Summarizing Web Forum Threads based on a Latent Topic Propagation Process

Zhaochun Ren[1], Jun Ma[1], Shuaiqiang Wang[2] and Yang Liu[1]
School of Computer Science and Technology
[1]Shandong University, [2]Shandong University of Finance
Jinan, China
{zhc.ren, shqiang.wang}@gmail.com, {majun, yangliu}@sdu.edu.cn

ABSTRACT

With an increasingly amount of information in web forums, quick comprehension of threads in web forums has become a challenging research problem. To handle this issue, this paper investigates the task of Web Forum Thread Summarization (WFTS), aiming to give a brief statement of each thread that involving multiple dynamic topics. When applied to the task of WFTS, traditional summarization methods are cramped by topic dependencies, topic drifting and text sparseness. Consequently, we explore an unsupervised topic propagation model in this paper, the Post Propagation Model (PPM), to burst through these problems by simultaneously modeling the semantics and the reply relationship existing in each thread. Each post in PPM is considered as a mixture of topics, and a product of Dirichlet distributions in previous posts is employed to model each topic dependencies during the asynchronous discussion. Based on this model, the task of WFTS is accomplished by extracting most significant sentences in a thread. The experimental results on two different forum data sets show that WFTS based on the PPM outperforms several state-of-the-art summarization methods in terms of ROUGE metrics.

Categories and Subject Descriptors

H.3.1 [**Content Analysis and Indexing**]: abstracting methods; H.3.3 [**Information Search and Retrieval**]: Information filtering; I.2.6 [**Artificial Intelligence**]: Learning

General Terms

Algorithms, Experimentation

Keywords

Web Forums, Thread Comprehension, Summarization, Topic Modeling

1. INTRODUCTION

Web forum is an online portal for open threads on specific issues. For each thread, a user makes an initial post and others express their opinions by replying to some previous ones. In web forums, people participate in threaded discussions to deliver knowledge and ask questions. Unfortunately, they have to face ascending amount of redundant thread information when browsing web forums. Generally, people often read only few posts ranked ahead in a thread with large posts quantities, whereas this scanning style only reflects incomplete views of the conversation. Hence with an increase of the thread volume, the requirement for summarizing each thread to help human comprehension is becoming more and more urgent. Intuitively, a smooth transition from multi-document summarization to web forums seems to be reasonable. However, as complicated and dynamic topics exist in each ongoing thread, traditional multi-document summarization methods fail to capture the following three characteristics that are crucial to summarize the content of one thread, especially in those ones with large quantities of posts:

Topic Dependencies. Thread is a kind of asynchronous conversation based on temporal topic dependencies among posts. When one thread participant A replies to another post's author B, we consider a topic dependency has been built from user B to user A. Yet there are other factors in reality, it is believed that the reply relations among posts dominant the topic dependencies [8].

Topic Drifting. A conversation always results in various subtopics. As the post conversation progresses, the semantic divergence among these subtopics will be widened.

Text Sparseness. Most posts are composed of short and elliptical messages. As short texts do not provide sufficient term co-occurrence information, traditional text representation methods, such as "tf-idf", have several limitations when directly applied to the mining tasks [5].

In this paper, we investigate the task of Web Forum Thread Summarization (WFTS), aiming to generate a compressed version of a given thread delivering the majority of topics adequately. To handle the three challenges, tracking those dynamic topics existing in each thread has become a significant work to generate the summarization. In recent years, topic models has attracted much attention to the topic discovery and tracking in complicated structural data [6, 9, 13, 1, 14]. This paper explores a novel topic model, called Posts Propagation Model (PPM) [11], that takes account of *topic dependencies*, *topic drifting* and *text sparseness* during the thread modeling process.

Illuminated by the Dirichlet-tree distribution [2], PPM connects topic dependencies based on reply-relations so that topic distributions depend on the product of parameters of its modified structural ancestors. Based on probabilistic distributions derived from the PPM, we accomplish WFTS task by extracting significant sentences to generate the summarization. Topic-sensitive markov random walks(MRW) and mutual reinforcement between posts and sentences is employed for sentence scoring process.

CIKM'11, October 24–28, 2011, Glasgow, Scotland, UK.

The rest of this paper is organized as follows: Our problem formulation is demonstrated in section 2. The Model (PPM) is demonstrated in section 3. In section 4 PPM-based WFTS is detailed. Section 5 presents and discusses the experiments and results, and section 6 concludes the paper.

2. PROBLEM FORMULATION

In this section, we discuss the summarization problem in detail and then present the definition of the WFTS task. First, we introduce some basic concepts in web forums:

DEFINITION 1 (POST). *A post is a user-submitted message containing the content and the time it was submitted. A group of related posts constitute into one thread, where all posts appear as boxes one after another. Except for the root post in one thread, each post is submitted to "reply" one of its previous post in the thread.*

DEFINITION 2 (THREAD). *A thread is a collection of posts, usually displayed from oldest to latest. One thread begins with a root post that may contain questions or news, and is followed by a series of non-root posts, each of which replies to one of its previous posts in the same thread.*

As mentioned in previous section, ascending amount of redundant thread in web forums make the quick comprehension become more and more difficult for participants. Based on this structure in web forums, we investigate the problem of quick comprehension of one forum's thread with multiple posts. An intuitive way to help people understand the thread rapidly is to generate a brief and understandable summary for all posts in the ongoing thread. Therefore, this paper handles the Web Forum Thread Summarization (WFTS) task that extracts sentences from the thread to generate a summary. To formally define our problem, we give the definition of the task of Web Forum Thread Summarization(WFTS):

DEFINITION 3 (WEB FORUM THREAD SUMMARIZATION). *For one thread with $|S|$ sentences $\{s_1, s_2, ..., s_{|S|}\}$, the task of Web Forum Thread Summarization(WFTS) aims to generate a brief statement of the whole thread by extracting Lim significant sentences from all posts, where Lim is a threshold of the summary size.*

An intuitive way to solve WFTS task is just transfer traditional document-summarization methods to the WFTS task. However, existence of multiple dynamic topics and the short-text nature in each post obstruct this transformation process. As we have mentioned previously, *topic dependencies*, *topic drifting* and *text sparseness* are three main challenges in WFTS process. To handle the three challenges, using a dynamic way to track all topics seems to be a reasonable solution. Thus, in this paper we employ a dynamic topic model, efficient in track topics among complicated documents, to model each thread in web forums.

3. THREAD TOPIC MODELING

Based on the reply relations among posts, we extract the reply-relation tree (RRT) structure to represent each thread, intuitively.

DEFINITION 4 (REPLY-RELATION TREE). *The built-in structure of a thread can be represented by a tree (r, V, E), where r is the root post and V refers to the set of posts in the thread. $\forall u, v \in V$, $\langle u, v\rangle \in E$ iff post v replies post u.*

For our modeling task, thread in web forums is a kind of ongoing document. If we handle all posts as a whole collection, every time to generate the summary would have to re-train the whole model. Thus a trade-off agglomeration process [7] is employed here to merge those short posts submitted close in time. At last, RRT$= (r, V, E)$ is transformed into a graph structure as RRG(reply-relation graph)$= (s, \widehat{V}, \widehat{E})$, which is defined below.

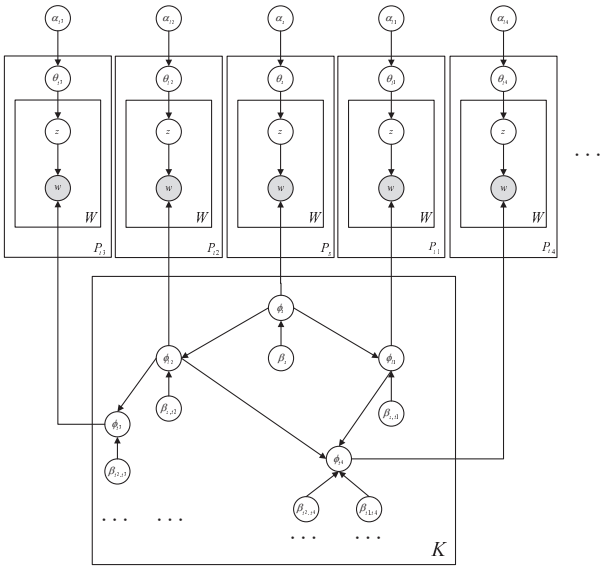


Figure 1: Graph model representation for Posts Propagation

DEFINITION 5 (REPLY-RELATION GRAPH). *A RRG$= (s, \widehat{V}, \widehat{E})$ is a directed network derived from a RRT$= (r, V, E)$, where s is a source node with in-degree is 0 and $r \in s$. $\widehat{V}$ and $\widehat{E}$ are the node set and edge set of RRG respectively. $\forall t \in \widehat{V}$, $note(t)$ denotes a nodes subset of V, such that, all nodes of $note(t)$ were reduced into the node in $\widehat{V}$ and labeled by t. Edge $\langle t_i, t_j\rangle \in \widehat{E}$ iff $\exists u \in t_i, \exists v \in t_j, \langle u, v\rangle \in E$ in RRT$= (r, V, E)$.*

Given the RRG$= (s, \widehat{V}, \widehat{E})$ after the agglomeration, we primarily define the following notations for each thread.

- One thread has been separated into $\left|\widehat{V}\right|$ discrete post collections
- All post collections cover the same vocabulary **W**
- All post collections share the same topic number K
- Each post collection $note(t)$ contains $|P_t|$ posts.

As we all know, the *topic dependencies* and the *topic drifting* both come with reply-relations among nodes in RRG. Thus for node t (not the source node) in RRG$= (s, \widehat{V}, \widehat{E})$, a reasonable assumption is derived that there are some semantic dependencies between t and all its predecessors in paths from t to s. Illuminated by the Dirichlet-tree distribution [2], we calculate the *topic dependencies* in node t using a product of Dirichlet distributions placed over the probabilities of each internal node on each path from source node to t. We denote this calculation as PoD process.

Given RRG$= (s, \widehat{V}, \widehat{E})$, for $note(t), t \in \widehat{V}$, dependencies ϕ_t is generated from a product of Dirichlet distribution over $t's$ predecessors. Let $\beta_{t,t'}$ be the edge weight from node t to one of its successors t', which characterizes the contact between t and t'. E_t is the set of edges included in any path from the source node s to

t. To generate $\phi_t \sim PoD(\{\beta_{c,d}, \phi_c, \phi_d\}_{(c,d)\in E_t})$, we first draws the posterior PoD values for t's predecessors; then ϕ_t is derived from the product of posterior parameters of each edge.

To reduce another influence of the *text sparseness*, background semantic similarities is added to the PPM modeling process. As mentioned earlier, participants usually use different terms to describe the same topic. Intuitively, it will be helpful to integrate several semantic similarities into the calculation of topic dependencies. For the vocabulary in a thread, we establish a $W \times \kappa$ similarity matrix $\mathfrak{W}_{sim}$ where each row corresponds to a similarity vector including semantic similarities from κ most similar words. We obtain the semantic similarity $sim(w, w')$ between word w and w', from the WordNet:Similarity tool[1]. Thus for post collection t we rewrite the parameter $\phi_{z,t}$ over topic z:

$$\prod_{(c,d)\in E_t} \frac{\Gamma(\sum_w \Delta_{z,c,d,w})}{\prod_w \Gamma(\Delta_{z,c,d,w})} \prod_w \phi_{z,t,w}^{\sum_{(c,d)\in E_t} \Delta_{z,c,d,w}-1} \quad (1)$$

$$\Delta_{z,c,d,w} = \sum_{w'}^{\kappa} sim(w, w')\beta_{z,c,d}\widehat{\phi}_{z,c,w'}\widehat{\phi}_{z,d,w'} \quad (2)$$

Thus the calculation of PoD with $\{\beta_{c,d}, \phi_c, \phi_d\}_{(c,d)\in E_t}$ has been introduced in Equation 1. Given RRG$= (s, \widehat{V}, \widehat{E})$ after the posts agglomeration, the generative process for each post collection $note(t)$ in the PPM is as follows:

1. For the vocabulary, establish the similarity matrix $\mathfrak{W}_{sim}$
2. For each topic $z, 1 \le z \le K$: Draw
   $\phi_{z,t} \sim PoD(\{\beta_{c,d}, \phi_c, \phi_d\}_{(c,d)\in E_t}, \mathfrak{W}_{sim})$
3. For each post $p, p \in note(t)$:
   - Draw $\theta_{p,t} \sim Dir(\alpha_t)$
   - For each term index $i, 1 \le i \le W$
     - Draw $z_{i,p,t} \sim Mult(\theta_p)$
     - Draw $w_{i,p,t} \sim Mult(\phi_{z_{i,p,t},t})$

Figure 2 shows a graph model representation of the PPM, where shaded and unshaded nodes indicate observed and latent variables, respectively. For each $note(t)$ in RRG$= (s, \widehat{V}, \widehat{E})$ the posterior distribution is intractable because of the unknown relation between ϕ_t and θ_t. To find an approximate inference method, we use the Gibbs EM algorithm [12] for inference that optimizes the edge parameters $\{\beta_{c,d}\}_{(c,d)\in E_t}$ and the Dirichlet prior α_t during each sampling iteration.

4. THREAD SUMMARIZATION

For each post collection $note(t)$ in RRG$= (s, \widehat{V}, \widehat{E})$, there are two parametric matrix Θ_t and Φ_t after Gibbs EM sampling, which reflect the topic-post distribution and the word-topic distribution, respectively. *i.e* $P(z|p) = \theta_{z,p}$; $P(w|z_t) = \phi_{z,t,w}$. Based on these distributions, a naive summarization method is proposed as the basic algorithm. Then we introduce the markov random walk(MRW) strategy and the mutual reinforcement effect between posts and sentences to our WFTS process.

[1]http://www.cogs.susx.ac.uk/users/drh21

4.1 Basic Algorithm

We can calculate the probability of each topic by summing and normalizing mixture components over topics for all of posts in the thread. By assuming that each post's generation is assigned the same probability, we have:

$$P(z) = \sum_p P(z|p)P(p) = \sum_p \frac{P(z|p)}{|V|} \quad (3)$$

where $|V|$ is the amount of all posts. Equation (3) actually indicates the synthetic salience of topic z all over the dynamic conversation. Meanwhile, we assume that each sentence is generated by a mixture of topics. Thus given sentence s_j, we have:

$$P(s_j) = \sum_z P(s_j|z)P(z) = \prod_{w\in s_j} \sum_z P(w|z)^{n(w)}P(z) \quad (4)$$

where $P(z)$ is derived from Equation (3), and $n(w)$ reflects the term-frequency of w in sentence s_j. Based on Equation (4), each sentence's probability in the thread is assigned. We rank the sentences by the value of probabilities from Equation (4).

4.2 Topic-sensitive Random Walks

The basic algorithm is a little unrealistic that each post's generation is assigned the same probability $1/|V|$. Illuminated by the idea of the Topic-sensitive PageRank [4], we utilize a new WFTS method with the markov random walks(MRW), where each sentence is scored by the "votes" from other sentences based on an complete weighted undirected graph.

Formally, let $G_S = (S, E_S)$ be the complete undirected graph with S nodes and E_S edges, where there are $|S|$ sentences in the thread and each edge $(s_i, s_j) \in E_S$ has an affinity weight that reflects the similarity between sentence s_i and $s_j, i \ne j$. After topic modeling, each sentence s has the topical feature vector $D_{s,z} = \{P(w|z, s)\}_{w=1}^{W}$, that:

$$P(w|z, s) = \begin{cases} \phi_{z,t,w} & w \in s \\ sim(w, w') \cdot \phi_{z,t,w} & w' \in s \\ 0 & else \end{cases} \quad (5)$$

where we utilize the $\mathfrak{W}_{sim}$ to reduce the sparsity when $sim(w, w') \in \mathfrak{W}_{w'}$. Using the Jensen-Shannon Divergence between $D_{s_i,z}$ and $D_{s_j,z}$, the topical divergence is given:

$$Di_{JS}(s_i, s_j, z) = \frac{1}{2}\left(KL(D_{s_i,z} \| M) + KL(D_{s_j,z} \| M)\right) \quad (6)$$

where M is the average of the two probability vectors, $KL(D \| M)$ is the Kullback-Leibler Divergence.The divergence is transformed into similarity measure by:

$$sim(s_i, s_j) = \frac{1}{K}\sum_z 10^{-\delta Di_{JS}(s_i,s_j,z)} \quad (7)$$

So we have the affinity matrix SIM for all S sentences, $SIM_{i,j} = sim(s_i, s_j)$. After the row-normalization for SIM, transition probability matrix $\widehat{SIM}$ is built that each row $\left\|\widehat{SIM}_i\right\|_1 = 1$. Adding smoothing vector $1/|S|$, salience score $Sco(s_j)$ for each sentence s_j is deduced by:

$$Sco(s_i) = \mu \sum_{i\ne j} \widehat{sim}(s_i, s_j) \cdot Sco(s_j) + \frac{(1-\mu)}{|S|} \quad (8)$$

where μ is the damping factor usually set to 0.85 as in PageRank algorithm. The salience score vector Sco for all sentences are set to 1 at the iteration beginning. Usually the convergence of the iteration algorithm is achieved when the difference between the scores computed at two successive iterations for any sentences falls below a given threshold. Sentences are ranked according to their salience scores converged in Sco.

4.3 Post's Influence to Summarization

All of the naive scoring and the MRW-based scoring ignore the effect of the posts to sentences' salience. Intuitively, a post is important if (1) it includes the important sentences; (2) it associates to other important posts. On the other hand, a sentence is important if (1) it appears in an important post; (2) it associates to the other important sentences. Thereby, mutual reinforcement(MR) [16, 15] between sentences and posts is employed to calculate the salience of sentences based on the MRW-based scoring procedure.

Let Pco denote salience scores for all posts in the thread, we construct the salience score vector $Rco = [Pco, Sco]^T$ for the MR framework. Each post p can be represented in the form of: (1) a vector of post-topic distributions $D_p = \{P(z|p)\}_{z=1}^{K}$; (2) a $K \times W$ matrix *KW* where item $KW_{i,j}$ denotes the topic-word distribution $P(w_j|z_i, p)$. $P(w_j|z_i, p)$ is defined similarly from Equation (5). The first representation is used for the post to post affinity calculation whereas the second is used for the affinities between posts and sentences. In the MR framework, the key problem is to construct the blocking matrix PSM including the affinities among objects (sentences and posts).

$$PSM = \begin{bmatrix} PIM & PS \\ SP & SIM \end{bmatrix} \tag{9}$$

where PIM denotes the post-post affinity matrix. Item $PIM_{i,j}$ reflects the similarity measure between post p_i and post p_j:

$$\begin{aligned} PIM_{i,j} &= sim(p_i, p_j) = 10^{-\delta Di_{JS}(p_i,p_j)} \\ Di_{JS}(p_i, p_j) &= \frac{1}{2}\left(KL(D_{p_i} \| M) + KL(D_{p_j} \| M)\right) \end{aligned} \tag{10}$$

Meanwhile, both affinity matrices PS and its transposed matrix of SP reflect the similarities between posts and sentences. For each item (i, j) in PS, we have:

$$PS_{i,j} = sim(p_i, s_j) = \frac{1}{K}\sum_{z} 10^{-\delta Di_{JS}(p_i,s_j,z)} \tag{11}$$

After the row-normalization for the block matrix PSM, the ranking of posts and sentences can be iteratively derived from the MR framework by:

$$\overrightarrow{Rco}^{(k+1)} = \mu PSM \cdot \overrightarrow{Rco}^{(k)} + \frac{(1-\mu)}{|V+S|} \tag{12}$$

The ranking process of the Equation (12) is the same way with the Equation (8) in section 4.2. After achieving the convergence of the iteration, both posts and sentences in the thread have been ranked by the salience scores Rco.

For summary extraction, we establish a greedy algorithm to detect the semantic orthogonality among summary sentences. Given selected sentences set S_s and the current candidate sentence s in one step of extraction, s will not be selected unless the maximal semantic similarity between s and S_s, calculated by Equation (7), is below one threshold.

5. EXPERIMENT DESIGN

5.1 Data Set and Evaluation Metric

A new data set is opt to be created because there is no existing benchmark data set for evaluating the WFTS task. In this paper, Apple Discussion[2] and Slashdot[3] are used as our data sources. We obtained threads from our two data sources with the limitation that each thread has no less than 5 posts. To evaluate our summary method performances with different thread volume, we classify threads into 4 different thread volume intervals. Since 15 posts constitute one web page in Apple Discussion, we set intervals as $[5, 15]$, $[16, 30]$, $[31, 45]$ and $[46, \infty]$, with the limitation that the amount of threads in each interval is equal. 100 threads (50 from Apple and 50 from Slashdot) on each interval are crawled respectively.

To generate the evaluation references, 4 human assessors were asked to summarize the content individually. The final evaluation score for a WFTS strategy is on average of all scores using each referenced summary. We use ROUGE toolkit[4] to measure our proposed WFTS methods. ROUGE-1 (Recall against unigram), ROUGE-2 (Recall against bigram), and ROUGE-L (Recall against longest common subsequence) are chosen for the WFTS performance measure. In the ROUGE settings we use Porter Stemming algorithm to stem the words to their root form.

5.2 Experimental Results

5.2.1 Strategy Selection

In section 3, we agglomerate posts from RRT= (r, V, E) structure into RRG= $(s, \widehat{V}, \widehat{E})$ structure. To identify how this strategy enhance the summarization performance by reducing the *text sparseness* during the PPM process, we compare the results of the PPM-based WFTS using posts agglomeration with the one that ignores the posts agglomeration. Based on the PPM, we denote PPM-S as the basic algorithm in section 4.1, PPM-ST as the topic-sensitive MRW-based method in section 4.2, and PPM-STP as the MR-based ranking method using mutual reinforcement in section 4.3. These methods are measured in terms of ROUGE metrics for 200 summary length when stop-words are kept.

Table 2 presents us performances with and without the posts agglomeration. We can find for each WFTS algorithm the results after merging posts outperform results without posts agglomeration. The difference is obvious so that it is easy to conclude the posts agglomeration is worthwhile even though this may generate a little information loss for reply-relations. As shown in Table 2, both PPM-STP and PPM-ST are able to produce better results than PPM-S that seems a little naive for WFTS calculation. However, the tradeoff between the low complexity cost and competitive ROUGE performances make PPM-S still valuable in practice.

5.2.2 Overall Performance

Several baselines, including both methods using other topic models and some effective algorithms in the field of document summarization [3, 17, 10], are introduced for comparison. Among the topic models, we choose **Latent Dirichlet Allocation** (LDA) and **Dynamic Mixture Models** (DMMs) as comparisons. All we proposed WFTS algorithms in section 4 are based on topic models, thus we compare different performances between the PPM and these models using the MR-based WFTS algorithm in section 4.3.

[2]http://discussions.info.apple.com

[3]http://slashdot.org

[4]version 1.5.5 is used here

Table 1: Overall summarization results in terms of ROUGE

Stop-word Kept								
Length	Constraint Set	PPM-STP	DMMs	LDA	HIERSUM	ST-C	MEAD	NIST-B
200	ROUGE-1	**0.4695**	0.4629	0.4606	0.4516	0.4176	0.4172	0.3753
	ROUGE-2	**0.1742**	0.1712	0.1701	0.1714	0.1339	0.1384	0.1021
	ROUGE-L	**0.4134**	0.4119	0.4062	0.4074	0.3723	0.3661	0.3338
400	ROUGE-1	**0.5133**	0.5012	0.4907	0.5006	0.4623	0.4556	0.4372
	ROUGE-2	**0.2266**	0.2156	0.2027	0.2069	0.1882	0.1891	0.1584
	ROUGE-L	**0.4653**	0.4517	0.4426	0.4462	0.4271	0.4055	0.3983
Stop-word Removed								
Length	Constraint Set	PPM-STP	DMMs	LDA	HIERSUM	ST-C	MEAD	NIST-B
200	ROUGE-1	**0.3605**	0.3587	0.3422	0.3581	0.323	0.3256	0.3017
	ROUGE-2	**0.1682**	0.1634	0.1574	0.1602	0.1486	0.1489	0.1269
	ROUGE-L	**0.3214**	0.3206	0.3103	0.3126	0.2925	0.2901	0.2767
400	ROUGE-1	**0.4112**	0.3918	0.3871	0.3910	0.3742	0.3667	0.3591
	ROUGE-2	**0.2053**	0.1916	0.1914	0.2015	0.1804	0.1833	0.1665
	ROUGE-L	**0.3641**	0.3472	0.3378	0.3511	0.3297	0.3172	0.3082

Table 2: Post-merging's influence for WFTS

WFTS Algorithms	ROUGE-1	ROUGE-2	ROUGE-L
Without Post-Merging			
PPM-S	0.4072	0.1622	0.3587
PPM-ST	0.4175	0.1648	0.3604
PPM-STP	0.4282	0.1681	0.3821
After Post-Merging			
PPM-S	0.4582	0.1731	0.4052
PPM-ST	0.4594	0.1741	0.4082
PPM-STP	**0.4695**	**0.1742**	**0.4134**

For all 400 threads, we calculate results of all baselines in terms of ROUGE-1, ROUGE-2 and ROUGE-L using 2 methods, keeping stop-words and removing stop-words. For each method, we evaluated performances for 200 and 400 length summary. As shown in Table 1, PPM-STP results in obvious improvements over the others: For 200 length summary, PPM-STP achieves an increase of 1.9%, 2.4% and 1.8% over LDA in terms of ROUGE-1, ROUGE-2 and ROUGE-L respectively. For 400 length summary, PPM-STP gives an increase of 4.6%, 11.8% and 5.1% over LDA when stop-words are kept. We further compare PPM with the WFTS using DMMs. By and large, for 200 length summary PPM-STP offers relative performance improvements of 1.4%, 1.8% and 0.4%, respectively, in the ROUGE-1, ROUGE-2 and ROUGE-L measures as compared to the DMMs; while the relative improvements are 2.4%, 5.1% and 3.0% in the same measurements for 400 length summary case. Thus although only slight improvement happens when the summarization length is relatively small, dissimilarities between PPM and DMMs rises with the increase of the summary length. A natural explanation to the fact is Dynamic Mixture Models (DMMs) cannot capture reply-relations existing in each thread.

5.2.3 Impact of Thread Volume to Results

To precisely illustrate the performances for various thread volume, the impact of the thread volume is evaluated in Figure 7 by comparisons of WFTS performances on each interval respectively. In Figure 7, (a) and (b) illustrate the ROUGE-1 and ROUGE-2 scores for 200 length summary whereas (c) and (d) reflect the ROUGE metrics for 400 length summary. In Figure 7(a) and (c), PPM, DMMs and LDA have similar performances in terms of ROUGE-1 when posts number ≤ 15. However, with the increase of thread volume, ROUGE-1 from the LDA and DMMs decreases rapidly while the PPM keep relatively stable. Shown by Figure 7(b) and (d), PPM has obviously better performance in terms of ROUGE-2 than others for each interval. All these improvements reflect the effectiveness for the thread summarization task to capture the reply-relations in each thread.

5.2.4 Impact of the Factor κ to Results

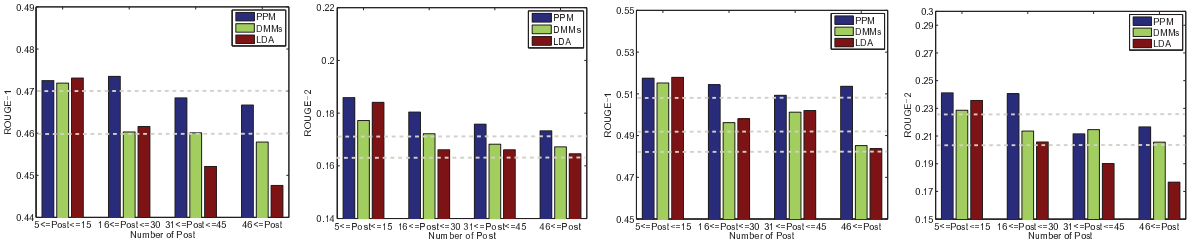
In section 3, given each thread's vacabulary W we establish a $W \times \kappa$ matrix including the background semantic similarities between each word and its top κ most similar words. The larger κ, the more dependencies for one word propagated from other similar words. Figure 8 (a) to (c) shows the ROUGE-1, ROUGE-2 and ROUGE-L curves for different WFTS strategies under the PPM, respectively. As shown in Figure 8, when κ is greater than 1, better performances are observed for both PPM-based strategies. This can suggest that semantic similarities among words improve the performance of each PPM-based WFTS strategy. Meanwhile, we can observe that each metrics (ROUGE-1, ROUGE-2 and ROUGE-L) keeps on increasing till the κ comes to 6, after that value curves for the PPM-based methods begin to decline.

6. CONCLUSION

In this paper, we propose the task of Web Forum Thread Summarization (WFTS) to help people understand the thread in web forums rapidly. Web Forum Thread Summarization works by extracting a group of sentences from all posts in one thread to constitute the summary. Based on a hierarchical Bayesian model that track all dynamic topics through the threaded discussion, we employ the markov random walk strategy and the mutual reinforcement to the sentence scoring progressively. Final summary is generated from a greedy sentence extraction process that keep the semantics orthogonality. In experiments, we establish our data set from popular web forums and verify the effectiveness of our WFTS strategies, which give us remarkable performances comparing with other effective baselines.

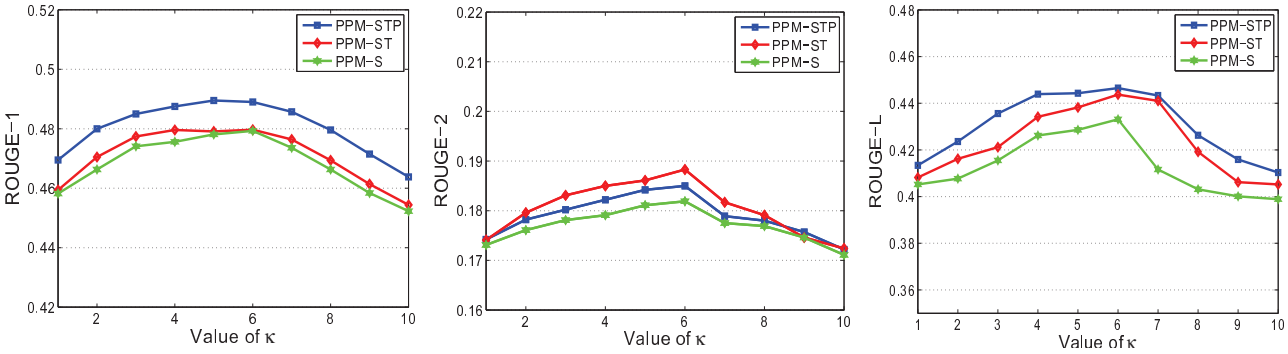
7. ACKNOWLEDGEMENT

This work is supported by the Natural Science Foundation of China (60970047, 60903108), IIFSDU (2009TB016) and the Natural Science Foundation of Shandong Province (Y2008G19).



(a) ROUGE-1, Length=200 (b) ROUGE-2, Length=200 (c) ROUGE-1, Length=400 (d) ROUGE-2, Length=400

Figure 2: Performances of methods using topic models on 4 volume intervals in terms of ROUGE metrics



(a) ROUGE-1 (b) ROUGE-2 (c) ROUGE-L

Figure 3: Impact of κ in terms of ROUGE metrics for 200 length summary

8. REFERENCES

[1] D. Blei and J. Lafferty. Dynamic topic models. In *ICML*, 2006.

[2] S. Dennis. On the hyper-Dirichlet type 1 and hyper-Liouville distributions. *Communications in Statistics-Theory and Methods*, 1991.

[3] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *HLT-ACL*, 2009.

[4] T. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 2003.

[5] X. Hu, N. Sun, C. Zhang, and T. Chua. Exploiting internal and external semantics for the clustering of short texts using world knowledge. In *CIKM*, 2009.

[6] T. Iwata, S. Watanabe, T. Yamada, and N. Ueda. Topic tracking model for analyzing consumer purchase behavior. In *IJCAI*, 2009.

[7] A. Jain, M. Murty, and P. Flynn. Data clustering: a review. *ACM computing surveys*, 1999.

[8] C. Lin, J. Yang, R. Cai, X. Wang, and W. Wang. Simultaneously modeling semantics and structure of threaded discussions: a sparse coding approach and its applications. In *SIGIR*, 2009.

[9] Y. Liu, A. Niculescu-Mizil, and W. Gryc. Topic-link LDA: joint models of topic and author community. In *ICML*, 2009.

[10] D. Radev, H. Jing, M. Sty, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 2004.

[11] Z. Ren, J. Ma, G. Wang, C. Cui, and X. Han. Dynamically modeling semantic dependencies in web forum threads. In *WI-IAT*, 2011.

[12] H. Wallach. Topic modeling: beyond bag-of-words. In *ICML*, 2006.

[13] C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. In *UAI*, 2008.

[14] X. Wang and A. McCallum. Topics over time: a non-markov continuous-time model of topical trends. In *SIGKDD*, 2006.

[15] F. Wei, W. Li, Q. Lu, and Y. He. Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In *SIGIR*, 2008.

[16] H. Zha. Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In *SIGIR*, 2002.

[17] L. Zhou and E. Hovy. Digesting virtual geek culture: The summarization of technical internet relay chats. In *ACL*, 2005.