Generating Synopses for Document-Element Search

Sumit Bhatia[†], Shibamouli Lahiri[†] and Prasenjit Mitra[†*]
[†]Department of Computer Science and Engineering, [*]College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA-16802, USA
{sumit,shibamouli}@cse.psu.edu, pmitra@ist.psu.edu

ABSTRACT

Scientists often search for document-elements like tables, figures, or algorithm pseudo-codes. Domain scientists and researchers report important data, results and algorithms using these document-elements; readers want to compare the reported results with their findings. Some document-element search engines have been proposed (especially to search for tables and figures) to make this task easier. While searching for document-elements today, the end-user is presented with the caption of the document-element and a sentence in the document text that refers to the document-element. Oftentimes, the caption and the reference text do not contain enough information to interpret the document-element. In this paper, we present the first set of methods to extract this useful information (synopsis) related to document-elements automatically. We also investigate the problem of choosing the optimum synopsis-size that strikes a balance between information content and size of the generated synopses.

Categories and Subject Descriptors

H.3.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval; H.5.2 [**INFORMATION INTERFACES AND PRESENTATION**]: User Interfaces

General Terms

Algorithms, Experimentation.

Keywords

classification, document-element, summarization, synopses.

1. INTRODUCTION

Authors use document-elements for a variety of purposes like reporting and summarizing experimental results (plots, tables), describing a process (flow charts) or presenting an algorithm (pseudocode). 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. 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 document-elements. CiteSeerX[1], a major computer science digital library, has introduced a table search feature in addition to normal document search. Likewise, a specialized search engine for biology documents, *Bio-Text Search Engine*, offers capability to search for figures and tables in the documents[3].

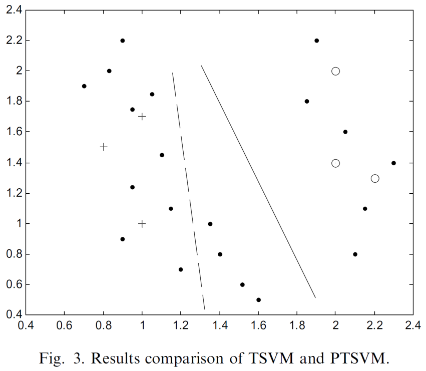


Figure 1: A sample figure and its caption. Figure is taken from[1].

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 is available in the snippets because he or she can not always interpret the information content of document-elements by examining just the snippet as illustrated by Figure 1. Even though the associated caption and legend help in understanding the information presented in a figure, they hardly provide enough details to fully understand and interpret them.

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

Extracting a synopsis for a document-element from a digital document involves filtering information related to the document-element

CIKM'09, November 2–6, 2009, Hong Kong, China.

[1] http://www.citeseerx.ist.psu.edu

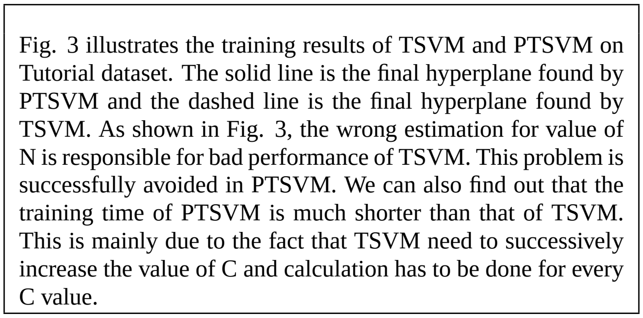


Fig. 3 illustrates the training results of TSVM and PTSVM on Tutorial dataset. The solid line is the final hyperplane found by PTSVM and the dashed line is the final hyperplane found by TSVM. As shown in Fig. 3, the wrong estimation for value of N is responsible for bad performance of TSVM. This problem 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 calculation has to be done for every C value.

Figure 2: Information extracted by our method for the figure described in Figure 1.

from the rest of the information contained in the document. Solving this problem accurately is easy if we understand the semantics of the text automatically. However, state-of-the-art techniques of natural language processing and statistical text processing still fall short in fully understanding the semantics of text documents. Additionally, good synopsis generation involves making a judgment call regarding the level of detail that may be useful to an end-user. If we generate a very large synopsis, it will be comprehensive, but the users' needs of finding information *quickly* will not be met. If we generate a very short synopsis, the user will not understand the document-elements clearly. We aim at striking a balance between these conflicting needs using automated synopsis-generation methods.

Previous research on document-elements has focused on knowledge extraction [5, 4] and developing techniques for document-element search [7, 3]. Futrelle introduces the idea of diagram summarization and explores various related issues and problems [2]. However, none of these work addressed the problem of actually *summarizing* a document-element and to provide related textual information that may help the user in the relevance judgment of a particular table or figure. In the present work we propose a method for extracting document-element related information from digital documents automatically. We adopt machine learning techniques and develop a novel feature set for identifying document-element related sentences. We also propose a simple model for sentence selection that tries to strike a balance between the information content and length of the synopsis.

2. IDENTIFYING DOCUMENT-ELEMENT RELATED INFORMATION

In this section we describe the strategies for automatically identifying document-element related information. We treat this problem as a classification task - each sentence is either relevant or non-relevant for a document-element.

2.1 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. The next step in the process involves parsing the document-element captions. Captions contain useful information cues that help understand the content of a document-element. A well-framed caption shows the purpose of the document-element. In order to deal with variations in caption format across different domains and writing styles, we propose the 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⟩

The CAPTION non-terminal in this grammar has 4 sub-parts. DOC_EL_TYPE specifies the type of the document element that can be a figure, a table or an algorithm. FIG_TYPE, TABLE_TYPE and ALGO_TYPE refer to the variations of the words "Figure", "Table" and "Algorithm" respectively. The DOC_EL_TYPE non-terminal is followed by an integer that represents the document-element number and is used to track the corresponding elements and their reference sentences. The integer is followed by a DELIMITER that can again be either ":" or ".". The final non-terminal TEXT gives a textual description of the element.

Although captions provide some details about the element of interest, we have to analyze the running text also in order to get complete understanding of the content and context of the document element under consideration [2]. Assuming good writing style, we hope to find at least one explicit reference to a particular document-element that can reveal certain details about the element. To identify such reference sentences, we use a grammar similar to the one used for caption parsing. Note that in the reference sentence, the delimiter will not be present in most cases and the integer will tell us which element this sentence is referencing to.

2.2 Feature Extraction

In this section we describe features that try to capture how well a sentence describes the content and context information of a document-element.

2.2.1 Content based Features

1. **Similarity with Caption (CapSYM):** This feature utilizes the information cues present in the caption. A query generated from the caption is used to assign a similarity score to each sentence in the document based on its similarity with the caption. We adapt Okapi BM25[9] as our similarity measure. It is defined as follows:

   If q is the generated query then the BM25 score of sentence s in document $\mathcal{D}$ is computed as:

$$BM25(q,s) = \sum_{t \in q} \left\{ \log \frac{N}{sf_t} . \frac{(k_1+1)tf_{ts}}{k_1((1-b)+b(\frac{l_s}{l_{av}}))+tf_{ts}} . \frac{(k_3+1)tf_{tq}}{k_3+tf_{tq}} \right\} \tag{1}$$

   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,
   tf_{tq} is the frequency of term t in query q,
   l_s is the length of sentence s,
   l_{av} is the average length of sentences in D,
   k_1, k_3 and b are constants which are set to 2, 2 and .75 respectively.

   After computing scores for all sentences, top 20 sentences with highest scores are selected and assigned a feature value of 1. All other sentences are assigned a feature value of 0.

2. **Similarity with Reference Sentence (RefSYM):** To utilize the information cues present in the reference sentences we compute their similarity scores with all the other sentences as described above. The top 20 highest scoring sentences are assigned a feature value 1 while all other sentences get a feature value 0.

3. **Cue Words and Phrases (CP):** Certain cue words and phrases are used frequently by authors while describing a document-element. For example, “shows”, “illustrates”, “slope”, “distribution”, “column” etc. A list of about 140 such words and phrases was created by manual inspection of the data set. A sentence with one or more cue words/phrases was assigned a feature value of 1. All other sentences were assigned a feature value 0.

2.2.2 *Context based features*

The features described above assume all sentences to be equally important. This assumption, however, is not true as generally when a document-element is referenced in the running text, the nearby sentences also relate to the *document-element* and become *“contextually”* more important than the other sentences. These *“nearby”* sentences provide the “context” in which a document-element is being described or used. We use the following features to identify and capture these contextually important sentences:

1. **IfReference Sentence (IfRefSent):** It is a binary feature with value 1 if a sentence is a reference sentence for the document-element. Otherwise, it has value 0.

2. **Paragraph Location (IsInSamePara):** It is again a binary feature and has a value 1 if a sentence belongs to the same paragraph as the reference sentence. Otherwise, the value is 0.

3. **Proximity:** This feature captures the fact that a sentence closer to the reference sentence has a higher probability of being related to the document-element than a sentence located far away from the reference sentence. For this, the first ten sentences on either side of a reference sentence are assigned a feature value of 1. All other sentences are assigned a feature value of 0.

2.3 Classification

We use Naïve-Bayes classifier for identifying document-element related sentences. Naïve-Bayes classifier has been previously used successfully for sentence extraction task for document summarization[6, 10] and is defined as follows:

Let the set of sentences that are related to the document-element d be $\mathcal{S}_d$ and let $\mathcal{S}$ be the set of all sentences in the document $\mathcal{D}$. Given the features $F_1, F_2, ..., F_n$ for sentence $s \in \mathcal{S}$, application of Bayes' rule assuming independent features yields the probability that s also belongs to $\mathcal{S}_d$, as follows:

$$P(s \in \mathcal{S}_d \mid F_1, F_2, ..., F_n) = \frac{\prod_{i=1}^{n} P(F_i \mid s \in \mathcal{S}_d) P(s \in \mathcal{S}_d)}{\prod_{i=1}^{n} P(F_i)} \quad (2)$$

The probabilities $P(F_i \mid s \in \mathcal{S}_d)$ and $P(F_i)$ are not known *a priori* but they can be estimated by counting occurrences in the training set. This gives a simple Bayesian classification function that assigns a probability score to each sentence in the document. The top-scoring sentences can be identified as related to document-elements. Note that $P(s \in \mathcal{S}_d)$ is same for all sentences in the document and is therefore a constant.

3. SENTENCE SELECTION - DETERMINING OPTIMAL SYNOPSIS SIZE

After identifying the document-element related sentences, we need to decide how many and what sentences to include in the synopsis that will be presented to the user. Presenting all the relevant sentences to the user might have a detrimental effect on the *readability* of the synopsis. A longer synopsis might be comprehensive, but it requires more time to read and understand, thereby defeating the whole purpose of making search results more user-friendly. It is therefore required to determine an optimum synopsis size that balances the trade-off between *information content* and *readability and effectiveness* of the synopsis.

In general, the sentence selection problem can be framed as follows: let U_k be the *Utility* measure of sentence s_k that tells us whether it is useful to select the sentence or not. Let the score of k^{th} sentence be $score_k$ and let all sentences be ranked in decreasing order of their scores so that *i* < *j* implies $score_i \geq score_j$. We define the *Utility* measure U_k as:

$$U_k = score_k - (1 - exp^{-\lambda(k-1)}) \quad (3)$$

We include a sentence in the synopsis if and only if its utility is greater than zero. Here Utility of a sentence is determined by two competing factors – (a) Relevance of the sentence to the document-element which is measured by the score of the sentence; (b) Penalty incurred by having an additional sentence s_k in the synopsis. λ is the *Penalty Parameter* that controls the magnitude by which sentences are being penalized and thus, determines the length of the synopses.

The final set of selected sentences is arranged in the order in which they appear in the document. Non-consecutive sentences are separated by ellipsis (. . .) to maintain readability and cohesiveness of the synopsis.

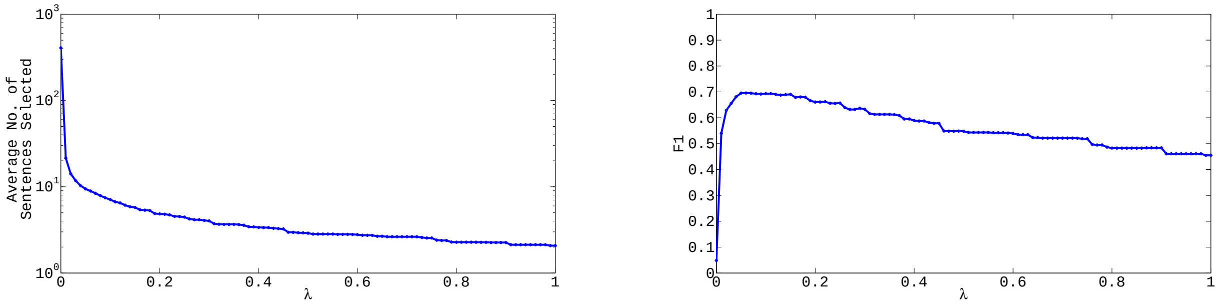
4. EXPERIMENTS AND RESULTS

In this section, we evaluate the effectiveness of the proposed method for extracting the document-element related information. For our experiments, we randomly selected 140 document-elements from different Computer Science publications. For each document-element, the relevant sentences were manually identified from the associated document by two human evaluators and were assigned a label 1. All other sentences in the associated document were assigned a label -1.

4.1 Relevant Sentence Identification

The aim of this experiment is to evaluate how well the proposed methods are able to identify the document-element related sentences. If the model learned is reasonable, then the sentences with a higher probability score are more relevant than sentences with a lower score. Thus, we get a ranked list of relevant sentences for each document-element. As discussed by Kanungo and Metzler, R-precision is more appropriate than using precision at a fixed value for sentence selection task because for different <document-element, document> pairs, the value of R is different and ideally, we want to return only these R relevant sentences[8].

We use 5-fold cross validation for evaluation. Table 1 reports the Precision values at N averaged over five validations where N = {1,2,3,4,5} and also the R-precision values. Note that here R is different for different document-elements. It can be observed that the performance of the proposed method is appreciable for the sentence extraction task. High precision values at top ranks indicate that the scores assigned on the basis of learned models are good indicators of the relevance of sentences to document-elements.



(a) No. of Sentences with λ

(b) F1 with λ

Figure 3: Effect of penalty parameter λ on (a) Average No. of Sentences Selected and (b) F1 Measure.

	Naïve Bayes
P@1	0.9572
P@2	0.9357
P@3	0.8857
P@4	0.8250
P@5	0.7871
R Precision	0.7387

Table 1: Different precision values for the sentence extraction task.

4.2 Sentence subset selection

In this section we evaluate our proposed sentence selection strategy to select a subset of top-ranking sentences that should be included in the final synopsis to be presented to the user. The penalty parameter λ, as defined in equation 3, controls the length of generated synopses by penalizing the inclusion of additional sentences in the synopses. In order to study the behavior of generated synopses with varying λ, we generated synopses for different values of λ, varying from 0 to 1, with increments of 0.01. For each value of λ, we compute the average length of synopses (in number of sentences) and F1 measure. The results are summarized in Figure 3. Note that the variation of average length of synopses is shown on a log scale.

From the figure, we observe that the average length of synopses decreases with increasing λ. For very small values of λ, almost no penalty is being incurred by inclusion of additional sentences. The model tries to maximize the *information content* and we end up with pretty long synopses. As we increase λ, the amount of penalty also increases and the less relevant sentences are being filtered out. For very high values of λ, the model favors highly concise synopses. The F1 measure, which considers both the precision and recall values simultaneously, follow an interesting trend. It first increases with increasing λ, achieves a maximum at $\lambda = 0.06$ and then gradually falls. The F1 values remain stable in the range 0.61 – 0.69 for λ = 0.05 – .35. The average synopses length in the same range lies in between 3.6 to 9.4. Here, the use of penalty parameter λ provides us with a simple but powerful means of generating variable length synopses as per the user needs. Initially, using a moderate value of λ (say 0.3), we can provide a concise and highly informative synopsis. Then, if the user wishes to know more about the document-element, synopses generated with lower values of λ can be presented (using relevant feedback techniques).

5. CONCLUSIONS AND FUTURE WORK

The present work identified the problem of generating synopses for document-elements like tables and figures in digital documents. Machine learning techniques are used to identify relevant sentences from the document text using a novel set of features that utilizes content and context information related to document-elements. A simple model is proposed to determine which sentences to include in the final synopsis. The model tries to balance the information content and length of the description so that the generated synopses are both effective and useful. 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 and document summarization.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 0535656 and 0845487.

7. REFERENCES

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

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

[3] 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.

[4] W. Huang, C. L. Tan, and W. K. Leow. Associating text and graphics for scientific chart understanding. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 580–584, Washington, DC, USA, 2005. IEEE Computer Society.

[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] J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 68–73, New York, NY, USA, July 1995. ACM Press.

[7] 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.

[8] D. Metzler and T. Kanungo. Machine learned sentence selection strategies for query-biased summarization. In *Proceedings of SIGIR Learning to Rank Workshop*, 2008.

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

[10] S. Teufel and M. Moens. Sentence extraction as a classification task. In *Workshop on Intelligent and Scalable Text summarization, ACL/EACL*, 1997.