Africa	

on these randomly collected documents. The average document length was about 1,700 words in the English collection and 900 words in the Turkish collection. The proposed system was evaluated using two different experimental settings as detailed in the following subsections.

Document hierarchy identification experiment

The structural processing algorithm was run on both document collections. The examination of the collections revealed that they included different levels of structured documents, ranging from flat documents to highly structured ones, with an average sectional hierarchy depth of around four. Figure 10 shows the distribution of hierarchy depths in both collections.

We compared the results of the proposed system with both the actual hierarchies (extracted manually) and a baseline (a hierarchy formed using only heading tags < h1 > through < h6 > in HTML). The accuracy of the system was measured using two criteria: accuracy of hierarchy identification and accuracy of heading identification. Hierarchy accuracy was computed as the ratio of the number of parent-child relationships (heading-sub-heading and heading-text relationships) correctly identified to the total number of parent-child relationships in the manually identified hierarchy. Table IV shows the average results obtained for the English and Turkish collections.

The average accuracy obtained in the proposed system was around 70 per cent for both collections, whereas it was 50 per cent in the baseline method for the English collection. The baseline method failed for the Turkish collection because the particular < h > tags were not used in any of the documents. Instead, the sectional hierarchy was achieved using other features on the DOM tree such as format tags. In order to test whether the methods work on Turkish documents in general, we performed an

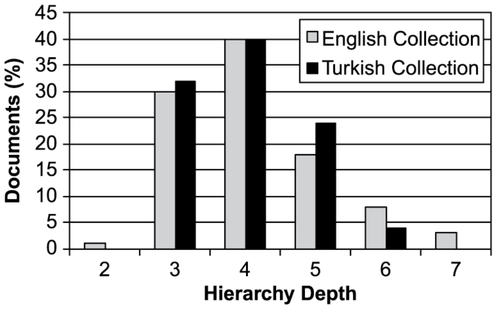


Figure 10.
Distribution of hierarchy depths in the document collections

	Hierarchy accuracy		Heading accuracy				
Collection	Baseline (only *h* tags)	Proposed system	Actual number	Proposed system recall	Proposed system precision	Proposed system F-measure	Baseline recall
English	0.50	0.71	8.82	0.88	0.64	0.71	0.43
Turkish	–	0.70	5.40	0.79	0.57	0.65	–

Table IV.
Results of the hierarchy identification experiment

additional analysis on documents of a Turkish university web site (50 documents on "boun.edu.tr" domain) and obtained 71 per cent accuracy using the proposed approach, which proves the robustness of the algorithm for Turkish web pages. The results indicate that a significant improvement in accuracy is possible via a heuristics-based document analysis.

The second criterion we employed was the accuracy of heading identification, where high recall rates were obtained for both collections. The precision rates were relatively lower because some non-heading texts were identified as headings due to the cluttered structure encountered in most web documents. Another analysis revealed that for the majority of the documents acceptable accuracy rates were obtained. For the English collection, about half of the documents had hierarchy accuracy of between 80 per cent and 100 per cent in the proposed system, while only about 25 per cent of the documents achieved this rate in the baseline method.

Summarisation experiment

The evaluation method used was extrinsic (task-based) evaluation where the summaries were judged according to their usefulness in a search engine. We preferred extrinsic evaluation as it is more suitable for cases where the summariser is embedded within another system (Mani, 2001), which was the case in this research. Four types of summary (document surrogate) were used for comparison:

(1) Google – query-biased extract provided by Google.

(2) Unstructured – query-biased summary without the use of structural information.

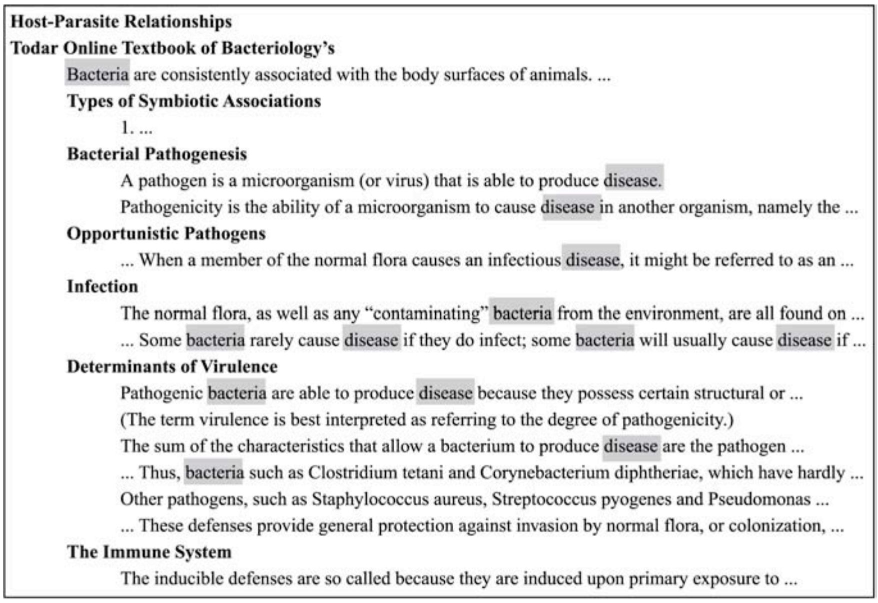
(3) Structured1 – structure-preserving and query-biased summary created by the proposed system using output of the hierarchy identification step.

(4) Structured2 – structure-preserving and query-biased summary created by the proposed system using manually identified structure.

All the summaries except the Google extracts were long summaries of the same size (about 25 sentences) to make them comparable with each other. In the case of unstructured summaries, the information obtained in the structural processing step (i.e. heading and location methods) was not used. An example output of the proposed system (Structured1) for the query "antibiotics bacteria disease" is given in Figure 11. The summaries are displayed in a hierarchical way in accordance with the sectional hierarchy. Also, headings and subheadings are given as bold and query keywords are highlighted. Each summary sentence is output as a single line; for this purpose, the end parts of longer sentences are replaced by "...", allowing a maximum of about 100 characters for each sentence.

For the performance measure, we used the relevance prediction approach proposed by Hobson *et al.* (2007) instead of relying on a gold standard. The relevance prediction approach compares the subject's relevance judgement of a summary with his or her own judgement of the original full-text document. In this way, we expected to reduce the effect of subject differences and obtain more reliable information.

The experiment methodology employed was a within-subjects (i.e. repeated measures) design to minimise the effects of differences among subjects (Montgomery, 2001). A total of ten subjects were used. The experiment was conducted using the



Host-Parasite Relationships

Todar Online Textbook of Bacteriology's

Bacteria are consistently associated with the body surfaces of animals. ...

Types of Symbiotic Associations

1. ...

Bacterial Pathogenesis

A pathogen is a microorganism (or virus) that is able to produce disease.

Pathogenicity is the ability of a microorganism to cause disease in another organism, namely the ...

Opportunistic Pathogens

... When a member of the normal flora causes an infectious disease, it might be referred to as an ...

Infection

The normal flora, as well as any "contaminating" bacteria from the environment, are all found on ...

... Some bacteria rarely cause disease if they do infect; some bacteria will usually cause disease if ...

Determinants of Virulence

Pathogenic bacteria are able to produce disease because they possess certain structural or ...

(The term virulence is best interpreted as referring to the degree of pathogenicity.)

The sum of the characteristics that allow a bacterium to produce disease are the pathogen ...

... Thus, bacteria such as Clostridium tetani and Corynebacterium diphtheriae, which have hardly ...

Other pathogens, such as Staphylococcus aureus, Streptococcus pyogenes and Pseudomonas ...

... These defenses provide general protection against invasion by normal flora, or colonization, ...

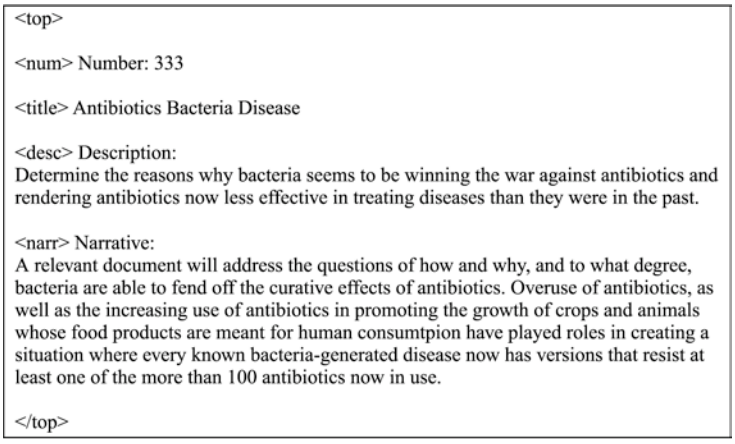
The Immune System

The inducible defenses are so called because they are induced upon primary exposure to ...

Figure 11.
An example summary output of the proposed system

queries in Table III for the English and Turkish document collections. The queries we used conform to the query formation behaviour of search engine users mentioned earlier. An example query, "antibiotics bacteria disease", is shown in Figure 12, in which each query is represented by a title (query terms), a description and a narrative part. The description states what is intended by the query terms and the narrative part provides a guide for deciding the relevance of a document. The queries cover different types of information need tasks, including search for a number of items (nine queries), decision search (one query) and background search (five queries).

For each document in the collections, five texts (the four summary types and the original document) were presented to the subject using a web-based interface. The



```
<top>

<num> Number: 333

<title> Antibiotics Bacteria Disease

<desc> Description:
Determine the reasons why bacteria seems to be winning the war against antibiotics and
rendering antibiotics now less effective in treating diseases than they were in the past.

<narr> Narrative:
A relevant document will address the questions of how and why, and to what degree,
bacteria are able to fend off the curative effects of antibiotics. Overuse of antibiotics, as
well as the increasing use of antibiotics in promoting the growth of crops and animals
whose food products are meant for human consumtpion have played roles in creating a
situation where every known bacteria-generated disease now has versions that resist at
least one of the more than 100 antibiotics now in use.

</top>
```

Figure 12.
An example TREC query used in the experiment

subject was required to complete all the documents of a query before continuing with the next query. The texts corresponding to a query were presented to the subject in random order to reduce carry-over effects, under the restriction that an original document was displayed after all four summaries of that document. For each text, the subject was asked to determine whether the corresponding document was relevant or irrelevant with respect to the query. In this way, each query and the corresponding document summaries were evaluated by three different subjects.

For each document summary, four types of results were identified through comparing the relevance judgements of the subjects for the summary and the original document: true positive (TP), false positive (FP), false negative (FN) and true negative (TN). Based on these values, accuracy (A), recall (R), precision (P) and f-measure (F) values for each summarisation method were calculated. The results are displayed in Table V together with the average time of making judgements for each method. Table VI shows the relative performance improvement provided by the proposed method ("Structured1") over the "Google" and "Unstructured" methods. The statistical tests (repeated measures ANOVA) verified that the "Structured1" method yielded significantly better results than both the "Google" and "Unstructured" summaries with $p < 0.05$ for the *f*-measure.

Discussion

The importance of summarisation in information retrieval tasks has been recognised in several studies related to human cognition. One of these studies compared the response time and accuracy of relevance assessment for original documents and their summaries, and showed that the time decreases more or less linearly with the length while accuracy decreases only logarithmically (Jackson and Moulinier, 2007). This

Table V. Results of the summarisation experiment

Collection	System	TP	FP	FN	TN	A	P	R	F	Time (s)
English	Google	107	38	60	95	0.67	0.73	0.62	0.63	14.58
	Structured1	137	25	30	108	0.82	0.85	0.80	0.80	27.60
	Structured2	138	23	29	110	0.83	0.85	0.83	0.82	28.58
	Unstructured	131	28	36	105	0.79	0.82	0.76	0.77	27.24
Turkish	Google	45	20	10	75	0.80	0.69	0.82	0.75	11.04
	Structured1	49	8	6	87	0.91	0.86	0.89	0.88	19.96
	Structured2	47	10	8	85	0.88	0.82	0.85	0.84	19.71
	Unstructured	43	13	12	82	0.83	0.77	0.78	0.77	19.96

Table VI. Improvement of the proposed method (Structured1) over other methods

Collection	System	P	R	F
English	Google	+16.44%	+29.03%	+26.98%
	Unstructured	+3.66%	+5.26%	+3.90%
Turkish	Google	+20.8%	+19.7%	+20.3%
	Unstructured	+19.2%	+26.4%	+23.6%

implies that we can gain time substantially without a significant loss in accuracy when using summaries rather than original documents. This hypothesis was also supported by Marcu (2000), who reported an experiment on an information retrieval task. The time taken when summaries were used was found to be about 80 per cent of the time required to perform the same task using the original documents, with recall and precision remaining approximately the same.

Current search engines use short extracts to display the results to the user. Such extracts focus only on the query words and thus miss the parts of the documents actually intended by the user. We have shown that a significant improvement in performance is possible by using summaries much longer than the extracts generated by traditional search engines (e.g. Google). Longer summaries can show the parts of a document that are relevant to the user query explicitly even if those parts do not contain any of the query terms. We have also shown the importance of maintaining document structure in the summaries. In our research, structured summaries increased the performance of the system significantly when compared with unstructured summaries of the same size. A structured summary provides an overview of the document and makes it much easier for the user to focus on the relevant parts, which can be considered as some sort of semantic information for the user.

With current search engines, a major limitation for the user is the size of the display screen, which constrains the number of results and the size of the extracts. To optimise the number of results reviewed per screen, most search engines display a few lines from the document that include the query terms. However, this does not seem to be suitable and it is argued that human cognition does not conform to this type of display (Van Oostendorp and De Mul, 1996). In this work, we combined the objective of displaying as many results as possible on a page with the objective of giving a detailed view for each result by activating a dynamic summary window.

The high success rates of the "Structured1" and "Structured2" methods indicate that combining structural processing with summary extraction is an effective approach. On average, the size of a summary prepared with these methods is about 16 per cent of the corresponding document. By looking at a summary of this size, users are able to determine the relevance with 80 to 90 per cent accuracy. When we compare these two methods with each other, we see that they give similar success rates. "Structured2" may be expected to yield a better performance since it is based on manually identified hierarchical structures of documents. This was indeed the case for the English data collection. However, interestingly, "Structured1" showed a slightly better performance than "Structured2" in the Turkish collection. We conjecture that this result was due to the two-stage nature of the process in the sense that the summarisation component can work on an imperfect structure and humans are good at coping with vagueness. Based on the high performance of the proposed method, "Structured1", we can conclude that it is a fully automatic method that can be incorporated into a search engine.

An analysis of user response times (Table V) shows that although the summaries were six to seven times longer than the Google extracts, they caused only a two-fold increase in response time. This indicates that people just look at the related parts of the summaries and then arrive at a decision. Thus we see that the proposed system has acceptable user response times despite the much longer summary size. An analysis of the time complexity of a search engine built on the proposed techniques shows that it is linear in document length. The document structural processing step is independent of the user query and needs to be performed once for each document. This process can be done offline, similar to the indexing phase of search engines. The summary extraction step has a linear time complexity. Given a document hierarchy with n nodes (sentences), the extraction algorithm (Figure 9) operates on most n nodes in a top-down fashion. At each node, either the quotas of the children nodes are calculated or the sentences are selected based on the quota of the node, both of which require constant time. As a result, the time complexity of the whole process is O(n).

Conclusions

In this paper, we have proposed a novel approach that can enrich the outputs of current search engines and that results in a substantial increase in performance. The method is composed of two steps: structural processing and summary extraction. The first investigates the problem of sectional hierarchy identification for web documents. We obtained about 70 per cent accuracy, which was shown in the second step to be an acceptable performance for the primary aim of this study. The second step is the summarisation phase. The performance of the proposed approach was found to be 80 to 90 per cent. This indicates that structure-preserving and query-biased summaries greatly influence the success of web search tasks. The approach was also applied to Turkish documents, which makes this the first automatic summarisation study of Turkish targeting web search.

In addition to the domain of search engines, the proposed approach can also be utilised in several other fields related to document processing. Information systems where large amounts of document need to be analysed, such as library systems, law and medicine, are typical candidates. The proposed methods allow browsing of large documents with the help of the structural information. Also, the summaries provided by the system can be taken as an outline in creating manual summaries.

Future work will address improving the success of the structural processing step and the summarisation engine. We are planning to incorporate machine learning techniques into the hierarchy identification algorithms. Also, the summarisation approach will be refined using linguistic information in syntactic and semantic levels.

Note

[1] TREC 2004 Robust Test Set includes a set of *ad hoc* information retrieval tasks compiled for the Text Retrieval Conference 2004. The set is available at: http://trec.nist.gov/

References

Alam, H., Kumar, A., Nakamura, M., Rahman, F., Tarnikova, Y. and Wilcox, C. (2003), "Structured and unstructured document summarisation: design of a commercial summariser using lexical chains", *Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh*, pp. 1147-1152.

Amini, M.R., Tombros, A., Usunier, N. and Lalmas, M. (2007), "Learning based summarisation of XML documents", *Journal of Information Retrieval*, Vol. 10 No. 3, pp. 233-55.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999), *Modern Information Retrieval*, Addison-Wesley, New York, NY.

Barzilay, R. and Elhadad, M. (1999), "Using lexical chains for text summarisation", in Mani, I. and Maybury, M.T. (Eds), *Advances in Automatic Text Summarisation*, MIT Press, Cambridge, MA, pp. 111-21.

Berry, M.W. and Browne, M. (1999), *Understanding Search Engines: Mathematical Modeling and Text Retrieval*, SIAM, Philadelphia, PA.

Broder, A. and Henzinger, M. (2002), "Algorithmic aspects of information retrieval on the web", in Abello, J., Pardalos, P.M. and Resende, M.G.C. (Eds), *Handbook of Massive Data Sets*, Kluwer Academic, Dordrecht.

Buyukkokten, O., Garcia-Molina, H. and Paepcke, A. (2001), "Accordion summarisation for end-game browsing on PDAs and cellular phones", Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Seattle, WA, pp. 213-220.

Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H.C. and Vursavas, O.M. (2008), "Information retrieval on Turkish texts", *Journal of the American Society for Information Science and Technology*, Vol. 59 No. 3, pp. 407-21.

Chaudhuri, B.B. (2006), *Digital Document Processing: Major Directions and Recent Advances*, Springer, London.

Chen, Y., Ma, W.Y. and Zhang, H.J. (2003), "Detecting web page structure for adaptive viewing on small form factor devices", *Proceedings of the 12th International World Wide Web Conference, Budapest*, pp. 225-233.

Chowdhury, G.G. (Ed.) (2006), *Introduction to Modern Information Retrieval*, Facet, London.

Chung, C.Y., Gertz, M. and Sundarsan, N. (2002), "Reverse engineering for web data: from visual to semantic structures", *Proceedings of the 18th International Conference on Data Engineering, San Jose, CA*, pp. 53-63.

Feng, J., Haffner, P. and Gilbert, M. (2005), "A learning approach to discovering web page semantic structures", *Proceedings of the Eighth International Conference on Document Analysis and Recognition, Seoul*, pp. 1055-1059.

Grossman, D.A. and Frieder, O. (2004), *Information Retrieval: Algorithms and Heuristics*, Springer, Dordrecht.

Guo, Y. and Stylios, G. (2005), "An intelligent summarisation system based on cognitive psychology", *Information Sciences*, Vol. 174 Nos 1-2, pp. 1-36.

Gupta, S., Kaiser, G.E., Grimm, P., Chiang, M.F. and Starren, J. (2005), "Automating content extraction of HTML documents", *World Wide Web*, Vol. 8 No. 2, pp. 179-224.

Hobson, S.P., Dorr, B.J., Monz, C. and Schwartz, R. (2007), "Task-based evaluation of text summarisation using relevance prediction", *Information Processing and Management*, Vol. 43 No. 6, pp. 1482-99.

Ingwersen, P. and Järvelin, K. (2005), *The Turn: Integration of Information Seeking and Retrieval in Context*, Springer, Dordrecht.

Jackson, P. and Moulinier, I. (2007), *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*, John Benjamins, Amsterdam.

Jansen, B.J. and Spink, A. (2005), "An analysis of web searching by European AlltheWeb.com users", *Information Processing and Management*, Vol. 41 No. 2, pp. 361-81.

Kowalski, G.J. and Maybury, M.T. (2002), *Information Storage and Retrieval Systems: Theory and Implementation*, Kluwer Academic, Boston, MA.

Liang, S.F., Devlin, S. and Tait, J. (2007), "Investigating sentence weighting components for automatic summarisation", *Information Processing and Management*, Vol. 43 No. 1, pp. 146-53.

Mani, I. (2001), *Automatic Summarisation*, John Benjamins, Amsterdam.

Mani, I. and Maybury, M.T. (Eds) (1999), *Advances in Automatic Text Summarisation*, MIT Press, Cambridge, MA.

Marcu, D. (1999), "Discourse trees are good indicators of importance in text", in Mani, I. and Maybury, M.T. (Eds), *Advances in Automatic Text Summarisation*, MIT Press, Cambridge, MA, pp. 123-36.

Marcu, D. (2000), *The Theory and Practice of Discourse Parsing and Summarisation*, MIT Press, Cambridge, MA.

Markey, K. (2007), "Twenty-five years of end-user searching, part 1: research findings", *Journal of the American Society for Information Science and Technology*, Vol. 58 No. 8, pp. 1071-81.

Maynard, D., Bontcheva, K., Saggion, H., Cunningham, H. and Hamza, O. (2002), "Using a text engineering framework to build an extendable and portable IE-based summarisation system", Proceedings of the ACL Workshop on Text Summarisation, Philadelphia, PA, pp. 19-26.

Moens, M.-F. (2002), *Automatic Indexing and Abstracting of Document Texts*, Kluwer Academic, Boston, MA.

Montgomery, D.C. (2001), *Design and Analysis of Experiments*, John Wiley, New York, NY.

Mukherjee, S., Yang, G., Tan, W. and Ramakrishnan, I.V. (2003), "Automatic discovery of semantic structures in HTML documents", *Proceedings of the Seventh International Conference on Document Analysis and Recognition, Edinburgh*, pp. 245-249.

Niyogi, D. and Srihari, N. (1995), "Knowledge-based derivation of document logical structure", *Proceedings of the Third International Conference on Document Analysis and Recognition, Montreal*, pp. 472-475.

Otterbacher, J., Radev, D. and Kareem, O. (2006), "News to go: hierarchical text summarisation for mobile devices", *Proceedings of 29th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, WA*, pp. 589-596.

Porter, M. (1980), "An algorithm for suffix stripping", *Program*, Vol. 14 No. 3, pp. 130-7.

Raggett, D., Le Hors, A. and Jacobs, I. (1999), HTML 4.01 specification, available at: www.w3.org/TR/html401/ (accessed 15 February 2008).

Sparck-Jones, K. (1999), "Automatic summarising: factors and directions", in Mani, I. and Maybury, M.T. (Eds), *Advances in Automatic Text Summarisation*, MIT Press, Cambridge, MA, pp. 1-12.

Szlávik, Z., Tombros, A. and Lalmas, M. (2006), "Investigating the use of summarisation for interactive XML retrieval", *Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon*, pp. 1068-1072.

Tombros, A. and Sanderson, M. (1998), "Advantages of query biased summaries in information retrieval", *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador*, pp. 2-10.

Vadrevu, S., Gelgi, F. and Davulcu, H. (2007), "Information extraction from web pages using presentation regularities and domain knowledge", *World Wide Web*, Vol. 10 No. 2, pp. 157-79.

Van Oostendorp, H. and De Mul, S. (1996), *Cognitive Aspects of Electronic Text Processing*, Ablex, Norwood, NJ.

Varadarajan, R. and Hristidis, V. (2005), "Structure-based query-specific document summarisation", *Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen*, pp. 231-232.

White, R.W., Jose, J.M. and Ruthven, I. (2003), "A task-oriented study on the influencing effects of query-biased summarisation in web searching", *Information Processing and Management*, Vol. 39 No. 5, pp. 707-33.

Xue, Y., Hu, Y., Xin, G., Song, R., Shi, S., Cao, Y., Lin, C.Y. and Li, H. (2007), "Web page title extraction and its application", *Information Processing and Management*, Vol. 43 No. 5, pp. 1332-47.

Yang, C.C. and Wang, F.L. (2008), "Hierarchical summarisation of large documents", *Journal of the American Society for Information Science and Technology*, Vol. 59 No. 6, pp. 887-902.

Yang, Y. and Zhang, H.J. (2001), "HTML page analysis based on visual cues", *Proceedings of the Sixth International Conference on Document Analysis and Recognition, Seattle, WA*, pp. 859-864.

Yeh, J.Y., Ke, H.R., Yang, W.P. and Meng, I.H. (2005), "Text summarisation using a trainable summariser and latent semantic analysis", *Information Processing and Management*, Vol. 41 No. 1, pp. 75-95.

About the authors

F. Canan Pembe is a PhD candidate in Computer Engineering at Boğaziçi University, İstanbul, where she received an MSc degree in Computer Engineering. Her research interests include information retrieval, automatic summarisation and natural language processing. She also works as a Teaching and Research Assistant at İstanbul Kültür University. F. Canan Pembe is the corresponding author and can be contacted at: canan.pembe@boun.edu.tr

Tunga Güngör is Associate Professor of Computer Engineering at Boğaziçi University, İstanbul, where he received a PhD degree in Computer Engineering. His research interests include artificial intelligence, natural language processing, machine translation, learning systems, automated theorem proving and theoretical computer science.