Semantic Thumbnails - A Novel Method for Summarizing Document Collections

Arijit Sengupta
Information Systems Department
Kelley School of Business
Indiana University
Bloomington, Indiana 47405
asengupt@indiana.edu

Mehmet Dalkilic*
School of Informatics
Center for Genomics & Bioinformatics
Indiana University
Bloomington, IN 47405
dalkilic@indiana.edu

James Costello
Center for Genomics & Bioinformatics
Indiana University
Bloomington, IN 47405
jccostel@indiana.edu

ABSTRACT

The concept of thumbnails is common in image representation. A thumbnail is a highly compressed version of an image that provides a small, yet complete visual representation to the human eye. We propose the adaptation of the concept of thumbnails to the domain of documents, whereby a thumbnail of any document can be generated from its semantic content, providing an adequate amount of information about the documents. However, unlike image thumbnails, document thumbnails are mainly for the consumption of software such as search engines, and other content processing systems. With the advent of the semantic web, the requirement for machine processing of documents has become extremely important. We give particular attention to electronic documents in XML and in RDF/XML, with a view towards the processing of documents in the semantic web.

Categories and Subject Descriptors

E.4 [**Data**]: Coding and Information Theory—*Data compaction and compression*; H.3.1 [**Information Systems**]: Information Storage and Retrieval—*Content Analysis and Indexing*

General Terms

Algorithms, Documentation, Design

Keywords

Document semantics, Document summarization, Thumbnails, Semantic web

1. INTRODUCTION

In the last few decades, improvements in technology has allowed us to collect and store both data and information[1] inexpensively and easily. The Internet has, similarly, allowed inexpensive and easy "publishing" of these data and information. The grim side of this new information space is that navigation or search is usually difficult, because it is conducted by moving through text—most often a serial list of phrases (and the respective documents in which they appear) that contain matches to keywords initially provided by the user[2]. The obvious motivation here is a chain of relationships: the semantic content of the document is related to certain keywords (syntactic elements) which are, in turn, related to the search terms provided by the user. Unfortunately, as we all have experienced, this connection provides a very high selectivity, but very low specificity. The user is forced to wade through many thousands, if not tens or hundreds of thousands, of documents navigating by keywords alone.

A similar, smaller version of this problem exists on personal computers. Popular operating systems try to help by iconifying the data type, *e.g.*, associating a cup and smokelike swirls with a Java program, an MS Word document with a sheet of paper and a 'W', and so forth. The idea motivating this is that the user can quickly scan different documents and choose the most meaningful. This is improved slightly by adding the file name to the icon—the user being able to make a better evaluation of the contents with little more overhead.

When searching for images instead of text, the visual element becomes even more useful and pronounced. An image is compressed into an image *thumbnail* (IT) that, in a very small footprint, can usually provide enough visual information to the user that a good guess as to its content can be reasonably made. Generating image thumbnails involves symmetric compression of pixels so that in spite of the loss in clarity, the thumbnail still keeps the basic shape and aspect of the original image. We believe that a document can likewise have a *semantic* thumbnail (ST): a "semantically" compressed representation of a document's content that would provide enough meaning to the user, so that it could be visually inspected quickly. We are interested in bringing the best of both visual and textual search. Like an IT, an ST should have a small footprint, and present enough information to make navigation concerning the content simple. We are also motivated by a current project in bioinformatics that requires a significant amount of text searching. We

*Partially supported by NSF IIS0082401 and IBM Lifesciences Grant

SIGDOC'04, October 10–13, 2004, Memphis, Tennessee, USA.

[1]We draw a distinction between data, *e.g.* FASTA format for genomics data, and information, *e.g.*, an experimental biology paper on, say, apoptosis (programmed cell-death). Information is much richer, semi- or unstructured, and is much more difficult to search.

[2]In this work we are not interested in examining how the documents themselves are ordered, say though link analysis.

decided to implement STs as a component of this text search component and develop it on its own called BioKnOT[6].

There have been attempts to provide document thumbnailing as a graphical problem, but from our point of view, there is not enough semantics provided (discussed below). The work on Resource Description Framework (RDF–discussed briefly below) provided us with the inspiration to create what amounts to mini-ontologies for the documents. Discussed more fully below, an ontology is a collection of entities (terms) and their interrelationships.

The concept of thumbnails has been extended for use with textual documents, potentially with embedded images. The research on document thumbnailing essentially treats the original document as an image representing the snapshot of the document when it is viewed, and uses the same compression techniques for image thumbnailing. This can be used for the purpose of quick summarization [16], representation of search results [15], enhanced browsing and scrolling using page thumbnails [1], understanding documents in other languages [13], as well as for the purpose of interactive browsing [11].

Treating documents as images, however, only summarizes layout, and not content. While this is adequate for the purpose of human viewing and browsing, this method of thumbnailing is not appropriate for deriving any semantic content from the thumbnail. In this paper, we adapt the concept of thumbnailing more for the purpose of capturing the semantics of the documents, rather than the layout.

The advent of the semantic web [2] provides an additional motivation for this work. Unlike the current world-wide web, documents in the semantic web are interlinked semantically, and search techniques for this new web will need to adequately, yet efficiently, use such embedded semantic information. For large document repositories, ontologies embedded in STs enable semantic applications (such as search engines) to quickly make retrieval decisions even without indexing.

Automated document keywording and summarizing is not entirely a new concept. Content analysis of documents is a common task for search engines, especially in search engines that do not create full-text indices. Many word processing tools include facilities for automated summarization. In such approaches, frequently occurring keywords are generated and sentences from the documents are ordered. A summary is then generated by picking sentences having the most number of keywords. The problem with this method is that it generates summarizations that are highly irregular, and although they give the appearance of being a readable document, they do not provide enough information for machine consumption.

The most important contribution of this work is in its implications for the next generation of semantic web systems where machines will be required to quickly process large sets of XML documents, often without the opportunity to index them ahead of time.

The rest of the paper is organized as follows. In Section 2, we discuss pertinent literature. In Section 3, we begin discussion of STs at a broad level and in Section 4, we present an overview of the system. Finally, we discuss proposed evaluation methods in Section 5, and conclude in Section 6.

2. LITERATURE REVIEW

Research in the context of compressing document content can be broadly classified into three groups based on the purpose of the compression:

1. *Thumbnailing*: Thumbnailing is primarily a visualization technique used for better interactive handling of large documents or document collections. Typically the thumbnail of an image representing the layout of the document is shown, potentially one image thumbnail (IT) for each page (*e.g.*, [1]). The purpose of thumbnailing is primarily to retain the layout, since the thumbnails have no content information. The size of the thumbnails can be easily controlled by the user.

2. *Summarization*: Summarization is the process of extracting keywords or potentially complete sentences that capture the text of the document. No layout information is retained. Some of the semantics of the document is captured. Again, the size of the summaries can be controlled by the user.

3. *Compression*: Compression is the process of reducing the size of a document by algorithms that make use of the unused bit spaces and repetitions in the document. This is an altogether different dimension, since usually compressions are lossless and reversible. The size of the compressed document depends on the algorithm used, and cannot be controlled by the user.

In order to properly motivate this research, we will consider current work being done in all three of the above areas. Compression is important because it plays a significant role in reducing bandwidth, although in this context its less important since the compressed documents are not human readable. The goal of this work is to produce summaries of documents that are effectively readable by both human and machine, and that can capture a significant portion of document semantics. Figure 1 shows how the above three directions compare with our approach of semantic thumbnails. We briefly discuss some of the research in the above three areas below.

2.1 Document thumbnailing

The concept of "document thumbnailing" is used in the process of interactive information retrieval [15]. The process of information retrieval consists primarily of retrieving documents based on a Boolean combination of keywords [17]. With the advent of the Web and the explosive growth in the number of documents accessible on the Web, the concept of information retrieval has been adapted to not only retrieve the documents, but also to appropriately rank, group, and associate documents [4].

Proper thumbnailing of documents is an important issue when documents are in a language unknown to the reader. Querying documents in a repository of multi-lingual documents involves the translation of either the searchable documents or the query itself [14]. Ogden and Davis [15] show that with human interaction, cross-language text retrieval can be improved using thumbnailing techniques. In such techniques, an overview of the search results is displayed using document thumbnails, in which the presence of the search terms can be visually indicated. The underlying language of documents does not play a role since the thumbnails are readable. The thumbnail view significantly improves the user's selection of search results [13, 16]. In addition, use of thumbnails can vastly improve the usability of overview+detail type document retrieval interfaces [20].

2.2 Document summarization

Document summarization is a heavily researched area in information retrieval. The generation of terms by automatic analysis of documents typically uses a controlled vocabulary list or a thesaurus [18]. These techniques are particularly useful for the purpose of text mining. Standard information retrieval methods include statistical features such as term frequency inverse document frequency (TFIDF) [19]. In these methods, semantic proximity

between words is computed using statistical methods, sampling several documents in the collection of a large number of documents. Multi-document summarization is especially important for the purpose of ranking search results. Multi-document summarization (e.g, in [11]) consists of content selection and filtering using statistical techniques, followed by proper content presentation preserving the original document ordering.

In addition to keyword generation, several techniques exist for automatic generation of readable text, consisting of full sentences. The most common method involves the ranking of sentences in the document for potential inclusion in the summary using a weighted combination of statistical and linguistic features [8, 12]. Incorporating linguistic analysis in addition to statistical analysis in such techniques have shown to significantly improve the generated summaries, and hence, the relevance of the retrieved documents.

2.3 Document Compression

The two compression formats referred to in this section are lossless and lossy. Both of these terms refer to whether or not, during the compression of a file, any of the original data is lost. Lossy compression, *e.g* the JPEG image file, permanently eliminates data during compression. Thus, when a lossy file is uncompressed, not all of the original data is present (although the human eye or ear may not be able to detect the loss). Lossless compression, *e.g.* the GIF image file, does not eliminate any of the original data during compression. Thus, when a lossless file is uncompressed, all of the original data is present.

Although not immediately relevant to the current research, the concept of compression indirectly fits in this context. Compression involves a lossless reduction of document size, primarily using binary representation of the characters, and making use of unused bits and repetitions in the documents [22]. A particularly interesting observation is that because of the highly repetitive content of XML documents, compression of XML documents can result in very high compression ratio [10, 21]. Use of semantics and allowing compression loss can result in more controlled compression of XML documents as well [5].

Although different from the lossy concepts of thumbnailing and summarization, both of which result in irreversible loss of information in the original documents, compression provides a basis on which thumbnailing tasks can be compared.

2.4 Discussion

Figure 1 shows a graph that places the above three document reduction methods in a 2x2 quadrant. The graph shows that compression retains both structure as well as semantic information of the documents, but since compressed documents are not human readable, and requires potentially processor-intensive decompression techniques to be usable, they are not suitable for fast searching and ranking. Document thumbnails are highly user-centric, and retain the document structure (layout), but they are not semantically rich, and cannot be used for machine-based automated retrieval. Document summarization provides adequate amount of information for automated retrieval methods, but loses semantic knowledge embedded in the document. This leads to the conceptualization of STs that fill the void. STs provide semantically rich thumbnails of documents that can be used for the purpose of user-centric, as well as machine-centric, retrieval purposes, while retaining adequate amount of semantic information within the documents.

3. SEMANTIC THUMBNAILS

As discussed above, STs have many potential applications from visual searching by human agents to parsing, classification, and

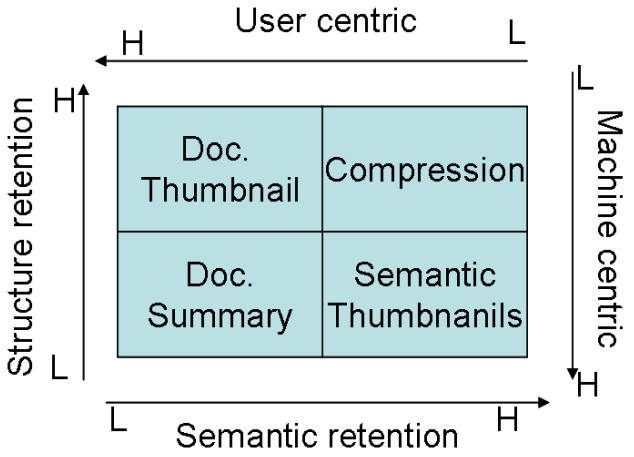

Figure 1: Comparison of the document thumbnailing, summarization and compression methods with respect to layout and semantic retention, as well as user and machine centricness. H=high, L=low

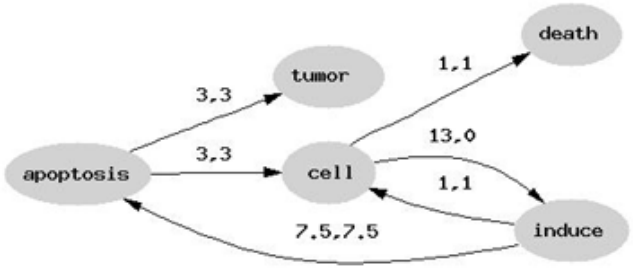

Figure 2: An ST over biological terms. The nodes in the graph are a subset of T_f. The edges reflect likely semantic significance based on nearness. Similarity between STs makes use of a log-odds model of location called a scoring matrix. An edge not present between to nodes means the terms are likely unrelated.

searching via machine agents. BioKnOT[3] is a practical application of STs that illustrate how useful and effective they can be in the bioinformatics setting.

BioKnOT is an interactive document retrieval system that allows users to quickly and easily "drill-down" on a topic. It implements the use of STs and also allows for the iteration of document sets, which allows for refinement of the specificity of a user's search. To aid discussion we present some notation. Formally, we have a set of documents D. By d_i we mean the i^{th} document in D and write $t \in d_i$ to mean term t occurs in document d_i. Let T_f denotes a set of terms from the documents of D, formally, $T_f \subseteq \{t \in d | d \in D\}$.

DEFINITION 1. *A semantic thumbnail for a document $d_i \in D$ is a directed, weighted graph $G_i = \langle V_i, E_i \rangle$ where $V_i = \{t | t \in T_f \wedge t \in d_i\}$, the set of nodes, is a collection of terms, and an edge $E_i \subseteq V_i \times V_i \rightarrow \mathbb{R}^2_{\geq 0}$ is a pair of weights that reflect intra-sentence and inter-sentence signficance. For this paper, we are focusing on the intra-sentence value. (see Fig.2).* □

Since the STs are built dynamically and interactively centered on the user, we describe the process here and treat some of the

[3](http://biokdd.informatics.indiana.edu/jccostel)

important elements in detail further in the paper. STs are built by first identifying the important nodes by TFIDF, then by establishing the weight of the edges through analysis of nearness by looking through the corpus of selected documents (discussed later in this section).

The generation of STs is initiated by a Boolean search provided by the user (see Figure 3). Those documents of D for which the Boolean function is true are used for T_f generation[4]. From T_f a *scoring matrix*[9] S_{ij} is created that indicates numerically whether pairs of words i, j, where $i, j \in T_f$, that occur within a certain reading frame of no more than 20 words (arrived at experimentally), are likely to be present other than by chance. The value is actually a log-odds ratio comparing observed frequency to, in this case, a random model, which is found from the set of documents that meet the Boolean search criteria. Scoring matrices are a universal tool in sequence alignments techniques that allow for disparate, though related molecules, to be substituted for one another in a sequence of molecules, and therefore, allow for non-identical sequences to be compared (see [6] for a more complete discussion). The scoring matrix is used to compare STs generated from the abstracts of the documents with a ST constructed from user input. The relationships are captured in the spirit of ontologies.

After the user has been prompted to choose the most relevant terms from T_f to the search, denoted T_u, she or he has the option of proceeding through BioKnOT by entering more information or simply doing a "quick search" and going directly to the results page.

If the user wants to enter more information for a more precise search, then after selecting the most relevant terms, T_u, the user will be prompted to enter a statement describing what type of document is being searched for, which will contain words from T_u that are then used to create a user-defined ST. This ST is critical to BioKnOT because it provides a means of capturing semantically related terms *with respect to the user* and is used to order the relevant documents' STs using S_{ij}. The user is then queried for how strong the relationships should be, owing to the fact that not enough information is provided by one or several sentences. All of this provided data is then used to score the documents within the documents selected from D.

The quick search option allows the user to bypass entering any more information after terms T_u are selected. The ST that would have been created from the user's input is derived by considering all term relationships from T_u with a distance of 1, rather than the distance that would have been determined from the user's statement. This ST will be used to score the documents within the document database (D).

Once a set of documents has been returned, the user can refine the search by selecting the documents that are most closely related to the user's ultimate search goals. A new T_f is created and the process continues as before. Choices that the user made in the previous search are used as default values and can be, in the case of the user's statement, either added to or modified.

Additionally we provide a kind of support through a second scoring mechanism that takes into account the number of times a document has been cited, weighting the more recently cited papers more. Thus, documents have a pair of values $[r, s]$, where r is what we call the semantic relevance score taken from the scored STs, and s, which is the support value taken from the citation information.

3.1 Choosing STs terms

The node terms in the ST are found through TFIDF calculations, which ranks document terms based on how often a term appears in an document and how many documents contain that term. TFIDF is the product of the term-frequency (TF) of a word and the inverse-document-frequency (IDF) of that word [3]. The intuition underlying TF is that a term is more important in describing information in a document if the term occurs more often, with the exception of stop words (*e.g., a, the, at, ...*)[3]. Let $tf_{i,d}$ denote the number of occurrences of the i^{th} term in document $d \in D$:

$$tf_{i,d} = \frac{|d_i|}{|\sum_j d_j|} \tag{1}$$

where $|d_i|$ is the count of term i in document d and $|\sum_j d_j|$ is the sum of all the terms j in document d.

The intuition underlying IDF is that if a term appears in many documents, then it is less significant in describing information than in a single document[3]. It is defined as the log of the total number of documents n over the total number of occurrences of term i in a document d from the total sets of documents D:

$$idf_{i,D} = log_2(\frac{|D|}{|\{d_i|d_i \in D\}|}) \tag{2}$$

The product of TF and IDF gives a term weight, where the weight of the i^{th} term in the d^{th} document can be defined as follows:

$$weight(i,d) = \begin{array}{ll} (1 + tf_{i,d})idf_{i,D} & \text{if } tf_{i,d} \geq 1 \\ 0 & \text{if } tf_{i,d} = 0 \end{array} \tag{3}$$

The TFIDF calculations are done through a well-known previously built system called LUCAS [7]. The system has a term representation database that was compiled at UC Berkley and Stanford, which contains document frequencies and ranks of 31,928,892 terms found on 49,602,191 pages on the internet (http://elib.cs.berkeley.edu/docfreq/).

3.2 Weighting Edges

The edges are a measure of distance between terms contained in T_u and are present in the user's statement in a sentence (inter-sentence) and among sentences (intra-sentence).

The inter-sentence and intra-sentence distances are a way of capturing a relationship between terms. Our motivation in seeking this distinction between inter-sentence and intra-sentence distance was that, for example, sentences like the ones below are very likely to have different meanings with respect to the relationship between the two words *mitochondria* and *permeability*:

> "... and is not present in the *mitochondria*. *Permeability* is another..."

> "... *mitochondria permeability* is an important aspect of apoptosis..."

Though both words are important individually, to some degree, the way they are presented in and among sentences offers us better insight on their relationship to each other.

3.3 Interactive vs. Automated Thumbnailing

BioKnOT currently does interactive Thumbnailing, where the user is involved in the generation of the ST with a Boolean query, and also in the process of subsequent relationship derivation phases. An automated system is currently being implmented. In general, a random component is added and several STs are created. The

[4]Currently, we have set the size of T_f at 50, though this will be made adjustable in the future.

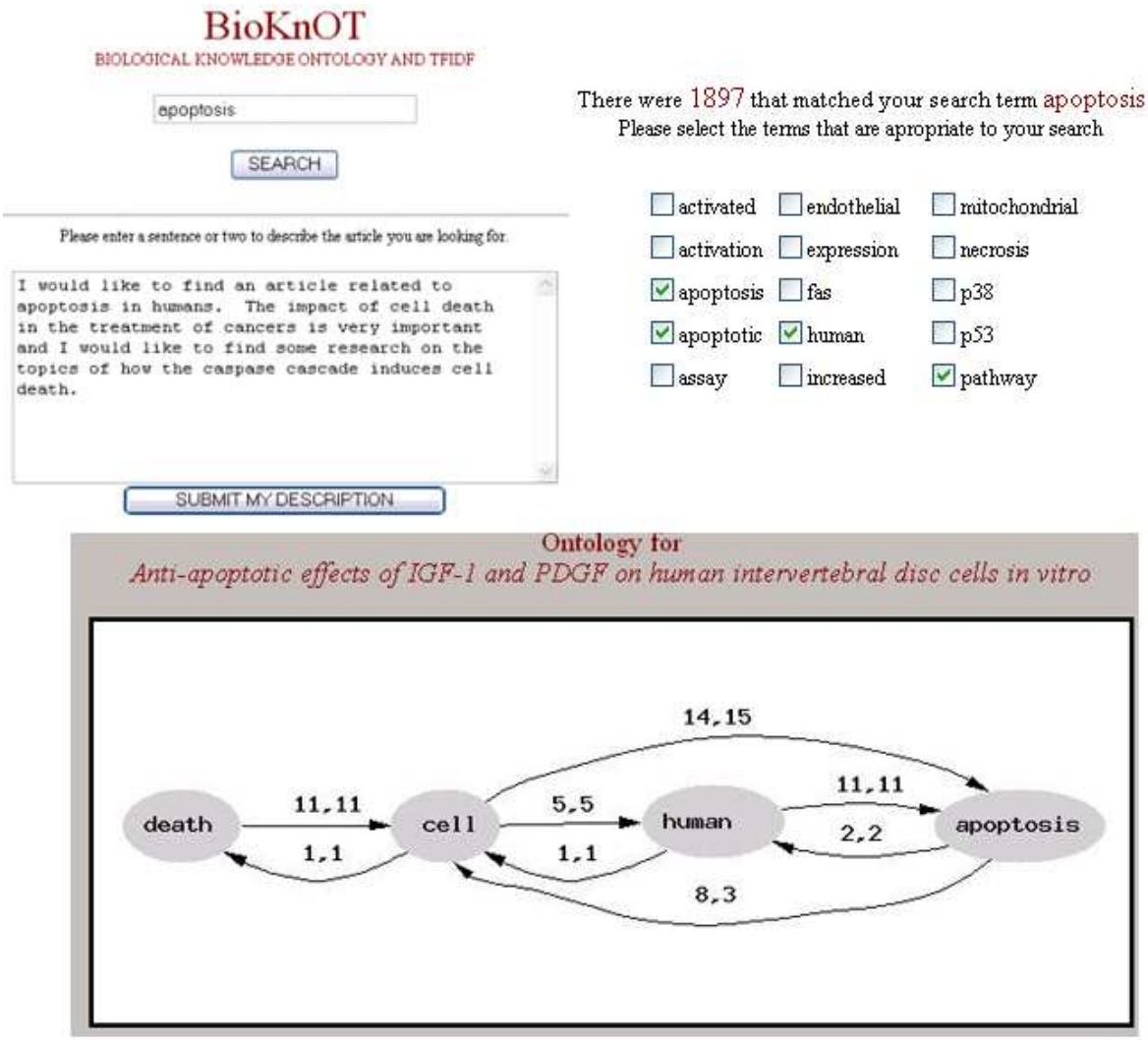

Figure 3: Screen shots of BioKnOT showing the user's semantic thumbnail creation and an ST of a relevant document

general procedure is to sample documents randomly, producing T_f from TFIDF and continuing with the quick search and default values. Our results will be presented in forthcoming work.

4. SYSTEM AND PROGRAM ARCHITECTURE

BioKnOT (figure 4) consists of 5 core interfaces that interact with a document database. The mode of communication from client to server is CGI with Perl 5.8.0. The DBI Perl module was used to interact with the database.

Figure 4 shows an illustration of the flow of the program. First the user enters a query on the initial search page. (1)A TFIDF calculation is done on the initially searched documents and the user is asked to rank these terms on the filter page. Next (2), the user is asked to enter a few sentences stating what type of document is being searhed for. (3) The scoring matrix is built and term relationships are constructed, and then the user is asked to supply these relationships with a score. (4) All the documents are scored and returned based on rank to the results page, which supplies the user with document data, illustration of the term relationships, and the URL to the document itself. The results page also serves as the re-

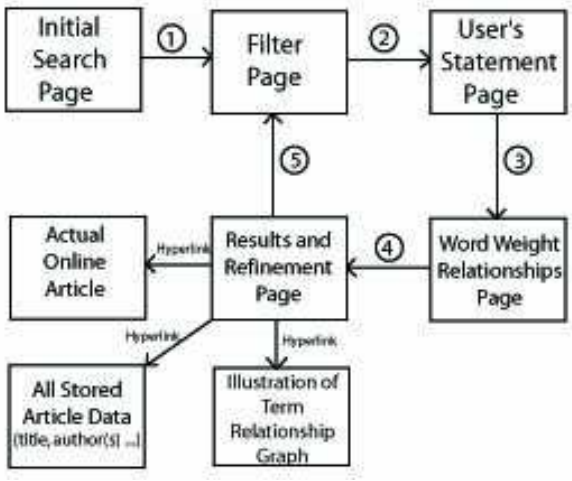

Figure 4: Illustration of BioKnOT program flow

finement page, which (5) allows the user to iterate over the search with a more specific set of data, based on selected documents instead of a random model.

The first web-based interface is the initial query page. This is where a user can enter Boolean search terms and using Boolean logic, a query is dynamically created to search the document database in the abstract and title fields.

The random model for comparing documents consists of the set of documents that meet the Boolean search criteria, noted D_s. The abstracts from set D_s are then pooled into a text file that will be used for the TFIDF calculations. This text file is passed to LUCAS. LUCAS is written in Java and communicates with BioKnOT through the SOAP protocol. Behind LUCAS is a term representation database, which is needed for the inverse document frequency calculations. The top 50 returned words from LUCAS, noted T_f, are used to create the filter page.

The filter page, which places set T_f into an HTML form, asks the user to select the most relevant terms to the search from T_f. These set of user selected terms, noted T_u, are stored in hidden HTML fields and the user is given two options to proceed. First, the user can select the "Quick Search" option, which will bring the user directly to the results page, or second the user can select the "Enter More User Input" option, which will prompt the user to enter more data for a more precise search.

If the user selects the "Enter More User Input" option, the set T_u is stored and the user is prompted to enter several sentences on what type of document is being searched for. These sentences are supposed to give a more user defined description of the document being searched for and would include some of the terms included in T_u.

There are two major events that happen at this point. First, a scoring matrix is created to show how strong the relationships between the terms in set T_u are when compared to the random model, D_s. Next, a user defined ST is created to show the relationship of terms that are present within the user's statement and set T_u. This ST will be the standard of comparison for all other documents.

The scoring matrix is created by taking the log-odds score (equation 4) for each term relationship within the set of terms T_f.

$$LOD = \log_2(\frac{P(t_2|t_1,\delta)}{P(t_1) \times P(t_2)}) \tag{4}$$

In equation 4, $P(t_1)$ and $P(t_2)$ are calculated as indicated in equation 5, where the count of a specified term is taken within D_s and then divided by the total count of all the terms within D_s.

$$Pt_i = \frac{|t_i|}{|\sum_i d_i|} \tag{5}$$

$P(t_2|t_1,\delta)$ in equation 4 is calculated by taking the probability of finding term 2 within a distance δ from term 1. We then divide this value by the total count of term relationships that are found in D_s. This equation is illustrated below.

$$P(t_2|t_1,\delta) = \frac{|\{t_2|t_1,\delta\}|}{|\{t_j|t_i,\delta \wedge i \neq j\}|} \tag{6}$$

This scoring matrix is created dynamically every time a user enters some search criteria and is iteratively done for each term within set T_u.

When the term relationships are found and the scoring matrix has been assembled, the user is then prompted to add weight to the user defined relationships, derived from the user's statement and set T_u, based on how important they are to the search. This additional information is then added to the scoring matrix as a multiplier to reflect the amount of relevance to the search these relationships represent.

After all this data has been interactively collected, STs of each documents' abstract are created using set T_u and then scored against the user defined ST using the scoring matrix.

The STs are compared to each other through an adjacency matrix. This matrix takes the ST, which is represented as a graph, and converts it into a structure that is comparable with other STs. Based on the presence or absence of a term relationship in the user defined and individual document STs, the scoring matrix value for that relationship, and the distance from term to term in that relationship, a score is generated for that term relationship. The sum of these relationship scores is used for the total semantic score of a document.

The user is then shown the top ranked documents as determined by semantic score. Along with the semantic score, a citation rate is supplied to provide the user with some more input on how important a document has been over the years.

In addition to the scores, three important hyperlinks are provided. First, a link to the actual online document is given. Next, a link to all the data in the database related to that document is provided, such as title, author(s), abstract, etc. Lastly, a link to the term relationship ST is provided. This ST will show the user how the terms are related to each other in the abstract of the document, as well as supplies the user with a way to visually compare document semantic relationships.

The overall concept behind BioKnOT is to supply the user with an effective interactive way to find documents related to very specific search criteria. Knowing this, one of the most important features of BioKnOT is to allow the user to do an iterative search over the database with more strictly defined search criteria.

The results page also provides the user with a means to further narrow down a search by selecting documents that are related to the user's specifications. After these documents have been selected, LUCAS is called again, but instead of the large broad sample set that was used on the first pass of the search, a very specific user selected set of documents are used to create the term filter page.

The process is then started all over again and can be run indefinitely.

BioKnOT saves the state of the users' previous searches and passes that information along to the scoring of the documents. The user is supplied this information and can change it at any time during the search. Figure 3 shows a screen image of different parts of the system.

5. EVALUATION

An initial pilot of BioKnOT was performed using data from the PubMed Life Sciences journals database[5] and the Gene Ontology (GO) Consortium[6]. The result from the pilot study was encouraging, and it demonstrated that without the presence of XML tags, the semantic thumbnails contain adequate amount of information regarding the document keywords. In addition, the relationships in the generated ontology adds more semantic knowledge regarding the documents. With the presence of XML tags, however, the ontologies generated become much more precise. A full study of the quality of the STs generated is currently being prepared. In this study, novice users will be given a collection of retrieval tasks. We will perform a between-groups study with one group having the semantic thumbnail information, and one group with only generated keywords. The efficiency (task completion speed) and accu-

[5]http://www.pubmedcentral.nih.gov/

[6]http://www.geneontology.org

racy (task answer correctness) will then be statistically compared to measure differences. The results from the pilot indicate that potential differences in user perspective do exist.

6. CONCLUSION AND FUTURE WORK

We have presented a framework for document summarization utilizing the semantic content embedded in documents. This summarization, which we call Semantic Thumbnails (ST) provides a means for visualizing and comparing the document content at a high level. These thumbnails capture more semantic information from documents than purely graphical representations of search results, as well as visual representation of the layout of the documents.

The generated thumbnails have a number of highly desirable properties. First of all, semantic thumbnailing is closed in the document format, *e.g.*, the generated structure for an RDF document is valid RDF, although the summary documents do not correspond to the original RDF schema. The most important aspect of this summarization strategy is in its accuracy of recall for purely keyword-based searches.

For future work, we are investigating other techniques to derive the semantic content than term frequencies. Also, we are implementing a method for automatically generating the document STs without user interaction. We are also in the process of developing STs for RDF/XML documents by utilizing the ontologies already embedded in such documents. We are also in the process of generating our own TFIDF repository (instead of LUCAS) — presumably from bioinformatics documents to more closely reflect the domain. Lastly, we are implementing a temporal component of the Semantic Thumbnails to take into account the timeliness of the content of the documents.

Acknowledgements

We thank Dr. Dennis Groth and Dr. Javed Mostafa for their valuable comments during the process of developing the work.

7. REFERENCES

[1] Adobe Systems, San Jose, CA, USA. *Adobe Reader 6.0 for Windows and Macintosh User Manual*, 2003.

[2] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.

[3] M. W. Berry. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer-Verlag New York, inc., New York, NY, 2004.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] M. Cannataro, G. Carelli, A. Pugliese, and D. Sacca. Semantic lossy compression of XML data. In *Knowledge Representation Meets Databases*, 2001.

[6] M. Dalkilic and J. Costello. BioKnOT: Biological knowledge through ontologies and TFIDF. In *Proceedings, Workshop on Search and Discovery in Bioinformatics, SIGIR-Bio*, 2004.

[7] Y. Fu and J. Mostafa. Toward information retrieval web services for digital libraries. *JCDL*, June 2004.

[8] J. Goldstein, M. Kantrowitz, V. O. Mittal, and J. G. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Research and Development in Information Retrieval*, pages 121–128, 1999.

[9] I. Korf, M. Yandell, and J. Bedell. *Blast*. O'Reilly & Associates, 2003.

[10] H. Liefke and D. Suciu. XMill: an efficient compressor for XML data. In *Proceedings, ACM SIGMOD 2000, SIGMOD RECORD 29(2)*, pages 153–164, 2000.

[11] C.-Y. Lin and E. Hovy. From single to multi-document summarization: a prototype system and its evaluation. In *Proceedings of the 40th Anniversity Meeting of the Association for Computational Linguistics (ACL-02)*, Philadelphia, PA, USA, 2002.

[12] K. McKeown, R. Barzilay, D. Evans, et al. Columbia multi-document summarization: Approach and evaluation. In *Proceedings of the Workshop of Text Summarization, ACM SIGIR 2001*, 2001.

[13] W. Ogden. Getting information from documents you cannot read: An interactive cross-language text retrieval and summarization system, 1999.

[14] W. Ogden, J. Cowie, M. Davis, E. Ludovik, S. Nirenburg, H. Molina-Salgado, and N. Sharples. Keizai: An interactive cross-language text retrieval system.

[15] W. C. Ogden and M. W. Davis. Improving cross-language text retrieval with human interactions. In *HICSS*, 2000.

[16] W. C. Ogden, M. W. Davis, and S. Rice. Document thumbnail visualization for rapid relevance judgments: When do they pay off? In *Text REtrieval Conference*, pages 528–534, 1998.

[17] G. Salton. Developments in automatic text retrieval. *Science*, 253:974–980, 1991.

[18] G. Salton, J. Allan, C. Buckley, and A. Singhal. Automatic analysis, term generation and summarization of machine readable texts. *Science*, 264:1421–1426, June 1994.

[19] G. Salton and C. Yang. On the specification of term values in automatic indexing. *Journal of Documentation*, 29:351–372, April 1973.

[20] B. Suh, A. Woodruff, R. Rosenholtz, and A. Glass. Popout prism: Adding perceptual principles to overview+detail document interfaces, 2002.

[21] P. Tolani and J. R. Haritsa. XGRIND: A query-friendly XML compressor. In *ICDE*, 2002.

[22] T. Welch. A technique for high-performance data compression. *IEEE Computer*, 17(6):8–19, 1984.