Multi-topic based Query-oriented Summarization *

Jie Tang†, Limin Yao‡, and Dewei Chen§

Abstract

Query-oriented summarization aims at extracting an informative summary from a document collection for a given query. It is very useful to help users grasp the main information related to a query. Existing work can be mainly classified into two categories: supervised method and unsupervised method. The former requires training examples, which makes the method limited to predefined domains. While the latter usually utilizes clustering algorithms to find 'centered' sentences as the summary. However, the method does not consider the query information, thus the summarization is general about the document collection itself. Moreover, most of existing work assumes that documents related to the query only talks about one topic. Unfortunately, statistics show that a large portion of summarization tasks talk about multiple topics. In this paper, we try to break limitations of the existing methods and study a new setup of the problem of multi-topic based query-oriented summarization. We propose using a probabilistic approach to solve this problem. More specifically, we propose two strategies to incorporate the query information into a probabilistic model. Experimental results on two different genres of data show that our proposed approach can effectively extract a multi-topic summary from a document collection and the summarization performance is better than baseline methods. The approach is quite general and can be applied to many other mining tasks, for example product opinion analysis and question answering.

1 Introduction

Query-oriented summarization (QS) tries to extract a summary for a given query. It is a common task in many text mining applications. For example, a user submits a query to a search engine and the search engine usually returns a lot of result documents. To 'click-and-view' each of the returned documents is obviously tedious and infeasible in many cases. One challenging issue is how to help the user digest the returned documents. Typically, the documents talk about different perspectives of the query. An ideal solution might be that the system automatically generates a concise and informative summary for each perspective of the query. The user then can determine whether she/he needs to refine the query or zoom into a specific perspective. Another example is product opinion analysis. When a user asks for information about a product, e.g., "iPod touch", she/he does not typically mean to find documents containing these two words. Her/his intention is to find documents describing different features (e.g., price, color, size, and battery) of the product. An ideal result should be an informative summary, which organizes the information with different features.

Much work has been done for document(s) summarization. Generally, document(s) summarization can be classified into three categories: single document summarization (SDS), multi-document summarization (MDS), and Query-oriented summarization (QS). SDS is to extract a summary from a single document; while MDS is to extract a summary from multiple documents. The two tasks have been intensively investigated and many methods have been proposed [14] [3] [18]. The methods for document(s) summarization can be further categorized into two groups: unsupervised and supervised. The unsupervised method is mainly based on scoring sentences in the documents by combining a set of predefined features [17] [6]. In the supervised method, summarization is treated as a classification or a sequential labeling problem and the task is formalized as identifying whether a sentence should be included in the summary or not [25]. However, the method requires training examples.

Query-oriented summarization (QS) is different from the SDS and the MDS tasks. First the summary obtained in a QS task should be closely related to the query; while the summary by a SDS/MDS task should be about the main information of the document(s). Secondly, there might be multiple topical aspects presented in the related documents of a query; while documents in a SDS/MDS task are usually assumed to be about a same topic. Previously, query-oriented summarization has been introduced as a summarization task in DUC, which also tries to extract a summary for a given query. [1] Our QS task addressed in this paper is, relevant, but different from that of DUC. The main difference is that documents of each QS task in DUC is assumed to be related to one main topic. Is it true that documents in a practical QS task are related to only one topic? Unfortunately, the answer is, indeed, no and, statistics show that there are 36.85% of document clusters/sets talking about multiple topics (e.g.,

*The work is supported by the National Natural Science Foundation of China (60703059), Chinese National Key Foundation Research (2007CB310803), and Chinese Young Faculty Research Funding (20070003093). It is also supported by IBM Innovation funding.

†Dept. of Computer Science and Technology Tsinghua University.

‡Dept. of Computer Science, University of Massachusetts Amherst.

§Dept. of Computer Science and Technology Tsinghua University.

[1] http://www-nlpir.nist.gov/projects/duc/duc2005/tasks.html

multiple events or multiple persons) instead of a single topic, even in the well-organized DUC data set [19]. Another statistical study conducted by Sekine and Nobata [24] show that there are 44.62% of the document sets of Japanese news articles talking about multiple topics.

Therefore, two interesting research questions are: (1) how to capture the query information for extracting the summary from documents? (2) how to discover multiple topics hidden in the documents?

In this paper, we aim to conduct a thorough investigation on the problem of multi-topic based query-oriented summarization. We identify the major tasks of the problem and propose a probabilistic approach to solve the tasks. Specifically, we present a statistical topic model to discover multiple topics in a document collection. We study two strategies for incorporating the query information into the topic model. The first strategy directly integrates the query information into the generative process of the topic model. Thus the model estimates a mixture of a document-specific topic distribution and a query-specific topic distribution. The other strategy is to use a regularization form to constrain the topic model by the query information. The basic idea is to "guide" the topic model by the query-specific topics. Based on the modeling results, we study several scoring schemes to rank sentences in a document collection. We further refine the summarization results using redundancy reduction.

We conducted experiments on two different genres of data: DUC data and Epinions data (www.epinions.com). Experimental results on both of the two data sets show that our proposed method outperforms the baseline method of using word frequency (i.e., TF) and that of using existing topic models (including pLSI, and LDA).

The rest of this paper is organized as follows: Section 2 presents problem definitions. In Section 3 we give an overview of our proposed approach. In Section 4 we describe the two strategies for incorporating the query information into the probabilistic model. In Section 5 we present our method for generating the summary based on the modeling results. In Section 6, we give the experimental results and in Section 7, we present the related work. Finally, in Section 8, we conclude this paper with future work.

2 Problem Formulation

In this section, we first present several necessary definitions and then define the tasks of query-oriented summarization.

DEFINITION 2.1. ***(Document):*** *We define a* document d *as a sequence of N_d words, denoted as $\boldsymbol{w}_d$, where each word is chosen from a vocabulary of size V.*

DEFINITION 2.2. ***(Query):*** *A query q consists of multiple words, denoted as a vector $\boldsymbol{w}_q$, where each word in the query is also chosen from a vocabulary of size V.*

DEFINITION 2.3. ***(Document Cluster):*** *A document cluster, denoted as $C = \{\boldsymbol{w}_d | 1 \leq d \leq M\}$, contains M documents relevant to the query q. The relevance can be considered as either syntax relevance (containing words in the query) or semantic relevance (containing relevant information of the query).*

The document cluster denotes the information source and the query denotes the information need. A document cluster is a sub set of the entire document collection. All documents in the cluster are related to the query. Thus, we can define the task of query-oriented summarization as:

DEFINITION 2.4. ***(Query-oriented Summarization):*** *Given a document cluster C and a query q, the task of query-oriented summarization is to identify the most representative sentences for the query, from the document cluster.*

Documents related to a query may talk about different perspectives of the query. For example, for the query "data mining", the topical aspects may include "classification", "clustering", and "association rule". Accordingly, we give definitions of the topic model and the query-oriented topic model of a document cluster.

DEFINITION 2.5. ***(Topic model of Document Cluster):*** *A topic model θ of a document cluster C is a multinomial distribution of words $\{p(w|\theta)\}$ [21]. Each document cluster is considered as a mixture of multiple topic models. The assumption of this model is that words in the document are sampled following word distributions corresponding to each topic, i.e., $p(w|\theta)$. Therefore, words with the highest probability in the distribution would suggest the semantics represented by the topic.*

DEFINITION 2.6. ***(Query-oriented Topic Model):*** *Different from a general topic model, a query-oriented topic model includes the main topics related to the query. A query-oriented topic model can be described by two multinomial distributions of words: $\{p(w|\theta)\}$ and $\{p(w_q|\theta_q)\}$.*

For the example of "data mining", suppose the query is about data mining application, we may only want to highlight topics related to applications of data mining algorithms and treat the other topics in the second place. Table 1 summarizes the notations.

Based on these definitions, the major task of query-oriented summarization can be defined as follows: given a query and a document collection, the goal is to retrieve a document cluster related to the query and summarize the document cluster from different topical aspects of the query.

It is challenging to perform the task defined above. First, existing topic models only consider the general topic distribution of multiple documents, but cannot capture the query information. It is challenging on how to incorporate

Table 1: Notations.

SYMBOL	DESCRIPTION
T	number of topics
M	number of document in cluster C
V	number of unique words
N_d, N_q	number of word tokens in document d and query q
$\mathbf{w}_d$	vector form of word tokens in document d
$\mathbf{w}_q$	vector form of word tokens in query q
w_{di}, w_{qi}	the ith word token in document d or query q
z_{di}	the topic assigned to word token w_{di}
θ_d	multinomial distribution over topics specific to document d
θ_q	multinomial distribution over topics specific to query q
ϕ_z	multinomial distribution over words specific to topic z
α, α_q	Dirichlet priors of the topic mixtures for document and query
β	Dirichlet priors of the multinomial word distribution for each topic
x	The parameter indicating whether a word inherits the topic from the query ($x = 0$) or from the document ($x = 1$)
λ	Parameters for sampling the switch parameter x
γ, γ_q	Beta parameters for generating λ

the query information into the topic model and how to discriminate the different topics in the document cluster in a principled way. Second, it is unclear how to make use of the modeling results to calculate the score of each sentence and how to generate the final summarization result.

3 Overview of Proposed Approach

At a high level, our approach primarily consists of four steps:

1. We retrieve relevant documents to the query. Thus we obtain a query and its related document cluster.
2. We propose a unified probabilistic approach to uncover query-oriented topics for each summarization task (corresponding to a query and a document cluster). Specifically, we present two strategies for simultaneously modeling the query and the document cluster. This is the key for the following steps and also the focus of this paper.
3. We present four scoring methods to calculate the importance of each sentence in the document cluster based on the modeling results.
4. We generate the summary using sentences with the highest scores. We remove the redundant sentences using a clustering method.

The purpose of the first step is to retrieve documents only relevant to a query from the document collection. In general, we may use any retrieval method. In this paper, we used a standard language modeling approach [30]. To ensure coverage of documents, we perform query expansion, a commonly used method in information retrieval. For each word w_{qi} in the query q, we extract its frequently co-occurring words in the document cluster and add them into the query. We consider words appearing in a window-size of the word w_{qi} as its co-occurrence words, i.e. words before and after the word w_{qi}. We set the window size as 1.

In the second step, our main idea is to leverage a topic modeling method for query-oriented summarization and our main technical contributions also lie in this step where we propose a unified probabilistic approach to simultaneously model the document cluster and constrain the topic distribution so that it is close to the query. This could be done by separately modeling document contents and the query. However, such an approach cannot capture dependencies between document contents and the query, thus cannot obtain query-oriented topic distribution of the document cluster. We propose two strategies to incorporate the query information into the topic modeling process. The first strategy uses two generative processes to discover topical aspects of documents and the query, and then uses a Bernoulli distribution to associate the query-specific topic distribution and the document-specific topic distribution. The first strategy might suffer from the "too many parameters" problem. We therefore consider the second strategy, which directly adds a regularization term related to the query into the log likelihood objective function of the topic model so that the topic modeling process can be guided by the query.

In the third step, we present four scoring schemes to calculate the importance score of each sentence and in the last step, we select sentences with the highest scores as the summary. We also remove redundant sentences using a clustering method.

In the remainder of this section, we will first introduce two baseline methods based on existing general topic models: pLSI [11] and LDA [5]. We will then describe our proposed topic model in detail and explain how we make use of the learned models in the following steps.

4 Modeling of Query-oriented Topics

4.1 Baseline Models

4.1.1 Probabilistic Latent Semantic Indexing The probabilistic Latent Semantic Indexing (pLSI) model [11] has been proposed by Hofmann. pLSI assumes that there is a hidden topic layer $Z = \{z_1, z_2, \cdots, z_T\}$ between word tokens and documents. For each document, a topic distribution is learned and each word is generated from a chosen topic according to a document-specific topic distribution. In this way, the probability of generating a word w from a document d can be calculated by using the topic layer:

$$P(w|d) = \sum_{z=1}^{T} P(w|z)P(z|d) \tag{4.1}$$

For estimating the importance of a sentence, one can sum up the probability $P(w|d)$ of all words in the sentence (or use some other methods presented in Section 5).

4.1.2 Latent Dirichlet Allocation Latent Dirichlet Allocation (LDA) [5] is a three-level Bayesian network, which also models documents using a latent topic layer. In LDA, for each document d, a multinomial distribution θ_d over topics is first sampled from a Dirichlet distribution with parameter α. Second, for each word w_{di}, a topic z_{di} is chosen from this topic distribution and the word w_{di} is generated from a topic-specific multinomial $\phi_{z_{di}}$. Accordingly, the generating probability of word w from document d is:

$$P(w|d,\theta,\phi) = \sum_{z=1}^{T} P(w|z,\phi_z)P(z|d,\theta_d) \tag{4.2}$$

In a word, it uses the topic distribution θ_d of the document and the probability ϕ_w of the topic generating word to calculate the probability of the word. We can further use the method presented in Section 5 to calculate the score of each sentence.

Model Limitation. In the pLSI and the LDA model, topics are modeled using only document contents, but ignoring the query information. As a result, the 'irrelevant' topics to the query may hurt the summarization result.

4.2 Qurey-oriented Topic Modeling In this section, we present two strategies for incorporating the query information into the topic model. The first strategy is to integrate the query information into topic model by introducing a query-specific generative process. The second strategy directly regularizes the topic distribution related to the query, to avoid the possible "too many parameter tuning" problem in the first strategy.

4.2.1 Query Topic Model The proposed model is called query Latent Dirichlet Allocation (qLDA). The basic idea in this model is to use two correlated generative processes to model documents and the query together. The first process is to model the topic distribution of the query and the second process is to model the documents. It uses a Bernoulli distribution to determine whether to generate a word from a document-specific distribution or from a query-specific distribution. Figure 1 shows the graphical structure of the qLDA model.

Formally, the generative process can be described as follows: first, the user would decide what topics to query according to $p(\theta_q|\alpha_q)$, which is the topic distribution of the query; then for a word w_{di} in each document d of a cluster C, a coin is tossed according to $p(x=0|d) \sim beta(\gamma_q,\gamma)$, where $beta(.)$ is a Beta distribution; when $x=0$, the word w_{di} is sampled from a selected topic t_{qi} specific to the query

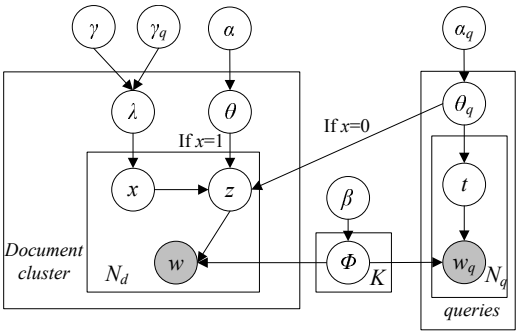

Figure 1: Graphical representation of the qLDA model.

q, according to distribution $P(w_{di}|\theta_{qt_{qi}})$; when $x=1$, the word w_{di} is sampled from a selected topic z_{di} specific to the document d, according to $P(w_{di}|\theta_{dz_{di}})$.

Intuitively, we use a Bernoulli distribution to determine whether a word in a document is related to ($x=0$) the query topic or not. In the qLDA model, the query information behaves like a kind of prior information to guide the topic distribution in the document, which in turn reflects the query information.

For inference, the task is to estimate the unknown parameters in the qLDA model. There are two sets of unknown parameters: (1) the distribution θ of M document-topics, the distribution θ_q of query-topics, the distribution λ of M document-Bernoulli, and the distribution ϕ of T topic-words; (2) the corresponding coin x_{di} and topic z_{di} for each word w_{di} in the document d, and the corresponding topic t_{qi} for each word w_{qi} in the query. It is usually intractable to do exact inference in such a probabilistic model. A variety of algorithms have been proposed to conduct approximate inference, for example variational EM methods [5] and Gibbs sampling [7]. We choose Gibbs sampling for its ease of implementation. Additionally, instead of estimating the model parameters directly, we evaluate (a) the posterior distribution on just t and then use the results to infer θ_q for the first generative process; (b) the posterior distribution on x and z, and then use the sampling results to infer θ, λ and ϕ for the second generative process. More specifically, we begin with the joint probability of the entire document cluster, and then using the chain rule, we obtain the posterior probability of sampling the topic for each word. For (a) we use the same sampling algorithm as that for the LDA model (cf. [7]), i.e. with the posterior probability:

$$P(t_{qi}|\mathbf{t}_{-qi},\mathbf{w}_q,\alpha_q) = \frac{n^{-qi}_{qt_{qi}}+\alpha_q}{\sum_t(n^{-qi}_{qt}+\alpha_q)} \frac{n^{-qi}_{t_{qi}w_{qi}}+\beta}{\sum_v(n^{-qi}_{t_{qi}v}+\beta)} \tag{4.3}$$

where n_{qt} is the number of times that topic t has been sampled from the multinomial distribution specific to query q; n_{tw} is the number of times that word w has been generated by topic t; the number n^{-qi} with the superscript $-qi$ denotes a quantity, excluding the current instance.

We can use an analogous method for parameter estimation in (b). We sample the coin x and the topic z together. Accordingly, the posterior probability of a word w inheriting topic z from the query-specific multinomial and the probability of the word w being generated from the document-specific multinomial are defined as:

$$P(z_{w_{di}}, x_{w_{di}} = 0 | \mathbf{w}_d, \mathbf{w}_q, \mathbf{z}_{-di}, \gamma, \gamma_q, \alpha_q, \beta) = \frac{n_{d0}^{-di} + \gamma_q}{n_{d0}^{-di} + n_{d1}^{-di} + \gamma_q + \gamma} \frac{n_{d0z_{w_{di}}}^{-di} + n_{qz_{w_{di}}} + \alpha_q}{\sum_z (n_{d0z}^{-di} + n_q + \alpha_q)} \frac{n_{z_{w_{di}} w_{di}}^{-di} + n_{z_{w_{di}} w_{di}}^{q} + \beta}{\sum_v (n_{z_{w_{di}} v}^{-di} + n_{z_{w_{di}} v}^{q} + \beta)} \tag{4.4}$$

$$P(z_{w_{di}}, x_{w_{di}} = 1 | \mathbf{w}_d, \mathbf{w}_q, \mathbf{z}_{-di}, \gamma, \gamma_q, \alpha, \beta) = \frac{n_{d1}^{-di} + \gamma}{n_{d0}^{-di} + n_{d1}^{-di} + \gamma_q + \gamma} \frac{n_{d1z_{w_{di}}}^{-di} + \alpha}{\sum_z (n_{d1z}^{-di} + \alpha)} \frac{n_{z_{w_{di}} w_{di}}^{-di} + n_{z_{w_{di}} w_{di}}^{q} + \beta}{\sum_v (n_{z_{w_{di}} v}^{-di} + n_{z_{w_{di}} v}^{q} + \beta)} \tag{4.5}$$

where n_{d0} is the number of times that topics of document d have been sampled from the query-specific topic distribution; n_{d1} is the number of times that topics of document d have been sampled from the document-specific topic distribution; a number n^q with the superscript q denotes that we count the numbers in all queries. For example, n_{zw}^q denotes the number of word w assigned to topic z in all queries.

As for the hyperparameters α, α_q, β, γ, and γ_q, one could estimate the optimal values by using a Gibbs EM algorithm [1] or a variational EM method [5]. For some applications, topic models are sensitive to the hyperparameters and it is necessary to get the right values for the hyperparameters. In the applications discussed in this work, we found that the estimated topic models are not very sensitive to the hyperparameters. Thus, for simplicity, we take fixed values (i.e., $\alpha = \alpha_q = 50/T$, $\beta = 0.01$, and $\gamma_q = 3.0$, $\gamma = 0.1$).

During parameter estimation, the algorithm keeps track of a $M \times T$ (document by topic) count matrix, a $T \times V$ (topic by word) count matrix, a $M \times 2$ (document by coin) count matrix, and a $|q| \times T$ (query by topic) count matrix. Given these matrices, we can easily estimate the probability θ_{dz} of a topic given a document, the probability θ_{qt} of a topic given a query, and the probability ϕ_{zv} of a word given a topic by:

$$\theta_{dz} = \frac{n_{dz} + \alpha}{\sum_{z'} (n_{dz'} + \alpha)} \tag{4.6}$$

$$\theta_{qt} = \frac{n_{qt} + \alpha_q}{\sum_{t'} (n_{qt'} + \alpha_q)} \tag{4.7}$$

$$\phi_{zv} = \frac{n_{zv}^d + n_{zv}^q + \beta}{\sum_{v'} (n_{zv'}^d + n_{zv'}^q + \beta)} \tag{4.8}$$

The qLDA model introduces a number of parameters, which are fixed for simplicity in our work. In a practical application, how to tune the parameters would be a challenging issue. We therefore consider the second strategy for incorporating the query information into the topic model using a regularization framework.

4.2.2 Topic Modeling with Regularization Our second strategy is to use the regularization framework to learn the query-oriented topic model. The basic idea is to use the query information to "guide" the topic model estimation by a discriminative approach. The method, called TMR, can be also considered with two sub-processes. The first process estimates the topic distribution of documents and the query. The second process adjusts the topic distribution so that topic distributions related to the query are strengthened. The two processes are trained simultaneously. Specifically, we aim at minimizing the regularized data likelihood:

$$O_\xi(C, q) = -\xi L(C) + (1 - \xi) R(C, q) \tag{4.9}$$

where $L(C)$ is the (log-)likelihood of the document cluster C by a topic model (e.g., LDA), $R(C, q)$ is a regularizer defined between the query q and the document cluster C, and ξ is a coefficient that controls the relative strength of the two terms. The regularizer $R(C, q)$ can be further defined as:

$$R(C, q) = \frac{1}{2} \sum_{d \in C} w(d, q) \sum_z (\theta_{qz} - \theta_{dz})^2 \tag{4.10}$$

where $w(d, q)$ denotes a similarity between document d and query q. We define the similarity as the cosine similarity based on word tokens. More accurately, we represent each document as a vector of words. The value of each element in the vector is calculated by $TF * IDF$. We also represent the query as a vector in the same way. We then calculate the cosine similarity of each document-vector and the query-vector.

By minimizing the objective function (Equation (4.9)), we attempt to not only find a probabilistic model that best fit the document cluster C, but also to guide the topic distribution toward the query-specific topic distribution.

Unfortunately, learning all the parameters in Equation (4.9) together is difficult, because we do not have a closed-form solution with the traditional EM-like or Gibbs sampling algorithm to re-estimate θ. We use instead an approach as Algorithm 1. The algorithm is suboptimal for optimizing the objective function. However, it provides an efficient way for parameter estimation and experiments also show its effectiveness. The regularization framework has been used previously for semi-supervised learning [31] and regularizing topic model with network information [20].

4.3 Computational Complexity We analyze the complexity of the proposed topic models. The qLDA model has a complexity of $O(L(\bar{N}_q + M\bar{N}_d)T)$, where L is the number of sampling iterations, $\bar{N}_q$ is the average number of word

Algorithm 1: Parameter estimation

1. initialize the parameters (θ, θ_q, ϕ) randomly;
2. run 200 burn-in sampling iterations using LDA [5];
3. run a two-stage training process iteratively:
   (a) train the model parameters (θ, θ_q, ϕ) using the objective function $O_1(C, q) = -L(C)$ with a standard Gibbs sampling procedure by setting the Dirichlet prior for each document as
   $$\alpha_{dz} = \alpha + \eta\theta_{qz}$$
   where θ_{qz} is the z-th dimension of the multinomial of query q with $1 \leq z \leq T$; α_{dz} is a parameter specific to the topic of document d; and η is a parameter.
   (b) fix ϕ, and re-estimate the multinomial θ to minimize O_ξ; we employ the algorithm proposed in [20] to optimize O_ξ, (i.e. run an iterative process to obtain the new θ for each document d by minimizing the objective function again.)
   $$\theta_{dz}^{(n+1)} = \mu\theta_{dz}^{(n)} + (1-\mu)w(d,q)\theta_{qz}$$
   where μ is a coefficient to smooth the topic distribution between documents and the query.

tokens in a (expanded) query, and $\bar{N}_d$ is the average number of word tokens in a document. It is difficult to accurately estimate the complexity of the TMR model, as it is hard to estimate the number of iterations k in Step 3(b) of Algorithm 1. Practically, the number of iterations k is between 7 and 15. Thus we have a complexity $O(L(M\bar{N}_d + k)T)$.

5 Generating Summary

5.1 Sentence Scoring Based on the learned topic models from the previous step, we can calculate the importance score of each word and then each sentence. We first compute the score of each word and use the average log of words' scores in a sentence as the sentence score. We test different methods to score the word. Specifically, we consider four word-scoring methods: *Max_Score*, *Sum_Score*, *Max_TF_Score*, and *Sum_TF_Score*.

Max_Score. A document cluster typically covers information of multiple topics. The query may ask for information about one specific topic, possibly the topic with the highest probability in the document cluster. Based on this idea, a word which is associated with this topic along with the highest probability should be assigned with a high score. The Max_Score is thus defined as

$$Max\_Score(w) = P(w|z = max_z(n_z^C)) \quad (5.11)$$

where n_z^C is the number of sampled topic z in the document cluster.

Sum_Score. In a topic model, with the number of topics increasing, intuitively the Max_Score would not be justified, the query may ask for information of several sub-topics. We thus consider the sum of the word's score on each topic: (We also tried a method by only considering the query-related topics. The result does not differ largely from that using the following form.):

$$Sum\_Score(w) = \sum_{z=1}^{T} P(w|z)P(z|d) \quad (5.12)$$

Max_TF_Score and Sum_TF_Score. The above two scoring methods only consider topic-based score for summarization. However, a topic based score may be too coarse and insufficient for obtaining high quality summaries. Nenkova et al. argued that the term frequency is also an important feature in determining the importance of a word/sentence [22]. We thus derive a combination form of the topic-based method and a term frequency-based method.

The first scoring method is called Max_TF_Score:

$$Max\_TF\_Score(w) = \delta\frac{n_w^C}{n^C} + (1-\delta)P(w|z_{max}) \quad (5.13)$$

The other method is called Sum_TF_Score is

$$Sum\_TF\_Score(w) = \delta\frac{n_w^C}{n^C} + (1-\delta)P(w|d) \quad (5.14)$$

where $P(w|z_{max})$ is the score by Max_Score; $P(w|d)$ is the score by Sum_Score; n_w^C is the number of w in the document cluster (like n_z^C); n^C is the number of all word tokens in C; and δ is a coefficient parameter.

5.2 Redundancy Reduction Finally, we use a clustering method to reduce redundancy in the summary. Specifically, we rank all sentences according to their scores and select the top ranked 150 sentences as candidate sentences. Then we represent each candidate sentence as a vector using words as features and their corresponding $TF * IDF$ scores as feature values. We cluster these sentences by CLUTO,[2] a single-link clustering method. After clustering, we obtain the center of each cluster by averaging all vectors in the cluster. Intuitively, a vector close to the center is more informative and can be selected to represent the cluster. We rank the vectors according to their distance to the cluster center. Each vector, i.e. a sentence, corresponds to a rank number. We assign the sentence a representative score as $(1 - r/n)$, where r is the rank number of the sentence and n is the number of sentences in that cluster. We combine this representative score with the score obtained from Section 5.1 to re-rank the candidate sentences.

[2] http://glaros.dtc.umn.edu/gkhome/views/cluto

Pros: Storage space, video quality, compatibility, sound quality, contact info, picture album
Cons: No integrated FM radio. can't place picture on background. the surface scratches easily
Full text: The new 80 GB iPod is wonderful! I just got it and uploaded my entire music library on to it and my friends as well and it handles all that with no problem...

Figure 2: A product review from www.epinions.com.

6 Experimental Results

In this section, we present the evaluation of the proposed approach.

6.1 Experimental Setup

6.1.1 Data Sets We evaluated the proposed approach on two different types of data sets: DUC data and Epinions data.

Document Understanding Conference (DUC), now moved to Text Analysis Conference (TAC), provides a series of benchmarks for evaluating approaches for document summarization. We conducted experiments on DUC2005, which contains 50 query-oriented summarization tasks. Documents are from Financial Times of London and Los Angeles Times. For each query, a relevant document cluster is assumed to be "retrieved", which contains 25-50 documents. Thus, the task is to generate a summary from the document cluster for answering the query. (In DUC, the query is also called "narrative" or "topic").

The Epinions data was crawled from `www.epinions.com`, a web site for posting product reviews. We collected reviews of 44 different "iPod" products. The reviews were written by Epinions users. Each review, as shown in Figure 2, may consist of "pros", "cons", and a full text review. "pros" denotes positive aspects provided by the user; "cons" denotes negative aspects; while the full text provides the detailed review. Each of these 44 product names was used as the query to retrieve relevant reviews from the website. For each query, we retrieved the top ranked reviews (≤ 50). In total, we gathered 1,277 reviews. We created a document cluster for each "iPod" product by using only full text reviews. Thus the task can be viewed as to find the best descriptive text to support users' "pros" or "cons" opinion. The Epinioins data does not have a ground truth summary. For evaluation purpose, we asked two students who are fans of "iPod" products to provide answers for each summarization task.

For both data sets, we preprocessed each document by (a) removing stopwords and numbers; (b) removing words that appear less than three times in the corpus; and (c) downcasing the obtained words.

6.1.2 Evaluation Measures We conducted evaluations in terms of ROUGE [15]. The measure evaluates the quality of the summarization by counting the number of overlapping units, such as n-grams, between the generated summary by a method and a set of reference summaries. Basically, ROUGE-N is an n-gram recall measure. It is defined as follows:

$$C_n = \frac{\sum_{C \in \{ModelUnits\}} \sum_{n-gram \in C} Count_{match}(n-gram)}{\sum_{C \in \{ModelUnits\}} \sum_{n-gram \in C} Count(n-gram)} \quad (6.15)$$

where $Count_{match}(n-gram)$ is the maximum number of $n-grams$ co-occurring in a generated summary and the reference summaries and $Count(n-gram)$ is the number of $n-grams$ in the reference summaries. We calculated the macro-average score of ROUGE-N on all document clusters. We utilized the tool ROUGE 1.5.5,[3], with the parameter setting "-n 2 -l 250 -w 1.2 -m -2 4 -u -f A -p 0.5 -t 0" for the DUC data, where "-l 250" indicate that only the first 250 words were considered in evaluation. For the Epinions data, we generate 100 words summary. Thus the length parameter "-l" was set to 100.

6.1.3 Summarization Methods We define the following baseline methods for query-oriented summarization:

TF: it uses only term frequency for scoring words and sentences. The method is similar to that proposed in [22], except that [22] also adjusts the scoring measure to further reduce redundancy.

pLSI: it uses a general topic model to learn the topic distribution from the data set. As for the topic model, we employed pLSI [11]. After learned the topic model, we used the same scoring methods as that in our approach, i.e. Max_Score and Sum_Score.

PLSI + TF: it combines the pLSI score and the TF scores. The Max_TF_Score and Sum_TF_Score are used.

LDA: it uses another general topic model, LDA [5], to learn the topic model. After learned the topic model, we used the same scoring methods as that in our approach, i.e. Max_Score and Sum_Score.

LDA + TF: it combines the LDA score and the TF score. The Max_TF_Score and Sum_TF_Score are used.

qLDA: it uses the proposed qLDA model to train the topic model and further uses the proposed scoring methods (cf. Section 5.1) to rank sentences.

qLDA + TF: it combines the qLDA score and the TF score. The Max_TF_Score and Sum_TF_Score are used.

TMR: it uses the proposed TMR model to train the topic model and further uses the proposed scoring methods (cf. Section 5.1) for ranking sentences.

TMR + TF: it combines the TMR score and the TF score. The Max_TF_Score and Sum_TF_Score are used.

[3] http://berouge.com/default.aspx

Table 2: Results of the nine methods on DUC ($T = 60$, Max_Score for ranking sentences). "w/o RR" denotes a result without redundancy reduction; "with RR" denotes a result with redundancy reduction.

	Method	Rouge1	Rouge2	RougeSU4
TF	w/o RR	0.36956	0.07017	0.12721
	with RR	0.36919	0.06883	0.12647
pLSI	w/o RR	0.34298	0.06217	0.11076
	with RR	0.34737	0.06386	0.11625
pLSI+TF	w/o RR	0.36215	0.06570	0.11834
	with RR	0.36348	0.06713	0.12091
LDA	w/o RR	0.36379	0.06421	0.12144
	with RR	0.36719	0.06757	0.12445
LDA+TF	w/o RR	0.37043	0.06862	0.12710
	with RR	0.37154	0.06845	0.12710
qLDA	w/o RR	0.37530	0.06959	0.12841
	with RR	0.37918	0.07003	0.13060
qLDA+TF	w/o RR	0.37571	0.07144	0.13015
	with RR	0.37479	0.07096	0.12991
TMR	w/o RR	0.37842	0.07196	**0.13217**
	with RR	**0.38093**	0.07095	0.13047
TMR+TF	w/o RR	0.38020	**0.07203**	0.13126
	with RR	0.37752	0.07147	0.13038

6.1.4 Topic Model Estimation For both LDA and qLDA, we performed model estimation with the same setting. As for the topic number, we set $T = 60$ for the DUC data set, and $T = 30$ for the Epinions data set. The topic number is determined by empirical experiments (more accurately, by minimizing the perplexity [2], a standard measure for estimating the performance of a probabilistic model, the lower the better). One can also use some solutions like [27] to automatically estimate the number of topics.

All experiments were carried out on a Server running Windows 2003 with two Dual-Core Intel Xeon processors (3.0 GHz) and 4GB memory. For the DUC data set, it took about 20 minutes for estimating the LDA model, 55 minutes for estimating the qLDA model, and 80 minutes for the TMR model (for 2000 sampling iterations when the topic number is 60). Epinions data took a shorter time, i.e., LDA in a few minutes, qLDA in around 15 minutes, and TMR in about 20 minutes.

6.2 Results on DUC Table 2 shows the performance of summarization using our proposed methods (with $T = 60$ and Max_Score) and the baseline methods on the DUC data set. We employed the same redundancy reduction process (cf. Section 5.2) for all methods. We see that both before and after redundancy reduction, our proposed approaches (qLDA and TMR) outperform the baseline methods in terms of all measures.

We can also see that the combination of the topic model with the term frequency can improve the summarization performance. The best results were obtained when the parameter δ in Equation (5.13) is between 0.4 and 0.6. (The value is found by empirical experiments.)

We give an analysis on the summarization task D307, which talks about "new hydroelectric projects". The query is "what hydroelectric projects are planned or in progress and what problems are associated with them?" Four human summaries (A, B, C, and D) are provided for this task. Specifically, summaries B, C, and D mainly focus on problems of the projects, like environmental problems, social problems, financial problems, and so on. Summary A lists all projects and abstracts of the problem. The summary obtained by TF and LDA contains sentences about the financial problem and the power of the projects, but does not cover all the topics. qLDA based summary covers more topics such as, environmental problem, social problem of displacing people. TMR based summary covers more project information than the other methods. Table 3 shows sample sentences from the summaries obtained by different approaches.

To reveal the reason why the topic-based approaches (e.g., LDA, qLDA, and TMR) can outperform the word-based approach (e.g., TF). We conducted an topical analysis for each document cluster. Figure 3 shows the topic distribution on the document cluster D357. We see that in D357, when set the topic number as 60, there is a major topic (Topic 6, talking about "boundary disputes involving oil"). But still about 60% of the information is captured by the other topics. In a word-based approach, the extracted summary would be dominated by frequent words (words from the major topic). When set with a large topic number ($T = 250$), there would be multiple dominated topics (e.g., #77, #117, #205, and #217). Our approaches can discriminate words from different topics, and thus obtain a better performance.

We further compared our results with the three best participant systems [12] [23] [28] in DUC2005. Table 4 shows the comparison results. We see that our proposed approaches outperform the three systems. In terms of Rouge1, TMR achieves the best result (0.38093) among all methods. In terms of Rouge2, the best performance (0.07147) is achieved by TMR+TF. We need note that our approaches do not make use of any external information; while the systems usually (heavily) depend on some external knowledge, for example System 15 employs WordNet for discovering semantic similarity between words [28]; and System 17 employs linguistic features such as named entity, linguistic patterns, and semantic similarity between words [12].

We also conducted experiments on the DUC2006 data. The task is similar as that for DUC2005. Table 5 shows the results of our approaches and the best three participant systems on DUC2006. We see that the best performance was achieved by the TMR method in terms of Rouge1.

Table 3: Sample sentences obtained by different methods for the summarization task of D307 in DUC2005.

Summary obtained by TF	The Pangue dam, already under construction, is likely to cost Dollars 450m. However, it needs foreign capital to complete even the first dam. If built, it would be by far the largest electrical project in the world financed by the private sector. The first dam alone will produce almost as much power as Nepal's existing generating stations which have a capacity of just 230MW. The dam will cost 1.13m people their homes by the time the project nears completion in the year 2008.
Summary obtained by LDA	The Pangue dam, already under construction, is likely to cost Dollars 450m. These include village-level micro-dams, medium-sized dams for towns and at least one large project - a Dollars 300m, 140MW project on the river Kaligandaki in central Nepal. The flood-control and power-generating benefits go primarily to the citizens of downstream Hubei province. So the scheme went ahead with stringent conditions for resettling and paying compensation to displaced villagers and for an environmental master plan. However, many tribespeople say they have not been consulted about Bakun or given any project details.
Summary obtained by qLDA	Yacyreta, being built by Impregilio of Italy and Dumez of France, is the second big hydro-electric dam to be built on the Parana, and two more are planned. It is being built at a site more than 500km upriver from Sarawak's coast. Upriver of the dam, Sichuan province bears the brunt of the project's environmental damage and has 85 per cent of the people to be moved. The south-east Anatolian project or Gap, as it sometimes better known, is currently the largest development project in the Mediterranean. Opponents of the project say the dam will be 'an ecological time bomb'. The dam will cost 1.13m people their homes by the time the project nears completion in the year 2008. Bakun is being built on the upper reaches of the Rejang, Malaysia's longest river. At the time opposition politicians and environmentalists questioned both the size of the project and its environmental impact.
Summary obtained by TMR	China's Three Gorges project on the Yangtse would: displace 1.13 million people; do major social and environmental damage to Sichuan province while providing the most benefits to Hubei province; and be of enormous expense. President Carlos Menem will today inaugurate what he once called Argentina's 'monument to corruption', the 3,200 MW Yacyreta hydroelectric dam on the Parana River. The funding for a controversial hydroelectric dam to be built in the heart of Malaysia's tropical rainforest will be generated from domestic sources, according to the company in charge of the project. New hydroelectric projects pose social, political, economic and environmental problems. The Yacyreta dam was delayed by Argentina's numerous political and economic crises.

Table 4: Comparison with the three best systems on DUC2005.

Method	Rouge1	Rouge2	RougeSU4
System15	0.37469	0.07020	0.13133
System4	0.37436	0.06831	0.12746
System17	0.36900	0.07132	0.12933
qLDA	0.37918	0.07003	**0.13060**
qLDA +TF	0.37571	0.07144	0.13015
TMR	**0.38093**	0.07095	0.13047
TMR +TF	0.37752	**0.07147**	0.13038

Table 5: Comparison with the three best systems on DUC2006.

Method	Rouge1	Rouge2	RougeSU4
System24	0.41062	0.09514	0.15489
System15	0.40197	0.09030	0.14677
System12	0.40398	0.08904	0.14686
qLDA	0.40211	0.08687	0.14419
qLDA+TF	0.40410	0.08967	0.14800
TMR	**0.41176**	0.08759	0.14213
TMR+TF	0.40626	**0.09132**	**0.15037**

6.3 Results on Epinions Table 6 presents the results of different approaches on the Epinions data set. Generally, we see that topic-based methods can obtain better performance than the word-frequency based methods (e.g., TF). The reason may be that one review always covers several features of a product. In a topic model, these features are modeled by different topics. A topic-level summary can thus include the multiple features. The word-based method may bias to a few features which are frequently talked. We have also found that the result of LDA + TF does not differ significantly from that of LDA, for both scoring schemes Max_TF_Score and Sum_TF_Score. This is different from what we discovered on the DUC data set. For qLDA and TMR, combining with TF improves the performance. TMR + TF (Sum_Score) obtains the best results among all the methods.

We show an example summary for a product "iPod Touch". Table 7 shows the summaries obtained by TF, LDA, qLDA, and TMR. For LDA, we only used the full review

Table 7: Sample sentences obtained by different methods for the product "iPod Touch".

Summary obtained by TF	The Touch is an iPod that also plays videos so their functions should be different. However, there's something about the iPods that get me every time. This is a con for all iPods, not just the new Touch. The multi-touch screen is very useful when browsing through the music on my ipod. The iPod Touch is an amazing device for not only music, but as a full on PDA. This is what the Touch was designed to do, and it does it well! The iPod Touch also needs a good chat application
Summary obtained by LDA	This is what the Touch was designed to do, and it does it well! As of now it is neither here nor there device. This is a con for all iPods, not just the new Touch. The iPod Touch is an amazing device for not only music, but as a full on PDA. The multi-touch screen is very useful when browsing through the music on my ipod. The Touch has WiFi, so the internet is available at any open access points. The overall design is functional, but elegant and impressive. The multi-touch interface adds many capabilities that other web-capable PDA's lack.
Summary obtained by qLDA	However, there's something about the iPods that get me every time. A new feature Apple includes on the iPod is the iTunes Store. The Touch is an iPod that also plays videos so their functions should be different. It was really noticeable when placed next to a video iPod running the same video. The 32GB model at 7,000 songs should be more than enough for most music listeners. It'll play music for 15 hours - or videos for 5 or 6 hours on a charge. Unless, of course, you want to download anything except music from the iTunes store.
Summary obtained by TMR	Great wireless applications, easy to use, video screen is great. Slim, lightweight, stunning, Wi-fi, Safari, beautiful graphics. Has internet access so you can still check e-mails online and browse the web pages. Love the wi-fi and all that it brings to the table. I'm extremely pleased with my iPod touch. The 16GB is perfect for several movies and TV shows, and music. Pricey, missing features to make it an all-around use device. Back case scratches easily, controls difficult to work "blind", no hard drive capability.

text to train the parameters. For qLDA and TMR, we expand the query by words in "pros" and "cons" of the review. We also expanded the query by extracting its frequently co-occurring words from the full text review. From the table we can see that summaries obtained by qLDA and TMR are more informative than the word-based method (TF). LDA also covers "iPod Touch" features like "wiFi" and "overall design" that cannot be extracted by the TF method. qLDA and TMR further cover the product features like "iTunes store", "storage capacity" and "battery life" that cannot be discovered by the TF method.

7 Related Work

Document(s) summarization has long been viewed as a challenging issue for text mining.

The extraction-based document summarization method ranks sentences by their scores and selects ones with the highest scores as summaries. Different approaches employ different methods for estimating the importance of sentences. Features such as term frequency [22] [29], cue words, stigma words, topic signature [13] and topic theme [10] are used to measure the importance of words. A composition function is utilized to score the sentences.

Nenkova et al. [22] argue that the term frequency is an important factor for extracting summaries. They treat the ratio of a word occurring number in a document cluster as the score of a word. They utilize an average sum of the word score to compute the score of a sentence.

Conroy et al. [6] propose an oracle score based on the probabilistic distribution of unigrams of human summaries. They utilize the query words and the topic signature terms to approximate the probabilistic distribution, corresponding to word score in our approach. Their query words and signature terms correspond to our expanded query. Our proposed model is relevant to this model in the aspect that both utilize the query and some document words to obtain the probabilistic distribution of words. The difference is that their model does not consider the topical aspects in the document cluster and it uses the signature terms for query expansion. More specifically, if we set the number of topics in qLDA to 1 and use the same signature terms as they use, our qLDA will degeneralize to their model.

Over the last several years, several methods have been proposed for documents summarization by making use of latent topics in the document cluster. For example, Barzilay and Lee [4] use the Hidden Markov Model to learn a latent topic for each sentence, called as a V-topic. They choose the "important" topics that have the high probability of generating summary sentences. The sentences generated by these topics are included in the summary. Their method

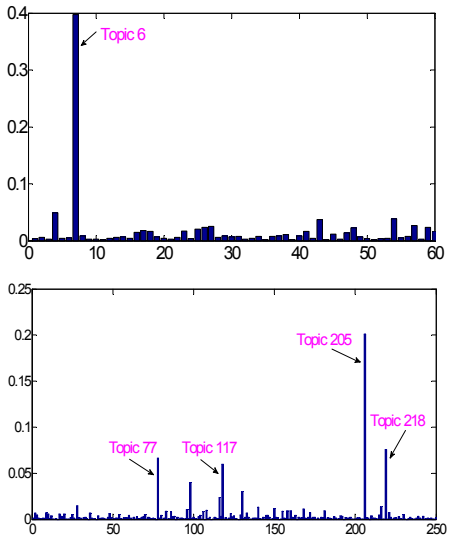

Figure 3: Topic distribution for in D357 (T=60 and T=250). The x axis denotes topics and the y axis denotes the occurrence probability of each topic in D357.

Table 6: Results of different methods on the Epinions data.

Method	Rouge1	Rouge2	RougeSU4
TF	0.28486	0.07579	0.10588
pLSI (Max Score)	0.28763	0.08031	0.10215
pLSI (Sum Score)	0.28291	0.07628	0.10072
pLSI+TF (Max Score)	0.28653	0.07956	0.10572
pLSI+TF (Sum Score)	0.29104	0.07832	0.10721
LDA (Max Score)	0.29155	0.08218	0.11245
LDA (Sum Score)	0.28761	0.07827	0.10671
LDA+TF (Max Score)	0.28483	0.07228	0.10403
LDA+TF (Sum Score)	0.29305	0.07802	0.10796
qLDA (Max Score)	0.29113	0.07765	0.10662
qLDA (Sum Score)	0.28815	0.08189	0.10803
qLDA+TF (Max Score)	0.29698	0.07430	0.10741
qLDA +TF (Sum Score)	0.30326	0.08774	0.11611
TMR (Max Score)	0.30453	0.08349	0.11590
TMR (Sum Score)	0.31271	0.08960	0.11931
TMR+TF (Max Score)	0.31647	0.08815	0.11758
TMR +TF (Sum Score)	**0.32375**	**0.09081**	**0.12894**

requires a labeled training data to learn the probability. Topic models like pLSI has been applied to text document summarization [4]. This model is limited in its scalability to large data collections. More importantly, all of the aforementioned topic-based methods do not consider the query information or integrate the query information in a heuristic way.

Another related problem is opinion summarization in Weblogs [8] [16], where the goal is mainly to mine user opinions by identifying and extracting positive and negative opinions or analyzing and extracting topical contents of blog articles. None of this body of work consider the query information.

More recently, some research effort has been made to incorporate the query information into the topic model. For example, Daumé and Marcu [9] propose a hierarchical Bayesian model to compute the relevance of a sentence to a query. The model uses a multinomial distribution to select whether to sample a word from a document-specific, a query-specific or a document cluster-specific distribution. However, the query is usually very short and a method that directly model the query generation may be in sufficient. In comparison, we use two strategies to associate the document content with the query. The idea of regularizing topic model is inspired by [31] and [20]. The difference is that [31] uses a regularization term to learn a semi-supervised model on a graph and [20] uses a regularization term to learn pLSI models with network information; while we use a regularization term to guide LDA for modeling query and documents together. We propose an approximate algorithm for parameter estimation in the model.

8 Conclusion

In this paper we investigate the problem of multi-topic based query-oriented summarization. We formalize the major tasks and propose a probabilistic approach to solve the tasks. We study two strategies for simultaneously modeling document contents and the query information. We present four methods to score sentences in the documents based on the learned topic models. Experimental results on the DUC data and the product opinion data show that our proposed approach outperforms the baseline method using word frequency (e.g., TF) and that of using general topic-based methods (e.g., pLSI and LDA). The proposed approach is quite general and flexible. The method can be also used for single document summarization and multi-document summarization. It can be applied to many mining tasks, such as product opinion analysis and question answering.

There are many potential future directions of this work. One potential issue is how to obtain more accurate prior information, for instance query expansion when the query is short. In addition, the approach can be extended to perform topic model in a (semi-)supervised manner. For example, in a summarization task, a user points out that some sentences should be included in the summary, we can incorporate these sentences into the approach as the supervised information. Additionally, we are going to integrate the topic-based documents summarization as a new feature into our academic search system ArnetMiner [26] (`http://arnetminer.org`). When the user inputs a query, the system will identify the major topics related to the query and then generate a summary by considering the different topics.

References

[1] C. Andrieu, N. de Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50:5–43, 2003.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, 1999.

[3] R. Barzilay, N. Elhadad, and K. R. Mckeown. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55, 2002.

[4] R. Barzilay and L. Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *Proceedings of HLT-NAACL'04*, pages 113–120, 2004.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[6] J. M. Conroy, J. D. Schlesinger, and D. P. O'Leary. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of ACL'06*, pages 152–159, 2006.

[7] T. L. Griffiths and M. Steyvers. Finding scientific topics. In *Proceedings of the National Academy of Sciences*, pages 5228–5235, 2004.

[8] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins. The predictive power of online chatter. In *Proceedings of KDD'05*, pages 78–87, 2005.

[9] I. Hal Daumé and D. Marcu. Bayesian query-focused summarization. In *Proceedings of ACL'06*, pages 305–312, 2006.

[10] S. Harabagiu and F. Lacatusu. Topic themes for multi-document summarization. In *Proceedings of SIGIR'05*, pages 202–209, 2005.

[11] T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99)*, pages 50–57, 1999.

[12] W. Li, W. Li, B. Li, Q. Chen, and M. Wu. The hong kong polytechnic university at duc2005. In *Proceedings of DUC2005*, 2005.

[13] C.-Y. Lin and E. Hovy. The automated acquisition of topic signatures for text summarization. In *Proceedings of COLING'00*, pages 495–501, 2000.

[14] C.-Y. Lin and E. Hovy. From single to multi-document summarization: a prototype system and its evaluation. In *Proceedings of ACL'01*, pages 457–464, 2001.

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

[16] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of WWW'05*, pages 342–351, 2005.

[17] I. Mani and E. Bloedorn. Machine learning of generic and user-focused summarization. In *Proceedings of AAAI'98/IAAI'98*, pages 820–826, 1998.

[18] K. McKeown, R. J. Passonneau, D. K. Elson, A. Nenkova, and J. Hirschberg. Do summaries help? a task-based evaluation of multi-document summarization. In *Proceedings of SIGIR'05*, pages 210–217, 2005.

[19] K. R. Mckeown, V. Hatzivassiloglou, R. Barzilay, B. Schiffman, D. Evans, and S. Teufel. Columbia multi-document summarization: Approach and evaluation. In *Proceedings of DUC'01*, 2001.

[20] Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceedings of the 17th International World Wide Web Conference (WWW'08)*, pages 101–110, 2008.

[21] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of WWW'07*, pages 171–180, 2007.

[22] A. Nenkova, L. Vanderwende, and K. McKeown. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of SIGIR'06*, pages 573–580, 2006.

[23] R. J. Passonneau, A. Nenkova, K. McKeown, and S. Sigelman. Applying the pyramid method in duc 2005. In *Proceedings of DUC'05*, 2005.

[24] S. Sekine and C. Nobata. A survey for multi-document summarization. In *Proceedings of HLT-NAACL 2003 Workshop: Text Summarization (DUC'03)*, pages 65–72, 2003.

[25] D. Shen, J. tao Sun, H. Li, Q. Yang, and Z. Chen. Document summarization using conditional random fields. In *Proceedings of IJCAI'07*, pages 2862–2867, 2007.

[26] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[27] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarhical dirichlet processes. In *Technical Report 653, Department of Statistics, UC Berkeley*, 2004.

[28] S. Ye, L. Qiu, T.-S. Chua, and M.-Y. Kan. Nus at duc 2005: Understanding documents via concept links. In *Proceedings of DUC'05*, 2005.

[29] W.-T. Yih, J. Goodman, L. Vanderwende, and H. Suzuki. Multi-document summarization by maximizing informative content-words. In *Proceedings of IJCAI'07*, 2007.

[30] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR'01*, pages 334–342, 2001.

[31] X. Zhu and J. Lafferty. Harmonic mixtures: Combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Proceedings of ICML'05*, pages 1052–1059, 2005.