Synopses Generation for Specialized Document-Element Search Engines

Sumit Bhatia
Computer Science and Engineering
The Pennsylvania State University
University Park, PA-16802, USA
sumit@psu.edu

Prasenjit Mitra
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA-16802, USA
pmitra@ist.psu.edu

ABSTRACT

Scientists often want to search for document-elements like tables and figures in digital documents. Using a document-element search engine helps them to retrieve a set of document-elements using keyword queries. Consequently, they need to decide whether the returned document-element is useful and then determine what information is contained in it. The last step is typically done by downloading the paper and reading it. In this paper, we investigate how to extract information (synopsis) related to document-elements from documents automatically. The extracted information can be indexed and provided along with the search results, enabling the end-user to quickly find the related information. Thus, this work has significant potential to facilitate ease-of-use for a document-element search engine, consequently increasing the productivity of the end-user. We propose a novel method to extract synopses, investigate the optimum synopsis-size and demonstrate the utility of our extracted synopsis in document-element understanding with a user study.

Categories and Subject Descriptors

H.5 [**Information Systems**]: Information Interfaces and Presentation

General Terms

Algorithms, Human Factors

Keywords

summarization, snippets, synopses, document-elements

1. INTRODUCTION

In academic writing, authors use a number of document-elements for a variety of purposes like reporting and summarizing experimental results (plots, tables), describing a process (flow charts) or algorithm (pseudo-code) etc. A *document-element* is defined as an entity, separate from the running text of the document, that either augments or summarizes the information contained in the running text of the document. Figures, Tables and Pseudo-codes for algorithms are the most commonly used document-elements in scientific literature and are sources of valuable information. Recently, significant efforts have been made to utilize and extract the information present in these document-elements. Kataria et al., describe algorithms to extract data from 2-D plots which can then be stored, indexed and eventually, queried[5]. *TableSeer*, a specialized search engine allows end users to search for tables in digital documents[6]. A specialized search engine for biology documents, *BioText Search Engine*, offers capability to search for figures and tables in the documents[4].

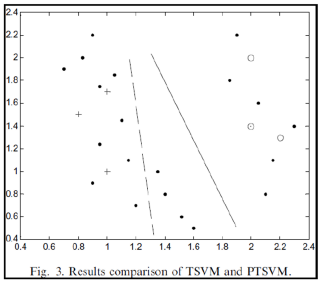

Figure 1: A sample figure and its caption. Figure is taken from[2].

Such special-purpose document-element search engines return a list of document-elements and a snippet constructed from the document. Often the end-user wants to examine more information than available in the snippets because the end user can not always interpret the information content of the document-elements by examining just the snippet or even the document-element itself as illustrated by Figure 1. It is hard to interpret the results just by looking at the figure itself as the figure does not tell anything about what the lines and different points in the figure mean. Even though the associated caption helps in understanding the purpose of the figure, it does not provide the details necessary to interpret and understand the figure completely.

In this work, we show how we can extract information from documents related to document-elements present in the documents automatically. We refer to this relevant information as a *synopsis*. Availability of a concise and relevant synopsis saves the end-users' time when they are examining the search results to find information that satisfies their information needs. In Figure 2, we show the synopsis generated by our tool for the figure shown in Figure 1 and it can be seen that the message of the figure now becomes much clearer with this additional information. Thus, our tool increases the degree of automation of information seeking and improves productivity of end-users.

Copyright is held by the author/owner(s).
WWW2009, April 20-24, 2009, Madrid, Spain.
.

Fig. 3 illustrates the training results of TSVM and PTSVM on Tutorial dataset.The solid line is the final hyperplane found by PTSVM and the dashed line is the final hyperplane found by TSVM.As shown in Fig. 3, the wrong estimation for the value of N is responsible for the bad per- formance of TSVM.This problem is successfully avoided in PTSVM. We can also find out that the training time of PTSVM is much shorter than that of TSVM.This is mainly due to the fact that TSVM need to successively increase the value of C and the calculation has to be done for every C value.

Figure 2: Synopsis generated by our system for the figure described in Figure 1.

Extracting a synopsis of a document-element from a digital document involves filtering the information related to the document-element from the rest of the information in the document. Solving the problem accurately can be easy if we could understand the semantics of the text automatically. However, state-of-the-art techniques from natural language processing and statistical text processing techniques still fall short in fully understanding the semantics of text in documents. Additionally, good synopsis generation involves making the judgment call about what level of detail is useful for the end-user. If we generate a very large synopsis, the users' needs of finding information quickly will not be met.

In this paper, we propose an algorithm for extracting synopses of document-elements from digital documents automatically. First, the caption and references (sentences referring directly) to the document-element are extracted and then, the algorithm generates a content similarity score for the other sentences in the document with the caption and reference sentences. We adapt Okapi BM25 weighing scheme[7] for this purpose. Each sentence is also assigned a *Distance Score* which varies inversely by its proximity to the reference sentence. Then the algorithm uses the top ranked sentences and a simple model that tries to strike a balance between the information content and length of the synopsis. The parameters of the model are determined empirically so as to optimize user satisfaction. We demonstrate the utility of synopsis generation for document-elements and validate our approach by a user study. To the best of our knowledge, this is the first work that introduces the concept of synopsis of document-elements and presents a method to automatically extract synopses from the documents.

2. PROBLEM FORMULATION

We now formulate the problem formally and describe the notations that are used in the rest of this paper. For a digital document $\mathcal{D}$, let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of sentences in the document. We define $D = \{d_1, d_2, \ldots, d_m\}$ to be the the set of *document-element*s in $\mathcal{D}$ with $C = \{c_1, c_2, \ldots, c_m\}$ as the set of associated captions, where c_i is the caption associated with d_i. Assuming that each *document-element* is referred at least once in the document text,we define a *Reference Sentence* $r_{ij} \in S$ as the j^{th} sentence that makes an explicit reference to d_i and let $R_i = \{r_{i1}, r_{i2}, \ldots, r_{ik}\}$ be the set of reference sentences for d_i. For each d_i, we define its synopsis as the set $S_i = \{s_k : s_k \in S$ and s_k satisfies $\mathcal{P}(s_k)\}$, where $\mathcal{P}$ is an appropriately defined predicate.

3. PROPOSED APPROACH

The pseudocode describing our approach is shown in Algorithm 1 and is described in detail below.

Algorithm 1 Algorithm to generate synopsis for the *document-element* d_i.

Input: Set S of all the sentences in the document, Caption c_i and set of reference sentences R_i for d_i.
Output: Synopsis S_i of d_i.

```
Set of sentences constituting the synopsis, S_i ← φ
▷ Generate synopsis for each document-element.
for each reference sentence r_ij ∈ R_i do
    Q_ij ← Query formulated with c_i and r_ij
    ▷ Compute scores for each sentence s_k ∈ S
    for each s_k ∈ S do
        Score, score_k ← BM25(Q_ij, s_k) + Distance(r_ij, s_k)
    end for
    Sort all the sentences in non-increasing order of their scores
    so that for all i < j, score_i ≥ score_j
    j ← 1
    U ← Utility of s_j as defined by equation (1)
    while U > 0 do
        S_i ← S_i ∪ s_j
        j ← j + 1
        U ← Utility of s_j as defined by equation (1)
    end while
end for
return S_i
```

3.1 Pre-Processing

The process of synopsis generation starts with the conversion of digital documents (pdf format) into text format followed by sentence segmentation which splits up the document text into its constituent sentences and yields the sentence set S.

3.2 Gathering Information Cues by Extracting Captions and Reference Text

Captions provide important information that helps in understanding the associated document-elements. In order to utilize the information cues present in the captions, we use following grammar to distinguish and extract caption sentences from rest of the sentences:

⟨CAPTION⟩::=⟨DOC_EL_TYPE⟩⟨Integer⟩
⟨DELIMITER⟩⟨TEXT⟩
⟨DOC_EL_TYPE⟩::=⟨FIG_TYPE⟩|⟨TABLE_TYPE⟩|
⟨ALGO_TYPE⟩
⟨FIG_TYPE⟩::=FIGURE|Figure|FIG.|Fig.
⟨TABLE_TYPE⟩::=TABLE|Table
⟨ALGO_TYPE⟩::=Algorithm|algorithm|Algo.|algo.
⟨DELIMITER⟩::= : |.
⟨TEXT⟩:⟨A String of Characters⟩

Though captions provide some details about the element of interest, they are not sufficient. In order to get complete understanding of the content and context of the document-element under consideration, one has to analyze the running text also[3]. Assuming that each document-element is referenced at least once in the running text, we identify and extract the set of reference sentences in a similar way. Note that in the reference sentence, the delimiter will not be present in most cases.

3.3 Scoring and Ranking of Sentences

The extracted captions and reference sentences for each document-element are then used to rate the sentences in the document on the following two criteria:

1. **Content Similarity and Relevance:** A set of keywords are extracted from the caption and reference sentence to generate a query which provides cues about the information contained in the document-element. Using Okapi BM25[7] as the similarity measure, the query thus generated is then used to assign *Similarity Scores* to all the sentences in the document based on their similarity to the query. If q is the generated query then the BM25 score of sentence s in document $\mathcal{D}$ is computed as:

$$BM25(q,s) = \sum_{t \in q} \log \frac{N}{sf_t} \cdot \frac{(k+1)tf_{ts}}{k((1-b)+b \times (\frac{l_s}{l_{av}})) + tf_{ts}} \quad (1)$$

where:
N is the total number of sentences in the document,
sf_t is the sentence frequency, i.e.,number of sentences that contain the term t,
tf_{ts} is the frequency of term t in sentence s,
l_s is the length of sentence s,
l_{av} is the average length of sentences in D,
k and b are constants which are set to 2 and .75 respectively.

2. **Proximity:** Generally,when a reference to a *document-element* is made in the running text, the nearby sentences are also describing the content of the *document-element*. For this reason we assign a *Distance Score* to all the sentences that decreases exponentially with their distance from the reference sentence. Let r be the reference sentence and s be the sentence under consideration and p_r and p_s be the positions of r and s in the document, then the *Distance Score* is computed as:

$$Distance(r,s) = e^{-0.5|p_r - p_s|} \quad (2)$$

Both the *Similarity Score* and *Distance Score* for a sentence are normalized in the range[0,1] so that the total score for a sentence lies in between 0 and 2.

3.4 Presentation of Synopses to User

After scoring and ranking all the sentences, we need to decide how many and what sentences to include in the synopsis to be presented to the user. Carbonell et al., describe Maximum Marginal Relevance as a criterion for selecting sentences for summarization that combines query-relevance and information novelty[1]. For a complete document, like a paper, there are a lots of sentences that convey the same information, for example, sentences in abstract, introduction, conclusion etc. Given that for a document-element, we get only a small subset of sentences that are related to the document-element, chances are very few that the small set of candidate sentences will introduce redundancy. However, presenting all these relevant sentences to the user has a detrimental effect on the readability of the synopsis, requires more time to read and understand and hence, defeats the whole purpose of making the search results more user friendly. Hence, it is required to determine an optimum synopsis size that balances the trade-off between *information content* and *readability and effectiveness* of the synopsis.

Let the score of k^{th} sentence be $score_k$ and let all the sentences be ranked in decreasing order of their scores so that $i < j$ implies $score_i \geq score_j$. We define:

$$U_k = \sum_{i=1}^{k} score_i - P \sum_{i=1}^{k-1} i \quad (3)$$

Here, U_k is the *utility* of the k^{th} sentence and measures whether it is useful to include a sentence in the synopsis or not. We include a sentence in the synopsis if and only if its utility is greater than zero. Thus, our predicate $\mathcal{P}$ here as described in section 2 is $U_k > 0$. Utility of a sentence is determined by two competing factors – (a) The increase in information content by including sentence s_k which is measured by the first term in utility equation that represents the information content of the synopsis if all sentences up to s_k are included; (b)Penalty incurred by adding the additional sentence s_k to the synopsis. The parameter P here is the *Penalty Parameter* which controls the magnitude by which the sentences are being penalized and thus, determines the length of the synopses. If $P = 0$, no penalty is being incurred by adding the additional sentences and we will have the whole document as the synopses. If P is very high, we will have very short synopses. We determine the value of P empirically as described in the next section.

The sentences thus selected are now arranged in the order in which they appear in the document. Further, the non-consecutive sentences are separated by ellipsis (. . .) in order to maintain the readability and cohesiveness of final synopsis.

4. EXPERIMENTS AND RESULTS

We conducted a user study to evaluate the effectiveness and utility of our approach. The subjects (5 graduate students not involved with the project) were asked to evaluate and rate the generated synopses on a scale of 0 to 10, based on how helpful these were in understanding the corresponding document-elements.

4.1 Determining the Penalty parameter, P

In order to determine the value of P that optimizes user satisfaction, we generated synopses for 43 document-elements selected randomly from different scientific documents at different values of P. The subjects were asked to rate all the generated synopses and the average scores for all the synopses at different P values were computed. The average length of synopses (in number of sentences) and average scores for different P values are tabulated in Table 1 and Figure 3 shows the variation of average scores with P.

Value of P	Average Synopsis Length	Average Score
0.1	13.31	2.12
0.2	8.17	3.35
0.3	6.31	5.47
0.4	5.10	6.74
0.5	**4.24**	**7.30**
0.6	3.70	6.70
0.7	3.16	5.93
0.8	3.01	5.88

Table 1: Average length(in number of sentences) and average scores of generated synopses for different P values.

It is observed that the effectiveness and usefulness of the generated synopses depends heavily on the synopses size. As

we can see from Table 1, the average length of generated synopses decreases as we increase the value of P. Further, as the value of P is increased, the average score first increases and then decreases, achieving a maximum value of 0.5. This is explained by the fact that the synopses generated at lower values of P are quite long and hence are not preferred by the users. These longer synopses take a lot of time to read and have reduced readability and effectiveness. The average scores increase as we increase the value of P indicating that the shorter and concise synopses are preferred by the users as they can find and comprehend the relevant information quickly. However, after a certain point ($P = 0.5$), the generated synopses are too short and prove insufficient to provide the necessary information to users and are thus, assigned lower scores.

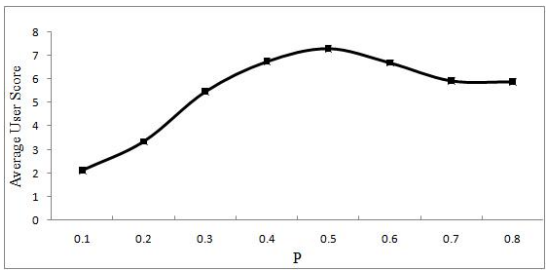

Figure 3: Variation in Average Score of Synopses with P.

4.2 Comparison with other methods

The aim of this experiment was to compare the proposed approach with state of the art methods and investigate and demonstrate the utility of synopses for document-element search engines. For this, we randomly selected 100 document-elements from different scientific publications and generated synopses by our method and following two methods:

1. **Google Desktop:** It is the desktop version of the most widely used commercial search engine Google and is used for searching documents stored on the user's desktop (http://desktop.google.com/). Along with the search results, it also provides *Query Biased* snippets to facilitate the search process. Though the exact algorithms used by it are unpublished, they are supposed to represent the state of art. We stored all the test documents on our desktop and then queried the desktop search engine with the same query formulated by extracting keywords from the caption and reference sentence as described in Algorithm 1. The synopses in this case are the *query biased* snippets accompanying the corresponding documents returned as search results.

2. **Caption and Reference Sentence:** Search engines like TableSeer and BioText Search Engine return the search results along with the caption and reference sentences for corresponding document-elements. Therefore, for this case, the synopses were generated for all the test cases by extracting the corresponding caption and reference sentences from the document text.

The subjects were then asked to rate all the synopses on a scale of 0–10 as before and the results are summarized in Table 2. The proposed method emerges as a clear winner when compared to the other two methods. We observed that the synopses produced by Google Desktop were of fixed length (2 lines) and were created by extracting portions of the text containing the keywords. Such incomplete and inadequate information is incapable to explain the information contained in the figure as is evident from the user assigned scores. The use of *document-element specific* information by TableSeer and BioText Search Engine results in much better synopses but since the information related to document-elements is generally spread throughout the document text, these also proved to be insufficient for providing enough information about the document-elements. On the other hand, our method utilizes the information cues present in the caption and reference text to isolate those sentences from the document text that describe the document-elements and hence, the resulting synopses were assigned much higher scores by the users as compared to the other two methods. The superiority of the proposed method, as evident from the results, shows the inadequacy of current state-of-the-art techniques to provide sufficient information for understanding document-elements and hence, corroborates the need for synopsis generation for document-elements.

Methods	Average Score
Google Desktop	1.94±1.52
Caption and Reference Sentence	4.69±1.96
Proposed Method	**7.27±1.36**

Table 2: Average scores for synopses generated by three methods.

5. CONCLUSIONS

The present work identified the problem of generating synopses for document-elements like tables and figures in digital documents. The proposed algorithm generates synopses by ranking sentences on the basis of their relevance to the document element and proximity to reference sentences. The algorithm then determines which sentences to include in the description, balancing the information content and length of the description so that the generated descriptions are both effective and useful. The usefulness of proposed approached is confirmed by a user study. Our future work would include developing more features to improve the quality of generated synopses and to investigate the use of synopses for improved document search.

6. REFERENCES

[1] Carbonell, Jaime and Goldstein, Jade. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, pages 335–336, 1998.

[2] Y. Chen, G. Wang, and S. Dong. Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, 24(12):1845–1855, 2003.

[3] R. P. Futrelle. Summarization of diagrams in documents. *Advances in Automated Text Summarization*, pages 403–421, 1999.

[4] M. A. Hearst, A. Divoli, H. Guturu, A. Ksikes, P. Nakov, M. A. Wooldridge, and J. Ye. Biotext search engine: beyond abstract search. *Bioinformatics*, 23(16):2196–2197, 2007.

[5] S. Kataria, W. Browuer, P. Mitra, and C. L. Giles. Automatic extraction of data points and text blocks from 2-dimensional plots in digital documents. In *AAAI*, pages 1169–1174, 2008.

[6] Y. Liu, K. Bai, P. Mitra, and C. L. Giles. Tableseer: automatic table metadata extraction and searching in digital libraries. In *JCDL*, pages 91–100. ACM, 2007.

[7] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-beaulieu, and M. Gatford. Okapi at trec-3. pages 109–126, 1995.