Comments-Oriented Document Summarization: Understanding Documents with Readers' Feedback

Meishan Hu, Aixin Sun, and Ee-Peng Lim
School of Computer Engineering
Nanyang Technological University, Singapore
{hu0004an,axsun,aseplim}@ntu.edu.sg

ABSTRACT

Comments left by readers on Web documents contain valuable information that can be utilized in different information retrieval tasks including document search, visualization, and summarization. In this paper, we study the problem of comments-oriented document summarization and aim to summarize a Web document (e.g., a blog post) by considering not only its content, but also the comments left by its readers. We identify three relations (namely, *topic*, *quotation*, and *mention*) by which comments can be linked to one another, and model the relations in three graphs. The importance of each comment is then scored by: (i) *graph-based method*, where the three graphs are merged into a multi-relation graph; (ii) *tensor-based method*, where the three graphs are used to construct a 3rd-order tensor. To generate a comments-oriented summary, we extract sentences from the given Web document using either *feature-biased approach* or *uniform-document approach*. The former scores sentences to bias keywords derived from comments; while the latter scores sentences uniformly with comments. In our experiments using a set of blog posts with manually labeled sentences, our proposed summarization methods utilizing comments showed significant improvement over those not using comments. The methods using feature-biased sentence extraction approach were observed to outperform that using uniform-document approach.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: Abstracting methods; H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Experimentation

Keywords

Document Summarization, Blog, Comments, Graph-based Scoring, Tensor-based Scoring

SIGIR'08, July 20–24, 2008, Singapore.

1. INTRODUCTION

1.1 Motivation

Document summarization has always been an important research topic in Information Retrieval (IR). Most existing document summarization tasks take either one document or multiple topically related documents as input, and generate a short document containing the main topics covered by the input document(s) [11, 19, 21]. The resultant summary is therefore determined by the document content provided by the author(s). With the popularity of social websites (e.g., blogs), many Web documents are now presented together with annotations given by their readers in the form of tags, comments, ratings, and others. These user generated annotations are valuable input from the readers and can be utilized in different IR tasks. For instance, tags were shown to improve Web search in [2]. In this paper, we focus on comments-oriented document summarization and aim to summarize a Web document using not only its content, but also the comments contributed by its readers.

The comments-oriented document summarization task to produce a concise document covering the topics presented in the document that are discussed among readers who gave the comments is interesting for at least three reasons. First, despite their informal tone and style of writing, comments represent readers' understanding or feedback about a Web document's content. By considering these comments, the generated summary can better capture the input from the readers, as opposed to the author of the document only. That is, a comments-oriented summary provides balanced views from both author and readers. Second, most websites present a Web document (e.g., blog post) together with its comments. In a separate study on blog conversation, it was found that readers treat comments associated with a post as an inherent part of the post [7]. A comments-oriented summary hence better matches one's understanding of the document as readers often read the document together with its comments. Third, the generated summary could better support many IR applications. One example application is blog search. Many existing blog search engines rank results by recency of posts [20]. A post is ranked at the top because of its recency and/or its containing of a query keyword. With comments-oriented summary, a post can be ranked high if the query keyword is relevant to the main topic of the post identified by its readers through comments.

1.2 Overview and Contributions

In this research, we aim to generate comments-oriented summary in the form of extracted sentences from a given

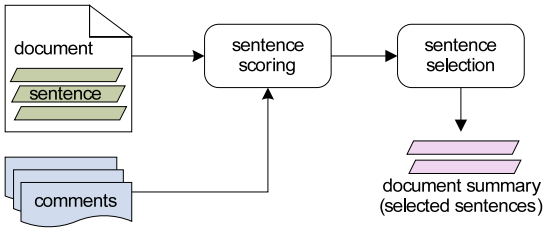


Figure 1: Comments-oriented summarization

Web document. As shown in Figure 1, we view the given document as a set of sentences and score them with input from both the document and its comments. A subset of sentences is then selected to form the summary satisfying the length requirement. The main focus is *sentence scoring*, which is expected to deal with two challenges. Firstly, the number of comments varies significantly from one document to another. In the extreme case, a document may not receive any comment. In this case, we may have to fall back to identifying important sentences by the document content only. Secondly, comments are inherently informal and noisy. Many comments may contain information irrelevant to the Web document content.

Depending on the way comments are utilized in sentence scoring, our proposed summarization techniques are classified into either *feature-biased approach* or *uniform-document approach*. In feature-biased approach, we treat comments-oriented summarization task as query-biased summarization task, where the query are those keywords derived from the comments. Determining the importance of comments (and hence the words appearing in them) is therefore the key of feature-biased approach. In uniform-document approach, we form a virtual document from the given document and its comments such that it consists of a set of text units. Here, a text unit is either a sentence from the document or a comment. The summary is composed by those highly scored text units that are sentences from the original document. Many techniques developed for ranking sentences in single document summarization could therefore be extended to address comments-oriented summarization using uniform-document approach. In particular, we show in this paper that the techniques for ranking comments (i.e., in the feature-biased approach) can be easily extended to score text units.

To score comments, we model the relationships among them using three graphs, namely, *topic*, *quotation* and *mention* graphs. Given the three graphs, two techniques are proposed in this paper. One is to merge them into one multi-relation graph and perform graph-based scoring using random walk algorithm [9, 19]. Another is to construct a 3rd-order tensor and score the comments using tensor decomposition. To the best of our knowledge, this is the first effort to bring tensor-based analysis into document summarization. With the combinations of two scoring techniques and two summarization approaches, we compare the four methods in our experiments using manually labeled documents.

Our major contributions in this research are as follows. Firstly, we propose two approaches to address comments-oriented document summarization. In the feature-biased approach, words appearing in comments but not in the given document do not contribute to scoring sentences. The approach is more tolerant to noise in comments. In the uniform-document approach, when there is no or very few comments, the problem naturally degrades to single document summarization. Secondly, we introduce tensor-based scoring to score comments (or sentences) in document summarization. Compared to graph-based scoring, tensor-based scoring can directly deal with multiple relations among comments (or sentences) that may also be presented in other document summarization problems besides comments-oriented summarization.

1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2, we review the related studies. The relationships among comments are given in Section 3. The graph-based and tensor-based scoring algorithms are proposed in Section 4, followed by the two approaches for comments-oriented summarization in Section 5. Our experiments are reported in Section 6. We conclude the paper in Section 7.

2. RELATED WORK

To generate an extractive summary, sentences in the document(s) are to be scored according to their salience in representing the major topic(s) of the document(s). Techniques proposed in literature that aim to measure the salience of sentences can be broadly categorized into three groups: lexical chain based methods [3, 25], feature-based methods [11, 21] and graph-based methods [9, 19, 23]. In lexical chain based methods, a lexical-chain is defined by a coherent sequence of related nouns, verbs, etc., computed based on a thesaurus such as WordNet. Sentences are then scored according to the lexical chains they belong to. In feature-based methods, each sentence is scored based on some features including position, length, cue phrases, signature words, etc. In graph-based methods, a sentence sharing similar content with another is considered as a semantic recommendation to the latter. PageRank [4] like algorithms are used to score sentences in a graph constructed through the semantic affinity among sentences. It was also shown that graph-based method can improve single-document summarization by incorporating multiple documents of the same topic [23].

Recently, Web document summarization has gained interest from many researchers. Based on the assumption that queries related to a Web page provide some human understanding about that page, Sun *et al* proposed to summarize Web pages using clickthrough data in [22]. In their work, a word was weighted by a linear combination of its term frequency in the page and its term frequency derived from the page's clickthrough record. Both LSA (Latent Semantic Analysis)-based and Luhn's methods [17] were applied to select sentences from Web pages and the two methods achieved comparable performance. The proposed approach is similar to our feature-biased approach as both involve extra knowledge to the document to be summarized. Nevertheless, comments contributed by readers on a Web document are quite different in nature from clickthrough record generated by a search engine.

Summarization of blog post, a new type of textual content on the Web, has not been well studied. Zhou *et al* viewed a blog post as a summary of online news articles it linked to, with added personal opinions [24]. A summary of a blog post is generated by extracting sentences from the blog post that are relevant to its linked news articles. Us-

ing a similar technique presented in [18], one sentence that is most dissimilar to the linked articles is deleted from the post at each iteration, until any more deletion would result in a drop in similarity between the summary and the linked articles. Comments associated with blog posts were however not used.

Utilizing comments to extract sentences from Web documents is quite related to the work of identifying most commented sentences reported in [8]. Eight features of comments, such as *number of terms common to the post in the comment*, *number of sentences in the comment*, and so on, were identified to represent each comment using a feature vector. The comments were then clustered based on their feature vectors and human experts were involved to determine the relevance of each cluster. The possible relationships among comments were not considered.

Constructing a quotation graph for the purpose of summarization is also related to the work by Carenini *et al* who summarized email conversation using "clue words" obtained from quoted email fragments [5]. Emails were first split into fragments and a *fragment quotation graph* was then constructed from fragments for identifying the "clue words". Their fragment quotation graph is similar to our quotation graph constructed from comments (see Section 3.2) as both rely on quotation relation. However, there are two major differences between the two graphs. First, in their fragment quotation graph, each node is a fragment identified from emails; in our quotation graph, each node is a comment (not fragment). Second, in [5], sentences could be extracted from the fragment quotation graph as a part of the summary. In our approach no sentence from comments is extracted.

3. UNDERSTANDING COMMENTS

In this section, we first formally define the problem of comments-oriented document summarization and then discuss the possible relationships among comments.

3.1 Problem Definition

Given a document D consisting of a set of sentences $D = \{s_1, s_2, \ldots, s_n\}$, and a set of comments $\mathcal{C} = \{c_1, c_2, \ldots, c_\ell\}$ associated with D, the task of *comments-oriented document summarization* is to extract a subset of sentences from D, denoted by S_c ($S_c \subset D$), that best represents the topic(s) presented in D and discussed among its comments $\mathcal{C}$.

In the above definition, D could be a news article, a blog post, or any other Web document. The set of comments $\mathcal{C}$ generally refers to those short textual messages contributed by readers and attached to D. All our following discussions refer to this setting. Nevertheless, the definition does not restrict the form of "comments". A blog post can also be considered as a comment to its referenced post or news article. Note that, the task of comments-oriented document summarization degrades to single-document summarization when the given document is not associated with any comment.

3.2 Comments Relationships

Comments provide readers' feedback about a Web document and also contribute to the discussion of topics presented in the document. The linkages among them often represent the discussion flow. Understanding the linkages among comments is hence critical for comments-oriented summarization. Based on our observation, three relations

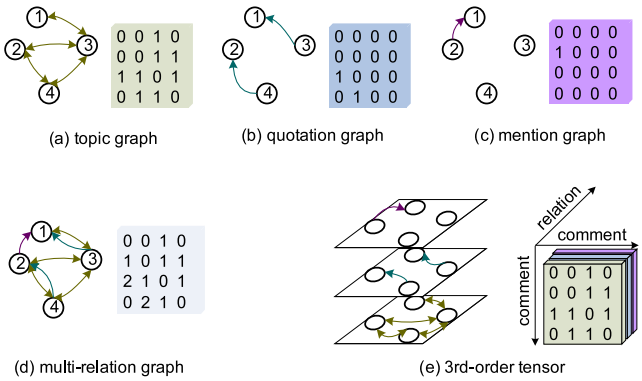


Figure 2: Model comments and relations

commonly exist among comments, linking one comment to others.

- *Topic relation.* Two comments are topically related if they talk about similar topic(s), often evidenced by sharing common words. The strength of the topic relationship can be measured by those commonly adopted metrics, such as *cosine similarity* and *Jaccard coefficient*. The topic relationship may be converted to binary weighted by comparing the strength to a predefined threshold (e.g., 0). Note that, the topic relationship between two comments are bi-directional.

- *Quotation relation.* Two comments are related through quotation if one quotes text segment(s) from the other. Quoting text segment is a strong indication that the current comment replies to the quoted comment or follows its discussion. Different from topic relationship, quotation relationship is binary and directional.

- *Mention relation.* If the contributor's name of an earlier published comment appears in a later published comment, the two comments are linked through mention. Here, we assume that comments are ordered by time. We consider mention as another type of indication that the current comment replies to the comment(s) left by the mentioned contributor[1]. Similar to quotation relationship, mention relationship is binary and directional.

Based on the above three relations, we derive three graphs, namely, *topic graph*, *quotation graph*, and *mention graph*. In each graph, the set of nodes are comments and the set of edges represent the particular relation. The weight associated with each edge is the strength of the corresponding relationship. Example of these graphs are given in Figures 2(a), (b), and (c), all involving the same set of four comments. The affinity matrix of each graph is shown on the right side of the graph. In these three graphs, all edges are binary weighted for clarity.

Compared to quotation and mention, topic relationship is more commonly found among comments. In most cases,

[1]If the mentioned contributor publishes several comments, all these comments are related to the mentioning comment through mention relation. Note that, mention could also occur in the passage of a quotation, as quotation is treated as part of a comment.

both quotation and mention graphs could be very sparse. However, two comments having very weak or even no topic relationship might be strongly related through mention or quotation. In other words, both quotation and mention relations complement topic relation in identifying the linkages among comments.

In some websites, comments are presented in a tree-like structure indicating *reply relationships* among them. As such reply relationship is website specific, we choose not to include it in our discussion. Nevertheless, our proposed techniques can be easily extended to include this relation (and other relations) if available.

4. SCORING COMMENTS

One important task in comments-oriented document summarization is to determine the importance of each comment in representing the discussed topic(s). Given the three graphs, we propose two methods to integrate the three graphs and score comments.

4.1 Graph-Based Scoring

A straightforward approach to integrate *topic graph*, *quotation graph*, and *mention graph* is to merge them into one multi-relation graph. In the merged graph, the weight of a directed edge between two comments is the total weights received from all the three graphs, as illustrated by Figure 2(d) together with the affinity matrix[2].

It is intuitive that important comments are those whose topics are discussed by a large number of other important comments. Based on this intuition, we propose to use PageRank [4] algorithm to score the comments (see Equation 1).

$$Score(c_i) = \alpha \cdot \frac{1}{|\mathcal{C}|} + (1-\alpha) \cdot \sum_{c_j} w(c_j, c_i) \cdot Score(c_j) \quad (1)$$

where α is the damping factor as in PageRank (in our experiments α=0.15). $|\mathcal{C}|$ denotes the number of comments associated with the given document, and $w(c_j, c_i)$ is the normalized weight from comment c_j to c_i derived from the multi-relation graph as shown in Equation 2.

$$w(c_j, c_i) = \frac{e(c_j, c_i)}{\sum_{c_k} e(c_j, c_k)} \quad (2)$$

In Equation 2, $e(c_j, c_k)$ is the weight on the edge from comment c_j to c_k in the multi-relation graph, which is the sum of the weights on the corresponding edges in the three graphs; $e(c_j, c_k) = 0$ if comments c_j and c_k are not related through any of the three relations.

4.2 Tensor-Based Scoring

Tensor provides a good means to represent multiple relations in one data structure. Given the three graphs, a 3rd-order tensor can be constructed, as shown in Figure 2(e). In this tensor, both its first and second dimensions (i.e., mode-1 and mode-2) are comments. The third dimension (i.e., mode-3) represents the relations through which each pair of comments are linked. The constructed tensor therefore captures all three relationships among comments.

Based on the tensor, we measure the importance of comments through tensor decomposition. There are two decomposition techniques, namely, High-order SVD (Singular Value Decomposition) and PARAFAC (PARAllel FACtor decomposition). The former leads to orthogonal singular vector(s) in each mode assuming that latent factors are independent of each other. The latter does not assume such independence and produces a number of parallel factors. Assuming topics discussed among comments are less independent from each other, we decompose the 3rd-order tensor in Figure 2(e) through PARAFAC decomposition, shown in Equation 3 and illustrated by Figure 3.

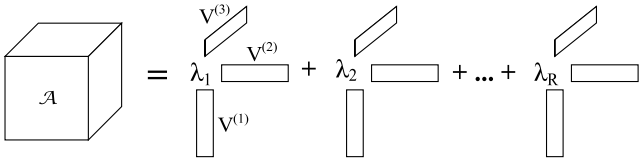


Figure 3: PARAFAC decomposition

$$\mathcal{A} = \sum_{r=1}^{R} \lambda_r \cdot V_r^{(1)} \circ V_r^{(2)} \circ V_r^{(3)} \quad (3)$$

In Equation 3, tensor $\mathcal{A}$ is decomposed into R parallel factors (see Section 6 on determining the value of R). λ_r ($1 \leq r \leq R$) is a scalar reflecting the salience of the corresponding factor, which is a topic discussed among comments in our setting. Each $V_r^{(n)}$ ($n = 1, 2, 3$) is a vector where values represent the salience of entries along mode-n with respect to factor r; and $\circ$ denotes outer product (see [6] for more details). In our setting, $V_r^{(1)}$ reflects the salience of comments in supporting topic r, $V_r^{(2)}$ reflects the salience of comments in representing topic r, and $V_r^{(3)}$ reflects the salience of relations in identifying topic r.

Based on the result of PARAFAC decomposition, we measure the importance of a comment c_i by the most salient topic it could best represent, as shown in Equation 4.

$$Score(c_i) = \max_{r \in R} \left(\lambda_r \times V_r^{(2)}(i) \right) \quad (4)$$

In this equation, $V_r^{(2)}(i)$ denotes the ith entry in vector $V_r^{(2)}$.

In both graph-based scoring or tensor-based scoring, the scores computed for comments are normalized in the range of $[0, 1]$ before they are used in other computations.

5. SUMMARIZATION WITH COMMENTS

We propose two approaches to incorporate comments into document summarization. The first approach scores document sentences based on keywords derived from comments; while the second approach scores document sentences and comments all together. The two comment scoring methods (i.e., graph-based scoring and tensor-based scoring) presented in Section 4 can be used in both approaches.

5.1 Feature-Biased Approach

In the feature-biased approach, the task of comments-oriented summarization is formulated as a query-biased document summarization problem where the queries are the keywords derived from comments.

With comments scored by their importance in representing the discussed topic(s), words appearing in many important comments are more topic representative. Thus, each

[2]Multiple edges between a pair of comments are shown solely for illustration purpose.

word derives its score by accumulating the scores of the comments it appears in, shown in Equation 5:

$$score(w_k) = \sum_{w_k \in c_i} score(c_i) \quad (5)$$

where $score(c_i)$ is the importance of comment c_i given by either graph-based scoring or tensor-based scoring; $w_k \in c_i$ denotes that word w_k appears in comment c_i.

Sentences in the document are scored according to their contained words. Specifically, every word in the document receives two weights, one for representing the topics discussed in the comments defined by Equation 5, and the other for representing the topics of the document. The latter is measured by the $tf \cdot idf$ value, where the document collection consists of all posts in a blog. The final weight of each word is the linear combination of the two weights after normalization, shown in Equation 6. In this Equation, β ($\beta > 0$) is a parameter to control the contribution of the weight received from comments. Note that, words in the document that do not appear in any comment receive 0 score from comments.

$$weight(w_k) = \frac{1}{1+\beta}\left(tf \cdot idf(w_k) + \beta \times score(w_k)\right) \quad (6)$$

We use density-based scoring to measure the importance of a sentence s in the given document [15].

$$Score(s) = \frac{K}{K-1} \sum_{k=1}^{K-1} \frac{weight(w_k) + weight(w_{k+1})}{distance(w_k, w_{k+1})^2} \quad (7)$$

where K is the total number of keywords (i.e., non-stopwords) in s; w_k and w_{k+1} are two adjacent keywords in s, and $distance(w_k, w_{k+1})$ denotes the distance between w_k and w_{k+1} in number of stopwords.

The comments-oriented summary is formed by extracting those top scored sentences. Note that, when there are very few comments associated with a document, the summary produced will be mainly based on the $tf \cdot idf$ values of the words contained in the document with density-based sentence scoring.

5.2 Uniform Document Approach

Through the three relations, i.e., topic, quotation, and mention, comments are linked together and modeled in either a multi-relation graph or a tensor. In uniform-document approach, we further extend topic and quotation relations to link comments to sentences from the given document. If a comment discusses a similar topic with a sentence, they are topically related. Similarly, a comment and a sentence are related through quotation if the comment quotes a text segment from the sentence.

With the extended relations, both the sentences from the document and the comments associated with the document are treated uniformly as *text units*. Based on our discussion in Section 4, these text units can be modeled in either a multi-relation graph or a tensor and scored with the corresponding scoring method. To generate a comments-oriented summary, those highly scored text units that are sentences from the original document are extracted to form the summary.

In uniform document approach, if a document is associated with very few or even no comment, the summary produced will be mainly based on the topic graph formed by the sentences (i.e., text units) in the document.

Table 1: Summarization methods

Scoring/Approach	Feature-biased	Uniform-doc
Graph-based	*FeatureGraph*	*DocGraph*
Tensor-based	*FeatureTensor*	*DocTensor*

Table 2: Dataset statistics

Average number of	
Sentences per post	22.22
Comments per post	26.04
Quotations among comments per post	2.65
Mentions among comments per post	17.18

6. EXPERIMENTS

Recall that both feature-biased and uniform-document approaches work with either graph-based scoring or tensor-based scoring. We have in total four comments-oriented document summarization methods, shown in Table 1. In this section, we evaluate these four methods with manually labeled documents.

6.1 Dataset and Performance Metric

Without existing benchmark dataset, we collected data from two blogs, Cosmic Variance[3] and IEBlog[4], both receiving large number of comments. From all posts collected, we randomly picked up 100 posts, 50 from each blog, to form our evaluation dataset. Table 2 gives the statistics on these 100 posts. To generate reference summaries, 4 human summarizers were asked to read both the posts and their corresponding comments and then label approximately 7 sentences[5] for each post[6]. The labeled set of sentences form the human generated summaries in our evaluation.

Two performance metrics, namely, *ROUGE* and *NDCG*, are used to evaluate the effectiveness of the proposed methods. ROUGE has been widely adopted for evaluating document summarization methods [16]. It evaluates the machine generated summary against human generated summary (labeled sentences in our setting) by counting overlapping units such as n-gram. We used ROUGE-1.5.5 package and report the F-measure of ROUGE-1 (i.e., unigram). We choose to report F-measure instead of recall as the human generated summaries are limited by number of sentences (not words). In our evaluation, for each document, each method returns the top 7 scored sentences to form the machine generated summary whose length (in number of sentences) matches human generated summary. The selected sentences are ordered according to their positions in the original document. The values reported are averaged over the 4 human generated summaries for the 100 blog posts.

Given a ranked list of retrieved documents with their relevance level in response to a query, NDCG (*Normalized Discounted Cumulative Gain*) [13] is computed through Equation 8:

$$NDCG = \frac{1}{Z} \cdot \sum_{p=1}^{K} \frac{2^{R(p)} - 1}{\log(1+p)} \quad (8)$$

[3] `http://cosmicvariance.com`

[4] `http://blogs.msdn.com/ie/`

[5] We fix the number of labeled sentences as "a constant summary length is more appropriate" [10].

[6] The user study was conducted in a similar manner to the one reported in [12], with more blog posts and more human summarizers involved.

Table 3: Different inputs for methods

Symbol	Method input
P	blog post only, no comments is given
PCt	topic relationship among text units
$PCtq$	topic and quotation relationships
$PCtm$	topic and mention relationships
$PCtqm$	topic, quotation, and mention relationships

where Z is a normalization factor derived from the perfect ranked list of K documents; $R(p)$ denotes the relevance level of document at rank position p. In our setting, each sentence in the extractive summary is an object to be ranked and the relevance level of each sentence is defined by the number of summarizers labeled it. For example, the relevance level of a sentence is 3 if three summarizers labeled the sentence and 0 if no summarizer labeled the sentence. In our evaluation, for a given document, K is also set to 7. The reported NDCG is averaged over 100 posts.

6.2 Methods

We evaluated the four methods listed in Table 1, namely, *FeatureGraph*, *DocGraph*, *FeatureTensor*, and *DocTensor*, with different inputs to simulate the cases where different relationships among comments are available. As shown in Table 3, P denotes that only the post is available to the summarization methods. The problem is analogous to single-document summarization. PCt denotes that both the post and the topic relationships among comments are available to each method. Similarly, $PCtq$, $PCtm$, and $PCtqm$ refer to the inputs consisting of the blog post and the corresponding relationships respectively. Specifically, with $PCtqm$, all the three relationships discussed in Section 3 are given to the summarization methods.

Different from quotation and mention that are binary (e.g., weighted by 1 or 0), the topic relationship between two comments needs to be measured. In our experiments, we used cosine similarity to measure the strength of topic relationship and evaluated two settings: *weighted* and *unweighted*. With weighted setting, the edges in the topic graph are weighted by the cosine similarity. With unweighted setting, every edge carries the same weight of 1 if the cosine similarity is greater than 0 as in [23]. Nevertheless, our experimental results showed that with unweighted topic graph, slightly better performance was achieved for almost all methods. For the sake of page space, we choose to report the results using unweighted topic graph only.

To perform the PARAFAC decomposition, we used Matlab Tensor Toolkit [1], where the number of factors (i.e., R) needs to be specified. As the number of salient factors in sentences and comments of a blog post is usually not known beforehand, R is simply set to be the number of text units (i.e., comments and/or sentences) to be scored.

6.3 Experimental Results

6.3.1 Method Comparison

In the first set of experiments, we compare the performance of the four methods with the five different inputs. For all methods with feature-biased approach we set $\beta = 2$ (see Equation 6). The impact of using different β's is further studied in the second set of experiments reported in Section 6.3.2.

The summarization accuracy by ROUGE-1 and NDCG are reported in Table 4 and plotted in Figure 4 for easy comparison. Given a particular input, e.g., $PCtqm$, the best performance is highlighted in bold in Table 4. From the results, the following observations can be made.

Firstly, for all four methods, much better performance were achieved when comments were used (i.e., PCt, $PCtq$, $PCtm$, or $PCtqm$), compared to using post only (i.e., P) according to both ROUGE-1 and NDCG. Almost all improvements are statistically significant ($p \leq 0.05$ based on paired t-test); the few non-significant ones are indicated by † in Table 4. Such results well support our hypothesis that comments contain valuable information for better document understanding.

Secondly, methods using feature-biased approach achieved better performance than those using uniform-document approach. According to ROUGE-1, FeatureGraph achieved significantly better performance than DocGraph with all inputs except P and $PCtq$ ($p \leq 0.05$, indicated by * in Table 4); FeatureTensor always significantly outperformed DocTensor. According to NDCG, FeatureGraph outperformed DocGraph significantly with all inputs except P, and FeatureTensor achieved significantly better performance than DocTensor with PCt, $PCtm$, and $PCtqm$. One possible reason for the worse performance of methods using uniform-document approach is that not all comments are quite relevant to the post due to noise. Using feature-biased approach, words contained in those noisy comments do not contribute to the sentence scoring as long as the words do not appear in the blog post. However, these comments might affect the sentence scoring for methods using uniform-document approach.

Thirdly, FeatureGraph and FeatureTensor performed comparably, with FeatureGraph being slightly better according to ROUGE-1, and FeatureTensor being slightly better according to NDCG. In specific, the best ROUGE-1 was achieved by FeatureGraph with $PCtqm$, and the best NDCG was achieved by FeatureTensor with $PCtqm$. In short, all the three relations considered in Section 3.2 could improve the comments-oriented summarization accuracy.

6.3.2 Impact of β to Feature-biased Approach

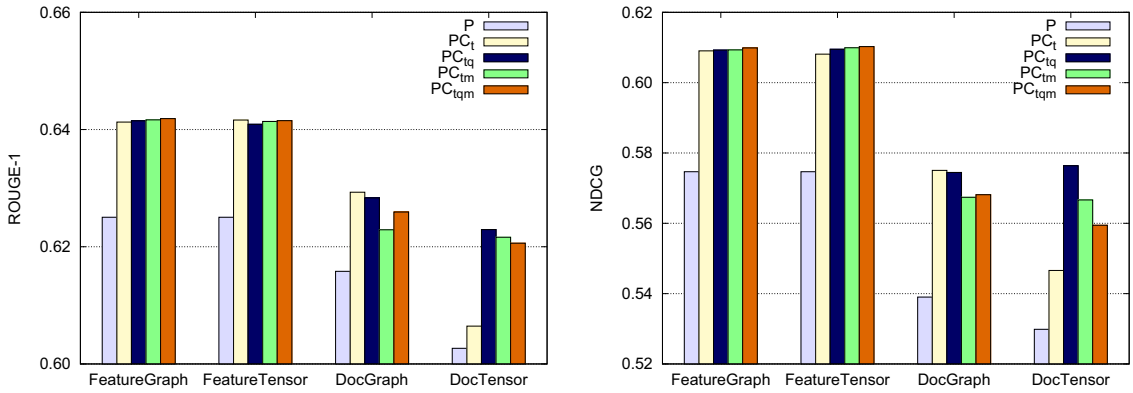
In this section, we study the impact of β to the methods using feature-biased approach, i.e., FeatureGraph and FeatureTensor. Recall that β is the coefficient involved in combining the two weights of a word received from the document and the comments respectively. The larger the β, the more emphasis is given to the weight received from comments (see Section 5.1). We varied β from 0 to 10 to observe its impact to the two methods.

The performance of FeatureGraph and FeatureTensor with different β values are reported in Figure 5. Note that, when $\beta = 0$, a word is weighted solely on blog post content, i.e., $tf \cdot idf$ value, and no comment is used in summarization.

As shown in Figure 5, when β is greater than 0, better summarization performances were observed for both methods compared to that with $\beta = 0$. This strongly suggests that incorporating comments benefits blog summarization. Evaluated by ROUGE-1, both FeatureGraph and FeatureTensor followed similar trend with different β values. Starting from 0, improvement on ROUGE-1 value was observed till $\beta = 2$ followed by small decrement till $\beta = 5$. Nevertheless, the decrement is not significant. Evaluated by NDCG, sum-

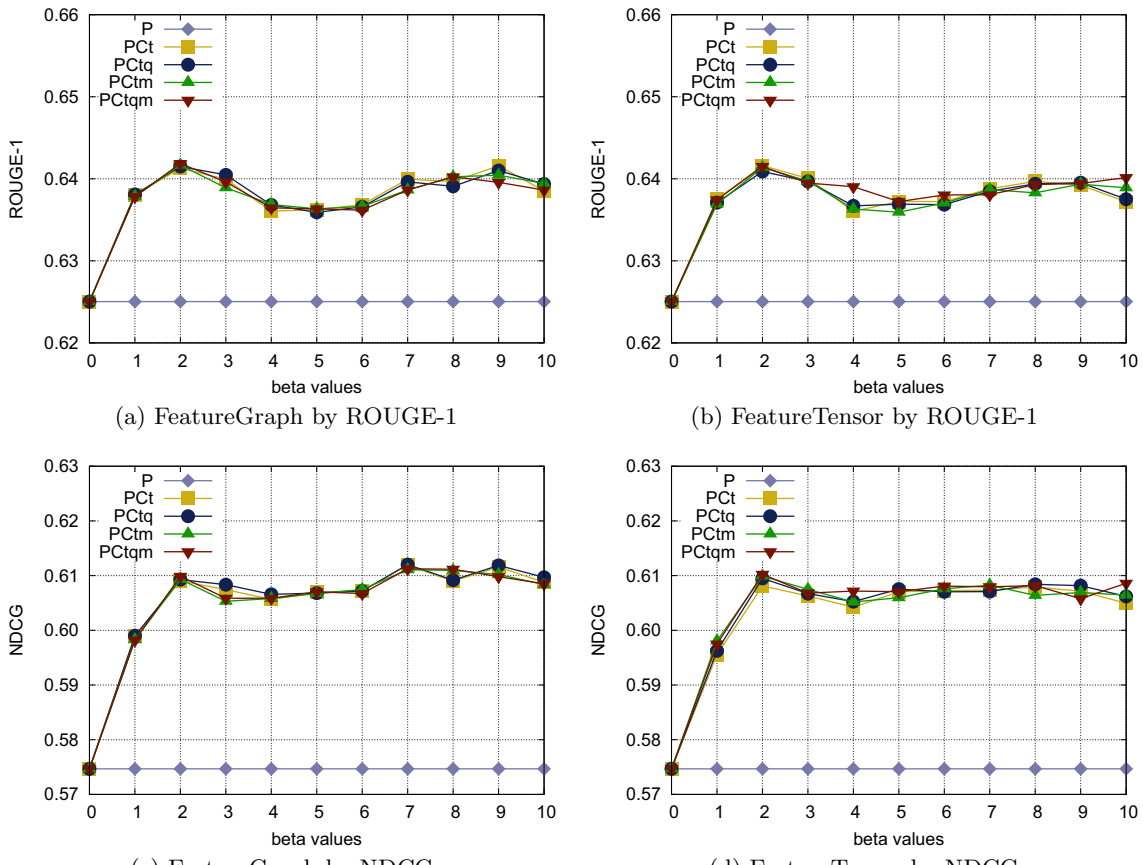
Table 4: Summarization accuracy by ROUGE-1 and NDCG

Method	ROUGE-1					NDCG				
	P	*PCt*	*PCtq*	*PCtm*	*PCtqm*	*P*	*PCt*	*PCtq*	*PCtm*	*PCtqm*
FeatureGraph	**0.6250**	0.6413*	**0.6415**	**0.6417***	**0.6419***	**0.5747**	**0.6090***	0.6093*	0.6093*	0.6099*
FeatureTensor	**0.6250***	**0.6416***	0.6409*	0.6414*	0.6415*	**0.5747**	0.6081*	**0.6095**	**0.6099***	**0.6102***
DocGraph	0.6158	0.6293	0.6283	0.6229†	0.6259†	0.5390	0.5751	0.5744	0.5674	0.5681
DocTensor	0.6027	0.6065†	0.6229	0.6216	0.6206	0.5298	0.5466†	0.5764	0.5666	0.5594



(a) ROUGE-1 (b) NDCG

Figure 4: Summarization accuracy (β=2)



(a) FeatureGraph by ROUGE-1 (b) FeatureTensor by ROUGE-1

(c) FeatureGraph by NDCG (d) FeatureTensor by NDCG

Figure 5: Impact of β to FeatureGraph and FeatureTensor

marization accuracy kept on increasing till $\beta = 2$. When β is larger than 2, the performance of the methods were less affected by the value of β.

In summary, based on this set of experiments, $\beta = 2$ is a reasonable setting in combining the two weights (derived from document and comments respectively) of a word.

6.3.3 Further Discussion

In our experiments, graph-based and tensor-based scoring methods achieved comparable summarization accuracies. In this section, we further discuss the two methods with respect to their computational cost and representation power.

Let N be the number of sentences and/or comments, M be the number of relations under modeling, R be the number of latent factors, and I be the number of iterations needed for convergence in the computation. Graph-based method has space complexity of $O(N^2)$ to store one transition matrix for its computation, while tensor-based method has space complexity of $O(MN^2)$. Nevertheless, $M \ll N$ in most cases, meaning that the two methods are comparable in space complexity. On the other hand, graph-based method has time complexity of $O(IN^2)$, while tensor-based method has time complexity of $O(IR^2(2N+M))$ (analyzed according to [14]), where $R = N$ in our case. In short, graph-based method is more computational efficient than tensor-based method.

However, tensor-based method has its power in representing multiple relations among the given set of objects and their relationships. In our summarization task, topic, quotation, and mention are three example relations that have been considered. Other relations, that link one sentence/comment to another, can be naturally incorporated into this method by extending the 3rd-mode of the tensor. However, in graph-based method, multiple relations are merged and their semantics are thus lost and unreconstructible.

7. CONCLUSION

Leaving comments on Web documents (or other Web objects) has become an important feature for many websites especially the social websites. Those comments contributed by readers provide valuable information to better understand the Web documents. In this paper, we studied comments-oriented document summarization that aims to generate an extractive summary for a Web document using comments contributed by its readers. We construct three graphs based on the three types of relationships among comments. Depending on the way the three graphs are merged into one data structure, two scoring methods known as graph-based scoring and tensor-based scoring are proposed to measure the importance of comments. We further propose two approaches to integrate comments into document summarization for generating comments-oriented document summaries. By varying input parameters and using a manually labeled set of blog posts, we evaluated four methods. Our experiment results suggest that including comments into the summarization improved summarization accuracy significantly.

8. REFERENCES

[1] B. W. Bader and T. G. Kolda. Algorithm 862: MATLAB tensor classes for fast algorithm prototyping. *ACM Trans. on Mathematical Software*, 32(4):635–653, December 2006.

[2] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing Web search using social annotations. In *Proc. of WWW'07*, pages 501–510, Banff, Alberta, Canada, 2007.

[3] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proc. of the Intelligent Scalable Text Summarization Workshop*, Madrid, Spain, 1997.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proc. of WWW'98*, pages 107–117, Brisbane, Australia, 1998.

[5] G. Carenini, R. T. Ng, and X. Zhou. Summarizing email conversations with clue words. In *Proc. of WWW'07*, pages 91–100, Banff, Alberta, Canada, 2007.

[6] L. De Lathauwer, B. D. Moor, and J. Vandewall. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[7] A. de Moor and L. Efimova. An argumentation analysis of weblog conversations. In *Proc. of Int'l Working Conf. on the Language-Action Perspective on Communication Modelling (LAP'04)*, New Brunswick, NJ, 2004.

[8] J.-Y. Delort. Identifying commented passages of documents using implicit hyperlinks. In *Proc. of HYPERTEXT'06*, pages 89–98, Odense, Denmark, 2006.

[9] G. Erken and D. R. Radev. LexPageRank: Prestige in multi-document text summarization. In *Proc. of EMNLP'04*, Barcelona, Spain, July 2004.

[10] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing text documents: Sentence selection and evaluation metrics. In *Proc. of SIGIR'99*, pages 121–128, 1999.

[11] E. Hovy and C.-Y. Lin. Automated text summarization and the SUMMARIST system. In *Proc. of a workshop on held at Baltimore, Maryland*, pages 197–214, Baltimore, Maryland, 1996.

[12] M. Hu, A. Sun, and E.-P. Lim. Comments-oriented blog summarization by sentence extraction. In *Proc. of CIKM '07*, pages 901–904, Lisboa, Portugal, 2007.

[13] K. Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proc. of SIGIR '00*, pages 41–48, Athens, Greece, 2000.

[14] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. Technical Report SAND2007-6702, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, November 2007.

[15] G. G. Lee, J. Seo, S. Lee, H. Jung, B.-H. Cho, C. Lee, B.-K. Kwak, J. Cha, D. Kim, J. An, H. Kim, and K. Kim. SiteQ: Engineering high performance qa system using lexico-semantic pattern matching and shallow nlp. In *Proc. of TREC'01*, pages 437–446, 2001.

[16] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proc. of NAACL'03*, pages 71–78, Edmonton, Canada, 2003.

[17] H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[18] D. Marcu. The automatic construction of large-scale corpora for summarization research. In *Proc. of SIGIR'99*, pages 137–144, Berkeley, California, USA, 1999.

[19] R. Mihalcea and P. Tarau. TextRank: Bringing order into text. In *Proc. of EMNLP'04*, pages 404–411, Barcelona, Spain, July 2004.

[20] G. Mishne. Information access challenges in the blogspace. In *Proc. of Int'l Workshop on Intelligent Information Access*, Helsinki, Finland, 2006.

[21] D. R. Radev, H. Jing, M. Stys;, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[22] J.-T. Sun, D. Shen, H.-J. Zeng, Q. Yang, Y. Lu, and Z. Chen. Web-page summarization using clickthrough data. In *Proc. of SIGIR'05*, pages 194–201, Salvador, Brazil, 2005.

[23] X. Wan and J. Yang. CollabSum: exploiting multiple document clustering for collaborative single document summarizations. In *Proc. of SIGIR'07*, pages 143–150, Amsterdam, The Netherlands, 2007.

[24] L. Zhou and E. Hovy. On the summarization of dynamically introduced information: Online discussions and blogs. In *Proc. of AAAI'06 Spring Symposium on Computational Approaches to Analyzing Weblogs*, March 2006.

[25] Q. Zhou, L. Sun, and J.-Y. Nie. IS_SUM: A multi-document summarizer based on document index graphic and lexical chains. In *DUC2005*, 2005.