Customized Organization of Social Media Contents using Focused Topic Hierarchy

Xingwei Zhu[1]*, Zhao-Yan Ming[2]†, Yu Hao[1], Xiaoyan Zhu[1], Tat-Seng Chua[2]
[1]State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Sci. and Tech., Tsinghua University
[2]Department of Computer Science, School of Computing, National University of Singapore, Singapore
etzhu192@hotmail.com, mingzhaoyan@nus.edu.sg, haoyu@mail.tsinghua.edu.cn, zxy-dcs@tsinghua.edu.cn, chuats@nus.edu.sg

ABSTRACT

With the popularity of social media platforms such as Facebook and Twitter, the amount of useful data in these sources is rapidly increasing, making them promising places for information acquisition. This research aims at the customized organization of a social media corpus using focused topic hierarchy. It organizes the contents into different structures to meet with users' different information needs (e.g., "iPhone 5 problem" or "iPhone 5 camera"). To this end, we introduce a novel function to measure the likelihood of a topic hierarchy, by which the users' information need can be incorporated into the process of topic hierarchy construction. Using the structure information within the generated topic hierarchy, we then develop a probability based model to identify the representative contents for topics to assist users in document retrieval on the hierarchy. Experimental results on real world data illustrate the effectiveness of our method and its superiority over state-of-the-art methods for both information organization and retrieval tasks.

Categories and Subject Descriptors

H.0 [**Information Systems**]: General; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

customized information organization; focused topic hierarchy; information need; social media corpus

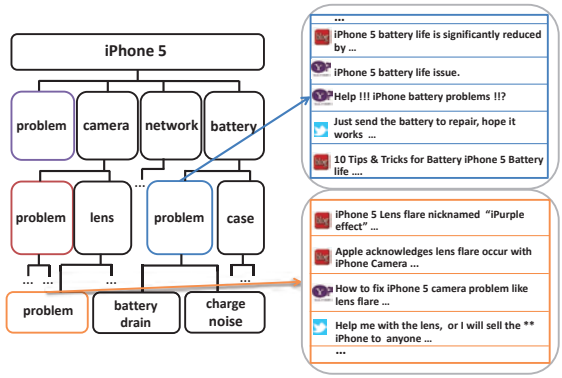

Figure 1: The focused topic hierarchy for "iPhone 5 problem", where different representative contents are collected for the four specific nodes of problem (highlighted with colors) accordingly.

1. INTRODUCTION

With the rapid growth of contents in social media sources like Twitter[1], Facebook[2] and Yahoo! Answers[3], users are able to access huge amounts of data from these sources to find their desired information. However, the information overload and noise in social media contents limit their potential usefulness. To tackle this problem, organizing the social media contents into a general topic hierarchy has been shown to be an effective solution [23] [24] [29], in which the user is presented with an in-depth overview of his/her desired topics in the form of hierarchically organized document clusters.

The information need of different users may vary greatly, ranging from an overall description of a topic, e.g., "iPhone 5", to contents about its particular aspects, for which a general topic hierarchy may not be applicable. For example, given a user information need on "iPhone 5 problem", a focused topic hierarchy as shown in Figure 1 should be more preferable, as compared to a general hierarchy of iPhone 5 (e.g., the one in Wikipedia[4] which contains subsections like "history" and "sale"), Moreover, even if the topic hierarchy is presented, it could still be time-consuming for users to

* This work was done when the first author was a visiting student in National University of Singapore.
† Corresponding author.

CIKM'14, November 3–7, 2014, Shanghai, China.

http://dx.doi.org/10.1145/2661829.2661896.

[1]https://twitter.com/
[2]https://www.facebook.com/
[3]http://answers.yahoo.com/
[4]http://en.wikipedia.org/wiki/IPhone_5

manually find useful contents in it. To this end, the ability to identify representative contents for each node on the hierarchy is necessary. For example, the representative contents for the node problem under battery in Figure 1 should focus only on iPhone 5's battery problem.

Generally, there are two major research challenges to organize information using focused topic hierarchies.

- *General Taxonomy vs. Focused Hierarchy*: As the key difference between our approach with the general topic hierarchy construction methods like [22][26] [29], a focused topic hierarchy should organize the given corpus into different structures to meet with different users' information needs. To this end, a novel method is required for the customized topic hierarchy construction.
- *General Clustering vs. Customized Organization*: Due to the information overload in social media, instead of providing users raw document clusters as in [16][24][27], we need further identify the representative contents in them, for which the measure of representativeness should be properly designed.

In this paper, we propose a novel method for the customized organization of social media contents using focused topic hierarchies. Given the user's information need, e.g., "iPhone 5 problem", we first use a propagation algorithm to collect the potentially useful topics, e.g., battery and solution. Next, we devise a function to estimate the likelihood of a topic hierarchy and use it to integrate the obtained topics into a topic hierarchy that fits both the given social media corpus and the user's personal information need. Finally, in order to further assist users to search on the hierarchy, we propose a probability based ranking model to identify the representative contents for topic nodes using both content and source based features of documents. To summarize, the main contributions of our approach are two fold:

- **Focused topic hierarchy construction**: We introduce the focused topic hierarchy to provide users a customized view of the social media contents, in which the information need is seamlessly incorporated into the topic hierarchy construction process.
- **Customized representative content selection**: We develop a probability model to identify the representative content for each topic node on the hierarchy, which enables fast information retrieval on the hierarchically organized social media corpus.

The rest of this paper is organized as follows. In section 2 we introduce our related work. In section 3 we will formulate our research problems. In section 4 we describe our proposed framework and its three modules. In section 5 we evaluate the performance of the proposed method using real world data and compare it with state-of-the-art methods. Finally, we summarize the paper and outline the future work in section 6.

2. RELATED WORK

2.1 Information Organization

To organize the information in a text corpus efficiently has been studied by many researches [2] [5][12][27][29] before. Compared to the early works [20][27] that proposed to split the corpus into shallow clusters, many recent approaches tend to organize the corpus into cluster hierarchies which could provide users with more in-depth view of the corpus and improve the search experience. Specifically, methods like agglomerative hierarchical clustering [6] [9][28], hierarchical LDA [2][15][23] were all adopted in this task. Recently, approaches like [1][16] proposed to further improve the performance of common hierarchical clustering algorithms using partially known hierarchies. Besides, researchers [3][24] also used time-line analysis techniques to enable evolutionary hierarchical clustering on ever-changing contents.

2.2 Topic Hierarchy Generation

Constructing concepts into taxonomies is useful for the organization of the underlying knowledge in a given domain. Previous approaches like [17][19][21][25] used is-a relations between terms to connect them into taxonomies. Besides, multi-branch clustering algorithms [4][13][22] were also proposed for hierarchy generation. In [22], a generative model was used to build up a topic hierarchy by splitting the topic set into smaller clusters as its children on the hierarchy iteratively. In [13], the authors proposed to use the bayesian rose tree clustering algorithm to automatically determine the depth and width of the resultant topic hierarchy. Besides, as shown in [14], linked data in Freebase were also shown to be helpful in identifying the underlying structures of topics. In a recent approach [8], word embeddings technique was also adopted in hierarchy generation tasks.

Similar to our proposed method, some researches [26][29] attempted to generate topic hierarchies from social media contents. However, since they didn't consider the users' information need as an input, they could construct only one general topic hierarchy for a given corpus. As a result, for a given social media corpus, they were not able to provide satisfactory results for users with various information needs.

In this research, we aim to realize the customized topic hierarchy generation and use it for information organization and retrieval tasks.

3. PROBLEM FORMULATION

3.1 Social Media Corpus

Given a social media source set $\mathcal{S} = \{s_0, s_1, ...\}$, we define a social media corpus $\mathcal{D} = \bigcup_{s_i} \mathcal{D}^{s_i} = \{d_1, d_2, ...\}$, in which $\mathcal{D}^{s_i}$ indicates the set of social media contents crawled from the source s_i and $d_i = (c_i, s_i)$, in which c_i is the document's content and s_i indicates its source. Next, we define the topic set $\mathcal{T}(\mathcal{D})$ for $\mathcal{D}$ as $\mathcal{T}(\mathcal{D}) = \{t_1, t_2, ...\}$, in which t_i is a noun phrase and indicates a subtopic in $\mathcal{D}$. For example, if the contents in $\mathcal{D}$ are all about iPhone 5, the corresponding $\mathcal{T}(\mathcal{D})$ may contain subtopics like battery and price.

3.2 User Information Needs

We formulate a *user information need* q, or *information need* for short, as $q = \{t_r, t_1, t_2, ...\}$, in which t_r indicates the root topic, e.g., "iPhone 5", that the user is interested in and t_i, $i = 1, 2, ...$ is a subtopic of t_r, e.g., battery or price of iPhone 5 on which the specific information is requested by the user.

Generally, user information needs can express the search intents of many keyword queries. Moreover, queries like "iPhone 5 problem" can even be directly interpreted into in-

formation needs, e.g., {iPhone 5, problem}. However, the process of transforming between the queries and information needs is beyond the scope of this paper. For the sake of consistency, we use the information need as the input of our proposed framework in the rest of this paper.

3.3 Focused Topic Hierarchy

In this research, a focused topic hierarchy is used to organize a social media corpus based on a user's information need. Given an information need $q = \{t_r, t_1, t_2, ...\}$ and a social media corpus $\mathcal{D}(t_r)$ in which all the contents are relevant to the root topic t_r, a focused topic hierarchy $\mathcal{H}_q$ consists of the following two components:

- The topic node set $\mathcal{V}_q$, in which: (1) each node v represents a topic $t(v) \in \mathcal{T}(\mathcal{D}(t_r))$ and is on a topic node tree rooted at v_r where $t(v_r) = t_r$, and (2) for all $v \in \mathcal{V}_q$, $t(v)$ must be relevant to the given information need.
- The topic content set $\mathcal{C}_q$, in which: (1) for each $v \in \mathcal{V}_q$, its relevant content set $c(v) \in \mathcal{C}_q$ includes the most relevant N documents to the topic $t(v)$ in $\mathcal{D}(t_r)$, and (2) all the documents in $c(v)$ must also be relevant to the given information need.

Generally, a focused topic hierarchy $\mathcal{H}_q$ can organize the social media contents for the information need q with hierarchically structured topics. For example, given the focused topic hierarchy for {iPhone 5, problem}, its topic node set may contain topics such as battery drain and problem, where battery drain is a subtopic of problem and the blog "iPhone 5 battery life is significantly reduced by ..." will be assigned to battery drain as its relevant content. Using the focused topic hierarchy, users can directly obtain their required information by clicking through the connected topics on the hierarchy and reading the linked social media contents.

However, to generate a proper focused topic hierarchy that best meets an information need is not trivial. Generally, there are three research problems that we need to address in this study:
(1) How to discover the potentially useful topics for an information need?
(2) How to obtain the optimal topic structure to organize the users' desired information?
(3) How to distinguish the representative contents for topics that directly meet the users' requirements?

4. METHODOLOGY

4.1 Framework

In order to tackle the three problems, in this paper we introduce a three step pipeline as shown in Figure 2, in which at each step we solve one of the problems sequentially. Specifically, given an information need $q = \{t_r, t_1, t_2, ...\}$ and the social media corpus $\mathcal{D}(t_r)$, we first collect q's relevant topics from $\mathcal{T}(\mathcal{D}(t_r))$. Next we construct the optimal topic hierarchy $\mathcal{H}_q$ for the given information need using the obtained topics. Finally, we assign each topic node on $\mathcal{H}_q$ with its most representative documents in $\mathcal{D}(t_r)$ as its relevant content set.

In next subsection, we will first show how we collect the social media corpus and model it using automatically extracted topics. The followed three subsections will describe the details of each module in Figure 2.

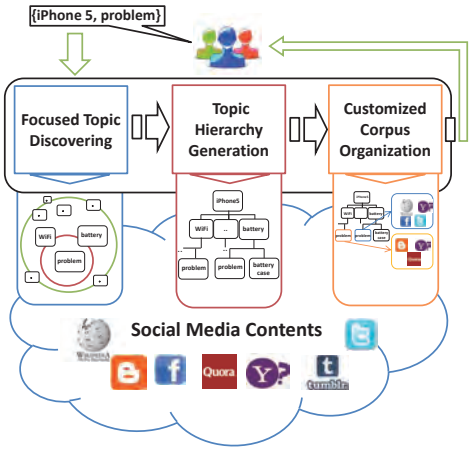

Figure 2: The proposed framework for the customized organization task.

4.2 Corpus Collection and Topic Modeling

The volume of social media contents increases rapidly. In this approach, we obtain topics and topic relations from the crawled social media contents incrementally and utilize them to model the resultant corpus. It is worth noting that the following corpus modeling process can be done off-line since it is not dependent on any specific user information need.

4.2.1 *Topic Extraction*

From a social media corpus $\mathcal{D}$, we use a TF-IDF based keyword extraction method to generate the topic set $\mathcal{T}(\mathcal{D})$. Next, for each topic t in $\mathcal{T}(\mathcal{D})$, we further use a salience score $\delta(t) \in \mathcal{R}_+$ to measure its importance. Generally, a topic with higher salience score should be more important in the corpus. For example, we may have $\delta(\text{wifi}) = 0.59$ and $\delta(\text{thing}) = 0.12$ on the corpus for iPhone 5, indicating that wifi is more important than thing here. In practice, we use the method proposed in [27] to estimate the topic salience scores with multiple corpus-derived features.

4.2.2 *Topic Relations*

Topics in $\mathcal{T}(\mathcal{D})$ could be correlated with each other through various relations. In the paper, we investigate the use of two topic relations, i.e., the topic relevance and subtopic relation.

The topic relevance $\theta(t_i, t_j) \in [0, 1]$ indicates the strength of semantic relatedness between two topics t_i and t_j. For example, we may have $\theta(\text{network}, \text{wifi}) = 0.65$ and $\theta(\text{screen}, \text{wifi}) = 0.17$, indicating that network is more relevant to wifi than screen. In this research, we adopt the method in [10] to estimate the topic relevance scores based on topic co-occurrence.

The subtopic relation strength $\mu(t_m, t_n) \in [0, 1]$, indicates the likelihood that t_n is a subtopic of t_m. For example, we may have $\mu(\text{network}, \text{wifi}) = 0.89$ and $\mu(\text{wifi}, \text{network}) = 0.07$, indicating that wifi is more likely to be a subtopic of network. We utilized the method in [29] to estimate $\mu(t_i, t_j)$ using heuristic rules.

4.3 Graph-based Focused Topic Discovering

Given an information need $q = \{t_r, t_1, t_2, ...\}$ and the social media corpus $\mathcal{D}(t_r)$, our first task is to collect a subset $\mathcal{T}_q \subset \mathcal{T}(\mathcal{D}(t_r))$, in which only the topics that fit the specific information need are included. We first use a simple

co-occurrence based method to construct a raw candidate topic set $\mathcal{T}_q^0$, which includes all the topics in $\mathcal{T}(\mathcal{D}(t_r))$ that co-occur with the subtopics in q. Next, we propose a graph-based label propagation algorithm to refine $\mathcal{T}_q^0$ using topics' semantic relatedness.

Regarding topics in $\mathcal{T}(\mathcal{D}(t_r))$ as vertexes, we construct a topic graph by bridging each topic pair t_i and t_j with a undirected edge weighted by their topic relevance score $\theta(t_i, t_j)$. We first assign an initial weight $w(t)$ for each topic t on the graph as follows,

$$w(t) = \begin{cases} \delta(t) & , \quad t \in \mathcal{T}_q^0 \\ 0 & , \quad \text{otherwise} \end{cases} \tag{1}$$

recall that $\delta(t)$ is the salience score of the topic t in the given corpus.

Based on the resultant topic graph, Equation 2 is used to propagate the topic weights between tightly related topics.

$$\begin{aligned} w(t) &= \hat{w}(t) + \Delta(t) \\ &= \hat{w}(t) + \sum_{t \in \Omega_k(t_i)} \sigma(t_i, t) \cdot w(t_i) \end{aligned} \tag{2}$$

$$\hat{w}(t) = \begin{cases} \delta(t) & , \quad \Delta(t) > 0 \text{ and } w(t) = 0 \\ w(t) & , \quad w(t) \neq 0 \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{3}$$

$$\sigma(t_i, t_j) = \begin{cases} \dfrac{\theta(t_i, t_j)}{\sum\limits_{t_s \in \Omega_k(t_i)} \theta(t_i, t_s)} & , \quad t_j \in \Omega_k(t_i) \\ 0 & , \quad \text{otherwise.} \end{cases} \tag{4}$$

In Equation 2, $\hat{w}(t)$ and $\Delta(t)$ indicate the impact of the topic itself and that from its neighbors in the propagation, respectively. Note that when estimating $\Delta(t)$, we only retain the topic t's influence to its nearest k neighbor set, i.e., $\Omega_k(t)$, so that the propagation can be limited to a dense sub-graph, hence keeping out the potential noisy topics in $\mathcal{T}(\mathcal{D}(t_r))$.

For the same purpose, in Equation 3, a zero-weighted topic t_i can obtain non-zero initial weight only if there is a topic t_j where $t_i \in \Omega_k(t_j)$ and $w(t_j) \neq 0$. In Equation 4, the transmitting probability from t_i to t_j, i.e., $\sigma(t_i, t_j)$ is non-zero only if $t_j \in \Omega_k(t_i)$.

Finally, we rank the topics using their final topic weights and collect the top ranked non-zero topics to form the focused topic set $\mathcal{T}_q$.

4.4 Information Need-Aware Topic Hierarchy Construction

4.4.1 Likelihood of Topic Hierarchy

For an information need $q = \{t_r, t_1, t_2, ...\}$, there are various ways to integrate the topics in $\mathcal{T}_q$ into a topic hierarchy $\mathcal{H}$. In this paper, we propose to use the following function $\mathcal{L}(\mathcal{H}) \in \mathcal{R}_+$ to measure the likelihood that the hierarchy H can fit a given information need.

$$\mathcal{L}(\mathcal{H}) = \sum_{\substack{v_i, v_j \text{if} \\ e(v_i, v_j) \text{exists}}} w(e(v_i, v_j)) \tag{5}$$

in which $e(v_i, v_j)$ is an edge from v_i to v_j on $\mathcal{H}$, indicating that v_j is a subtopic of v_i on the hierarchy. $w(e(v_i, v_j)) \in \mathcal{R}_+$ is the weight of $e(v_i, v_j)$, indicating the likelihood of this edge.

Given Equation 5, if all the edge weights on the hierarchy are properly estimated, we can find the optimal focused topic hierarchy straight-forwardly. In this research, we introduce the following two assumptions to measure an edge's weight, e.g., $w(e(v_i, v_j))$ from two perspectives.

The information need's perspective: $w(e(v_i, v_j))$ is only relevant to the importance of v_i and v_j for the given information need and their subtopic relation strength. For example, given the information need {iPhone 5 device}, the edge e(camera, lens) should be higher weighted than e(camera, photo) since: (1) the subtopic lens is more relevant to the information need, and (2) the subtopic relation between camera and lens is also stronger.

The taxonomy structure's perspective: $w(e(v_i, v_j))$ is only relevant to the fitness of the edge $e(v_i, v_j)$ on the current topic hierarchy. For example, For the edge e(problem, battery drain), if it is part of a path "network → problem → battery drain" on the hierarchy, it should be weighted low since battery drain is irrelevant to network. On the contrary, for the path "battery → problem → battery drain", this edge should be weighted high.

In order to combine the two assumptions, we define a *path* $L = \bigcup_{k=0}^{|L|-1} e(v_k, v_{k+1})$, in which $v_0 = v_r$, i.e., the root of the hierarchy and $v_{|L|}$ is any non-root node on the hierarchy. We then estimate $w(L)$, i.e., the *path weight* of L as follows.

$$w(L) = \frac{\sum\limits_{k=0}^{|L|-1} w(t_k) \cdot w(t_{k+1}) \cdot \mu(t_k, t_{k+1})}{|L|} \tag{6}$$

in which $t_k = t(v_k)$ and $t_{k+1} = t(v_{k+1})$.

We can see that Equation 6 involves both the information need-related factors (i.e., $w(t_k)$, $w(t_{k+1})$ and $\mu(t_k, t_{k+1})$) and the taxonomy-related factors (i.e., the path from the root to $v_{|L|}$). As a result, $w(L)$ can be used to combine the above two assumptions mathematically. Finally, for each edge $e(v_i, v_j)$ on the hierarchy, we estimate the edge weight $w(e(v_i, v_j))$ as the weight of the path on the hierarchy that ends with this edge and has the maximum path weight, resulting in the following equation:

$$w(e(v_i, v_j)) = \max_{\substack{L \text{ ends} \\ \text{with } e(v_i, v_j)}} w(L) \tag{7}$$

Generally, the likelihood of a topic hierarchy can be correlated with the user's information need through the estimation of $w(e(v_i, v_j))$, in which the more the topic hierarchy meets the information need, the higher likelihood it can obtain. This is one of the major differences between our method and previous approaches on general topic hierarchy construction [25][29].

4.4.2 Topic Hierarchy Construction Algorithm

Using the proposed likelihood function, we propose Algorithm 1 for topic hierarchy generation. Generally, the algorithm first sets the root of the resultant hierarchy at t_r. Next, it runs iteratively for η iterations (`Line 2`). In each iteration, the algorithm selects a specific topic and add it to a specific position on the current hierarchy so that the likelihood of the resultant hierarchy is maximized.

In practice, although we can find the optimal topic to insert by enumeration, to find the optimal position on the hierarchy for the insertion is not trivial. To this end, we denote v_{new} as the candidate new node and $\mathcal{V}_q^{i-1}$ as the

topic node set of the current hierarchy. Then for each topic node v_{old} in $\mathcal{V}_q^{i-1}$, we first link v_{new} with it using a new edge, i.e., $e(v_{new}, v_{old})$ or $e(v_{old}, v_{new})$ *iif* the corresponding subtopic relation exists(`Line 6 - 13`), resulting in $\hat{\mathcal{H}}_q^{i-1}$.

The resultant $\hat{\mathcal{H}}_q^{i-1}$ could be problematic. For example, it may contain conflicting nodes and undirected cycles, which could hinder the usage of the generated topic hierarchy. In this research, we propose to solve the problems by pruning $\hat{\mathcal{H}}_q^{i-1}$ using the following two functions.

The *RemoveDup*(.) function (`Line 14`) is used to solve the following two conflicts caused by the existence of duplicated nodes for a same topic on $\hat{\mathcal{H}}_q^{i-1}$.

- *Duplicated Children*: It happens when v_{old} has two subtopics v_{new} and v_{old2}, for which $t(v_{new}) = t(v_{old2})$. We can solve it by removing $e(v_{old}, v_{new})$ from $\hat{\mathcal{H}}_q^{i-1}$. Using a similar solution, we can solve the *Duplicated Parent* problem.

- *Duplicated Path Node*: It happens when there exists a path like $v_{old2} \rightarrow ... \rightarrow v_{old} \rightarrow v_{new}$ in which $t(v_{new}) = t(v_{old2})$. To solve this problem, we remove $e(v_{old}, v_{new})$ from $\hat{\mathcal{H}}_q^{i-1}$. Similarly, we can solve the problem on the converse path.

The *OptimumBranching*(.) function (`Line 15`) aims to break the undirected cycles, which happens when there exists an undirected path like $v_{new} - v_{old2} - ... - v_{old} - v_{new}$ on $\hat{\mathcal{H}}_q^{i-1}$. To solve this problem, we run the optimum branching algorithm [7] on $\hat{\mathcal{H}}_q^{i-1}$, which aims to obtain a tree from $\hat{\mathcal{H}}_q^{i-1}$ which has the highest sum of edge weights. Note that in Equation 5 the likelihood of a topic hierarchy is defined as the sum of the edge weights, the *OptimumBranching*(.) function can guarantee that its output topic hierarchy is the one with the highest topic hierarchy likelihood.

Finally, note that we introduce a parameter ζ to limit the maximum depth of the resultant topic hierarchy (`Line 17`). The intuition behind is simple, that users usually do not prefer too complex topic hierarchies. Generally, both ζ and the mentioned parameters η, i.e., the limit of the resultant topic hierarchy's size, can be tuned using human generated topic hierarchies like Amazon product categorization[5].

4.5 Topic Hierarchy based Customized Corpus Organization

In this section, we propose to organize the social media contents in $\mathcal{D}(t_r)$ for a given information need q by assigning them to their relevant nodes on the generated topic hierarchy $\mathcal{H}_q$. Specifically, we first collect the relevant documents for each topic in $\mathcal{T}_q$. Next, for each topic node on the hierarchy, we rank its relevant documents using a probability based model and select the top ranked ones to form its relevant content set.

4.5.1 Collection of Relevant Document for Topics

For each topic $t \in \mathcal{T}(\mathcal{D}(t_r))$, there could be many documents in the social media corpus that focus on it. Without loss of generality, we assume that we have a topic extraction function $Topic(d) \rightarrow 2^{\mathcal{T}(\mathcal{D}(t_r))}$ to reveal the focuses of documents, then a *relevant document* for the topic t can be defined as a document that contains t as one of its focused topics. Next, denoting $\mathcal{D}(t) \subset \mathcal{D}(t_r)$ as the *relevant document set* for topic t, it can be obtained as follows.

$$\mathcal{D}(t) = \{\bigcup_i d_i | d_i \in \mathcal{D}(t_r), t \in Topic(d_i)\} \quad (8)$$

In practice, a simple TF-IDF based keyword extraction method is employed as the $Topic$(.) function. It is noted that $\mathcal{D}(t)$ is only defined for topics, not topic nodes on topic hierarchies.

[5]http://www.amazon.com/

Algorithm 1 Topic Hierarchy Construction Algorithm

Input: $\mathcal{T}_q$, the focused topic set;
η, ζ, hierarchy size and depth constraints, respectively;
Output: $\mathcal{H}_q$, the resultant topic hierarchy;

```
1: Initiate : set the hierarchy root at t_r, resulting in H_q^0;
2: for i=1; i ≤ η ; i++ do
3:   for each t_s ∈ T_q do
4:     create a new node v_new, where t(v_new) = t_s;
5:     Ĥ_q^{i-1} = H_q^{i-1};
6:     for each v_old ∈ V_q^{i-1} do
7:       if μ(t(v_new), t(v_old)) ≠ 0 then
8:         Ĥ_q^{i-1} ← e(v_new, v_old);
9:       end if
10:      if μ(t(v_old), t(v_new)) ≠ 0 then
11:        Ĥ_q^{i-1} ← e(v_old, v_new);
12:      end if
13:    end for
14:    Ĥ_q^{i-1} = RemoveDup(Ĥ_q^{i-1});
15:    H^i_(q,t_s) = OptimumBranching(Ĥ_q^{i-1});
16:  end for
17:  H_q^i = arg max_{H^i_(q,t_s) where depth(H^i_(q,t_s))≤ζ} L(H^i_(q,t_s));
18: end for
19: H_q ← H_q^η.
```

4.5.2 Representative Document Selection

Obtained from social media sources like Twitter, the relevant document set could be very large and noisy for many topics. In this step, in order to present the users with more compact and precise information on each topic node v on the hierarchy, our goal is to identify the top N representative documents in $\mathcal{D}(t(v))$ to form the relevant content set of v, i.e., $c(v)$.

Generally, there are two factors that we should consider here. First, when determining a document's representativeness for a topic node, besides the corresponding topic term, we should also consider its position on the hierarchy. For example, the representative documents for the topic problem on the edge lens → problem could be very different from those on the edge battery → problem. On the other hand, the document's source may also affect its potential usefulness. Take tweets as the example, although they are usually noisy and less informative than blogs or cQAs for formal topics like the policy of "Barack Obama", they can also be useful when encountering timely topics such as release date and price of "iPhone 5s".

In this research, given a topic node v and a document $d \in \mathcal{D}(t(v))$, we propose to use a probability $p(d|v)$ to indicate the document d's representativeness for v. Recall that the document $d = (c, s)$, in which c indicates the content of d and s indicates the source that d belongs to. Assuming that c and s are independent, we can decompose $p(d|v)$ into

two parts as in Equation 9. By doing so, the content and source based features of d can be combined to estimate the document's representativeness.

$$p(d|v) = p(c, s|v) = p(c|v)p(s|v) \quad (9)$$

In Equation 9, $p(c|v)$ indicates the probability of the document's content c given the topic node v. As discussed before, it could be relevant to both the topic $t(v)$ and its immediate ancestors on the hierarchy. Denote $\mathcal{F}(v)$ as the set of v and its immediate ancestors, let $d(v_i, v_j)$ be the distance between v_i and v_j on the hierarchy, we employ the Okapi BM25 function [18] $okapi(.,.)$ to estimate $p(c|v)$ as follows.

$$p(c|v) = \frac{1}{\mathcal{Z}} \sum_{v_i \in \mathcal{F}(v)} \alpha^{d(v_i, v_j)} \cdot okapi(c, t(v_i)) \quad (10)$$

in which $\mathcal{Z}$ is a constant that guarantees $p(c|v) \in [0, 1]$ and $\alpha \in [0, 1]$ is a decay factor which controls the impact of v's distant ancestors on $p(c|v)$. Note that when $\alpha = 0$, no structure information on the hierarchy will be considered. More details on the impact of α will be shown in Section 5.5.2.

On the other hand, the second factor in Equation 9, i.e., $p(s|v)$ indicates the impact of the document's source to its representativeness. In this research, we further expand it using Bayesian rules, resulting in Equation 11.

$$p(s|v) = \frac{p(s)p(v|s)}{p(v)} \propto p(s)p(v|s) \quad (11)$$

In Equation 11, $p(s)$ is the a priori probability of source s, which can be interpreted as the overall content quality of the social media source s. The intuition behind this is that documents from high quality sources (e.g., blogs and cQAs) are usually more preferable by users. In practice, we estimate this probability as the average salience score of the topics extracted from this source, i.e., $p(s) \propto \frac{\sum_{t \in \mathcal{T}(\mathcal{D}^s)} freq(t) \cdot \delta(t)}{\sum_{t \in V(\mathcal{D}^s)} freq(t)}$, in which $\mathcal{T}(\mathcal{D}^s)$ indicates the topic set of $\mathcal{D}^s$ and $freq(t)$ is the frequency of the topic t in $\mathcal{D}^s$.

Next, the probability $p(v|s)$ in Equation 11 can be regarded as the supportiveness of the source s to topic $t(v)$. In this paper, we formulate it as the frequency of $t(v)$ in D^s, i.e., $p(v|s) \propto \frac{|\mathcal{D}^s(t(v))|}{|\mathcal{D}^s|}$ in which $\mathcal{D}^s(t(v))$ indicates the relevant document set of $t(v)$ in source s. The intuition behind this is that if a topic is frequently discussed in a source, the contents in this source could be useful to describe this topic.

Finally, for each topic node v on the topic hierarchy, its relevant content set $c(v)$ can be generated by aggregating the top ranked N documents by $p(d|v)$ from its relevant document set $\mathcal{D}(t(v))$.

5. EVALUATION

5.1 Social Media Corpus

As shown in Table 1, we crawled blogs, cQAs and tweets to build up the test social media corpus. In order to demonstrate the performance of the proposed method in open-domain applications, the data we crawled is focusing on 9 root topics, which belong to various domains such as digital products, politicians, corporations and tv series. For each root topic, we crawled the blogs by querying Google Blog search[6] and collecting the first 250 returned blogs. For cQAs and tweets, we used the Yahoo! Answer API and Twitter API to obtain real-time data from the data stream [7].

[6]http://www.google.com/blogsearch

Topic	blogs	cQA	tweet	#topic
iPhone 5	237	2,476	504,467	498
iPhone 5s	244	901	156,830	267
Barack Obama	234	1,781	544,213	774
Chanel	219	1,284	143,640	288
Game of Thrones	226	458	130,157	210
The Walking Dead	231	918	99,650	151
Facebook Inc.	229	2,152	430,693	447
Microsoft Corp.	244	1,034	409,778	409
NBA	238	962	109,016	233

Table 1: A brief statistics on our data set.

{iPhone 5, camera}, {iPhone 5, problem}
{iPhone 5s, features}, {iPhone 5s, release}
{Barack Obama, policy}, {Chanel, makeup}
{Game of Thrones, cast}, {Chanel, shop}
{The Walking Dead, comic}, {Facebook, games}
{The Walking Dead, reviews}, {NBA, players}
{Microsoft, office}, {Microsoft, device}

Table 2: List of the test information needs.

After pre-processing the crawled data, 395.6 subtopics are extracted for each root topic on average. Among these topics, it could be very time-consuming for users to manually find the relevant topics for their information needs. Moreover, the average amount of relevant documents per topic is 84.9 (could be thousands for popular topics like microsoft office), making it also impossible for users to go through all of them. Generally, the above observations demonstrate the necessity of our proposed task in this paper.

5.2 Test Information Needs

For the 9 root topics, we manually created 14 information needs for evaluation as listed in Table 2. They were selected by the following process: First, we submitted each root topic to search engines and collected the phrases in the "related search" panels as candidates. Next, for each root topic, we manually selected one or two phrases (e.g., "iPhone 5 problem") from the candidate set that have hierarchical subtopics to construct the test information needs.

5.3 Focused Topic Discovering

5.3.1 Experiment Setup

In this section, we first show the effectiveness of the proposed method in focused topic discovery. We compare our method ($ThiR$) with 2 baselines: (1) *Topic Salience* (Sal) [27], in which $\mathcal{T}_q$ was generated by ranking the topics in $\mathcal{T}_q^0$ by their salience scores, and (2) *Topic Relevance* (Rel)[11], in which Markov random walk was adopted for topic weight propagation while the topics' initial weights were equally set as 1. For all methods, we collected their top 50 topics as their results.

We compare all the methods against the gold standard topic sets, which were generated as follows: First, four undergraduate students were asked to label each topic produced by the compared methods as *relevant* or *irrelevant* to the information need. Next, only the topics that were labeled as relevant by all the annotators were added into the gold standard sets.

[7]Our data set is available at data.csaixyz.org/kdd-2014/packed.rar

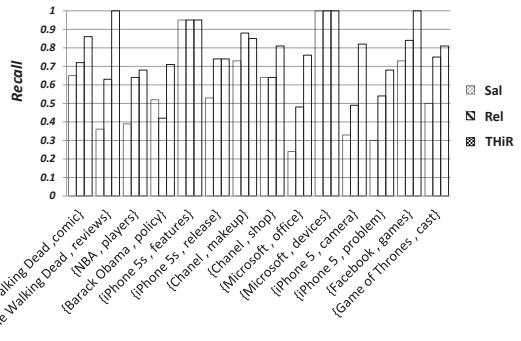

Figure 3: Recall comparison of different methods for focused topic discovering.

As to the parameters in the proposed method, we find that when k is around 10 for $\Omega_k(.)$, our resultant focused topic sets can cover most of the gold standards. This observation is reasonable since the amount of tightly relevant topics for any given topic is usually limited. Finally, we set $k = 10$ and set the iteration number to 5 as in [11]. The *recall* of all methods is reported in Figure 3, which can be calculated as $recall = \frac{|\mathcal{T}_q \cap \mathcal{T}_q^s|}{|\mathcal{T}_q^s|}$, in which $\mathcal{T}_q^s$ indicates the gold standard set.

5.3.2 Evaluation Results

From the result we can see that on average, the proposed method outperforms both baseline methods significantly (t-test, p-value < 0.05) by 48.3%, 20.0%, respectively. Compared to the *Sal* method, the latter two methods obtain better results in most cases, indicating the effectiveness of the topic weight propagation in focused topic discovery. On the other hand, the topics' salience scores are also useful for this task, resulting in 20.1% performance improvement of our method over the *Rel* method.

5.4 Focused Topic Hierarchy Construction

5.4.1 Experiment Setup

In this Section, we evaluate the performance of the proposed topic hierarchy construction method against manually created gold standards. For each information need, the gold standard hierarchy was constructed as follows: First, given the resultant focused topic set from Section 5.3, six undergraduate students were asked to connect the topics therein into a topic hierarchy independently. They then resolved the conflicts through discussions and came up with the final gold standard. Note that in order to prevent the gold standard hierarchies to become too complex to use, they are required to contain less than 40 nodes. On average, the gold standard hierarchies have 29.6 nodes, 30.6 edges and the depth is 4.2.

We use precision, recall and F1 score to measure the performance of the compared methods: Denoting E and E_g as the edge sets of an output hierarchy and the gold standard, the metrics are calculated as: precision (pre.) $= \frac{|E \cap E_g|}{|E|}$, recall (rec.) $= \frac{|E \cap E_g|}{|E_g|}$ and F_1 score $(F_1) = 2 \cdot \frac{\text{pre.} \cdot \text{rec.}}{\text{pre.} + \text{rec.}}$.

5.4.2 Verification of Underlying Assumptions

In this section, we first verify the usefulness of the two assumptions introduced in Section 4.4. To this end, we compare the proposed method ($THiR$) with its two variants: (1) *BooleanEdge*, in which we change Equation 6 into $w(L) = \frac{\sum_{k=0}^{|L|-1} w_{const}}{|L|}$, where $w_{const} = 1$, in which the impact of the users' information need is discarded, and (2) *MaxAllSub*, in which we change Equation 7 into $w(e(o_i, o_j)) = \mu(t(o_i), t(o_j))$ so that no taxonomy information is injected into the model. According to the average size and depth of the gold standard hierarchies, we set the parameter η and ζ as 30 and 5, respectively for all the compared methods.

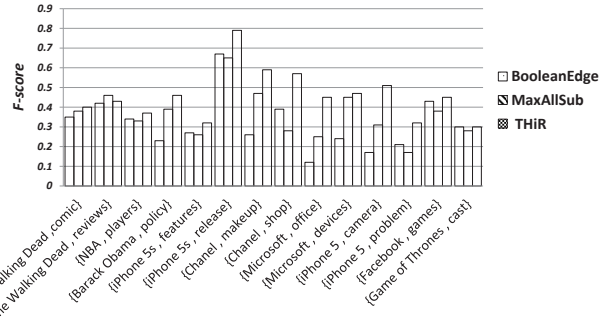

Figure 4: Comparison of F1-scores between the proposed methods and its two variations.

Figure 4 illustrates the comparison results on F_1 scores. We can see that the full model ($THiR$) outperforms its two variants significantly (t-test, p-value < 0.05), in which the average improvements against the *BooleanEdge* and *MaxAllSub* methods are 45.1% and 25.6%, respectively. From this we can conclude that, both of the two assumptions are important for the estimation of a topic hierarchy's likelihood, hence help to achieve better performance in topic hierarchy construction.

5.4.3 Comparison with State-of-the-Art Methods

In this section, we compare the proposed method with three state-of-the-art methods: (1) *Snow's Method* [21], which introduced a probability model to obtain the most probable topic hierarchy from a given topic set; (2) *Yu's Method* [26], which designed an information function to guild the topic hierarchy construction using semantic distance between topics and (3) *Zhu's Method* [29], in which a graph based iterative algorithm was adopted for topic hierarchy construction. For the implementation of all the baseline models, the subtopic relation strength was used in the probability, semantic distance and edge weight calculations, accordingly. Finally, although neither of the three baseline methods are developed for focused topic hierarchy construction, since we provide them the same focused topic sets as used in our method, they are still comparable to our method on the performance of topic hierarchy generation.

The experimental results are shown in Table 3, from which we can see that the proposed method outperforms all the three state-of-the-art methods by 41.6%, 64.8% and 60.7%, respectively. The reasons are two fold: First, compared to *Snow's* and *Yu's Methods*, the proposed method is more robust to the insertion errors due to our greedy algorithm framework, which also explains why we can obtain the highest precision on most test samples. On the other hand, although *Zhu's Method* also employed a greedy algorithm,

Information Need	*Snow's Method*			*Yu's Method*			*Zhu's Method*			*Our Method*		
	pre.	rec.	F_1	pre.	rec.	F_1	pre.	rec.	F_1	pre.	rec.	F_1
{The Walking Dead, comic}	0.27	0.24	0.25	0.19	0.17	0.18	0.24	0.21	0.22	0.48	0.35	**0.40**†
{The Walking Dead, reviews}	0.33	0.20	0.25	0.33	0.20	0.25	0.27	0.13	0.17	0.46	0.40	**0.43**†
{NBA, players}	0.33	0.10	0.15	0.25	0.07	0.11	0.42	0.12	0.19	0.58	0.27	**0.37**†
{Barack Obama, policy}	0.50	0.39	0.44	0.45	0.36	0.40	0.58	0.39	**0.46**	0.55	0.39	**0.46**
{iPhone 5s, features}	0.22	0.20	0.21	0.22	0.20	0.21	0.18	0.15	0.16	0.33	0.30	**0.32**†
{iPhone 5s, release}	0.75	0.60	0.67	0.69	0.55	0.61	0.88	0.70	0.78	0.83	0.75	**0.79**†
{Chanel, makeup}	0.37	0.43	0.40	0.33	0.39	0.36	0.33	0.39	0.36	0.62	0.57	**0.59**†
{Chanel, shop}	0.60	0.44	0.51	0.45	0.33	0.38	0.27	0.15	0.20	0.58	0.56	**0.57**†
{Microsoft, office}	0.19	0.17	0.18	0.14	0.13	0.13	0.10	0.10	0.10	0.41	0.50	**0.45**†
{Microsoft, devices}	0.42	0.25	0.31	0.50	0.30	0.37	0.50	0.25	0.33	0.43	0.50	**0.47**†
{iPhone 5, camera}	0.44	0.40	0.42	0.44	0.40	0.42	0.28	0.25	0.26	0.48	0.55	**0.51**†
{iPhone 5, problem}	0.32	0.21	0.25	0.24	0.16	0.19	0.25	0.16	0.19	0.40	0.26	**0.32**†
{Facebook, games}	0.27	0.20	0.23	0.18	0.13	0.15	0.50	0.27	0.35	0.44	0.47	**0.45**†
{Game of Thrones, cast}	0.67	0.17	0.27	0.33	0.10	0.14	0.56	0.14	0.23	0.40	0.23	**0.30**†

Table 3: Performance comparison between the proposed method with three state-of-the-art methods. The bold number indicates the highest F-score and † indicates the improvement against all baseline methods is significant in t-test, p-value < 0.05.

our method can make better decisions in each iteration because we optimize the topic selection and insertion uniformly, therefore we can find the global optimal solution for every iteration.

In Figure 5 we show a case study in which we compare the generated topic hierarchy for {NBA, players} of our method and that of the best baseline method, i.e., *Zhu's Method*. From the result, another advantage of our method is demonstrated: since the proposed method allows duplicated topic nodes for a topic, it can provide every NBA player a subtopic career on the hierarchy correctly. However, since all the baseline methods only allow career to emerge once on the hierarchy, their performance on recall are low.

5.5 Customized Corpus Organization

5.5.1 Experiment Setup

In this section, we evaluate the performance of the proposed method on representative document identification. The data set used in this experiment is generated as follows: First, for each focused topic hierarchy generated in Section 5.4, we selected 2 topic nodes, resulting in the 28 test topic nodes as shown in Table 4. Generally, the test nodes were carefully chosen so that they vary in many aspects, e.g., node depth, node frequency and number of subtopic nodes.

Next, for each test topic node, its relevant documents were collected using the method as described in Section 4.5.1. Without loss of generality, we set $N = 20$, thus only the top 20 documents from each method are collected as the relevant content set of the test topic nodes. To evaluate the performance of each method, we asked four undergraduate students to annotate each document in its results with a score as 5/2/0, indicating whether it is *representative*/*relevant*/*irrelevant* to the corresponding topic node. For conflicts in the annotation results for a document, we adopted the lowest one as its final score. Finally, for all compared methods in the following sections, we use *nDCG* to measure their performance, which has a higher value when the documents of higher scores are ranked higher.

lens, problem, battery life, camera, wifi, camera, price, china, afghanistan, economy, lipstick, stores, sarah connor, actor, france, shop, review, character, developer, player, story, actor, kobe bryant, players, install, excel,xbox, software

Table 4: List of the test topic nodes ordered by the positions of their corresponding information needs in Table 2.

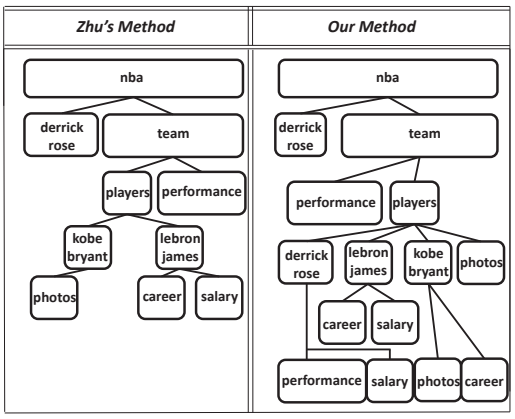

Figure 5: The comparison of the topic hierarchy generated for {NBA, players} by Zhu's method and our method.

5.5.2 The Impact of Topic Hierarchy

In this Section, we first evaluate the impact of the topic hierarchy's structure on representative document selection. In practice, we vary the parameter α in Equation 10 from 0 to 1 and observe the trend of the average *nDCG* on all the test topic nodes. Intuitively, a larger α would lead to larger impact of the structure information on the resultant relevant content sets.

The experimental results are presented in Figure 6. We can see that when $\alpha = 0$, where the structure information is not used, the *nDCG* is poor, indicating that the topic

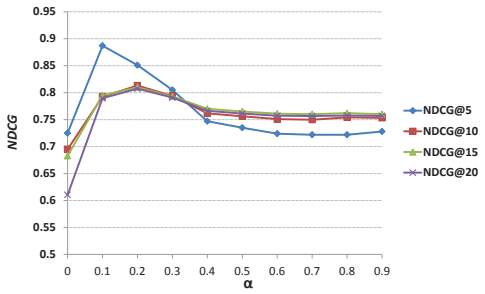

Figure 6: The $nDCG$ performance of the proposed method for different α.

hierarchy's structure is an essential factor for representative contents identification. Besides, we also observe that the $nDCG$ increases when α increases and the maximum $nDCG$ is achieved when $\alpha \in [0.1, 0.3]$, where we achieve a 31.1% improvement on $nDCG$@20 as compared to that when $\alpha = 0$. Based on this observation, we chose $\alpha = 0.2$ in our method.

5.5.3 Ablation Study on the Contribution of Features

In this section, to demonstrate the contributions of different features in the proposed ranking model, we compare our full model ($THiR$) with its four variations: (1) $BM25$, which ranks the documents by their BM25 scores given only the test topic node; (2) $THiR_{-src}$, which simplifies the full model by setting $p(s|o)$ in Equation 9 as 1, thus the source-based features are discarded; (3) $THiR_{-sup}$, which enhances $THiR_{-src}$ by setting $p(s|o) = p(s)$, i.e., adding the sources' a priori probabilities; and (4) $THiR_{-pro}$, which enhances $THiR_{-src}$ by setting $p(s|o) = p(o|s)$, i.e., adding in the supportiveness of different sources.

Method	*nDCG@N*			
	N=5	N=10	N=15	N=20
$BM25$	0.592	0.561	0.552	0.572
$THiR_{-src}$	0.759	0.669	0.636	0.615
$THiR_{-sup}$	0.787	0.701	0.663	0.648
$THiR_{-pro}$	0.809	0.807	0.808	0.804
$ThiR$	**0.813**†	**0.810**†	**0.810**†	**0.807**†

Table 5: The nDCG performance of different methods in representative document selection. † indicates the improvement is significant against the first three methods in t-test, p-value < 0.05.

The performance of all methods are shown in Table 5. From the results we can see that the full model can achieve satisfying performance ($nDCG@N > 0.8$) in identifying the representative contents for a given information need. Moreover, the full model also outperforms all the other methods, indicating that all the features introduced in this paper are useful to help finding the representative documents for topic nodes. Specifically, compared to $BM25$, $THiR_{-src}$ gains 7.5% improvement on $nDCG$@20 since it considers the structure of the whole hierarchy. The $THiR_{-sup}$ and $THiR_{-pro}$ methods further improve the performance by 5.4% and 30.7% respectively on $nDCG$@20 over the $THiR_{-src}$ method by the introduction of the social media source-based features.

5.5.4 Improve Document Retrieval Performance using Focused Topic Hierarchy

In this section we investigate the usefulness of the focused topic hierarchy on document retrieval task. By treating each information need, e.g., {iPhone 5, problem} as a keyword query, e.g., "iPhone 5 problem", the topic hierarchy can be used to improve its retrieval results on the social media corpus in two ways. First, if the hierarchy contains multiple nodes for the given subtopic, e.g., problem of camera and problem of battery, we can combine the representative documents of them together to generate a more comprehensive search results. Second, if the corresponding node bears many subtopic nodes, e.g., diagnosis of problem and solution of problem, we can also use the subtopics' representative documents to offer the user with more detailed information for the given query.

For the quantitative evaluation, we generate our test data set as follows: First, we aggregated together all the representative contents of topic nodes that meet the above two conditions for each test information need. Second, we ranked these contents again using the estimated probability and collected the top ranked 50 documents as the final search results. It is noted that the annotation process here is similar to that described in Section 5.5.1, except that the annotators were provided with the corresponding information need, i.e., query, instead of a topic node on the topic hierarchy to represent the user's search intent. In Table 6 we report the average $nDCG$ performance of our method ($THiR$) and a BM25 based document retrieval model ($BM25$), for which the information need is used as the input query.

Method	*nDCG@N*				
	N=10	N=20	N=30	N=40	N=50
$BM25$	0.713	0.699	0.701	0.711	0.715
$ThiR$	**0.873**	**0.870**	**0.875**	**0.870**	**0.871**

Table 6: The performance comparison of our method and a BM25 based method on query based document retrieval.

From the result we can see that the proposed method outperforms the BM25 based model by 23.1% on average $nDCG$. The reasons for the improvement are twofold. First, the relevant topics on the topic hierarchy can assist the retrieval model with more information, hence enabling it to understand the query better and return more relevant contents. Second, the diversity of the retrieved documents is also increased. Take the information need {iPhone 5 problem} as an example, compared to the search results of the BM25 model which only include the contents about iPhone 5's major problems like battery problem, the results of $THiR$ method can also cover various problems of iPhone 5, including the minor ones such as the wifi and ios problems.

6. CONCLUSION

In this paper, we proposed a novel method for customized social media organization using focused topic hierarchies, in which the social media contents can be organized into different structures to meet with different users' personal information needs. We developed rigorous methods to incorporate the user's information need into the focused topic discovery and topic hierarchy construction process. To further assist users to search on the hierarchy, we developed a probability

based model to obtain the representative contents for each topic node. The evaluation results demonstrated the effectiveness of our method for both information organization and retrieval tasks. In the future, we will try to enhance the present framework with data from knowledge bases and social networks. It is also interesting if we can apply it to more sophisticated tasks like question answering.

7. ACKNOWLEDGMENTS

This research is supported by the National Basic Research Program (973 Program) under grant No.2012CB316301 & 2013CB329403, the National Science Foundation of China project under grant No.61332007 and No.61272227, the Tsinghua University Initiative Scientific Research Program (with No.20121088071) and the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

8. REFERENCES

[1] K. Bade and A. Nürnberger. Creating a cluster hierarchy under constraints of a partially known hierarchy. In *SDM'08*, pages 13–24. SIAM.

[2] D. M. Blei, T. L. Griffiths, M. I. Jordan, and J. B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *NIPS'03*.

[3] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *SIGKDD'06*, pages 554–560. ACM.

[4] S.-L. Chuang and L.-F. Chien. A practical web-based approach to generating topic hierarchy for text segments. In *CIKM '04*, pages 127–136. ACM.

[5] M. Danilevsky, C. Wang, F. Tao, S. Nguyen, G. Chen, N. Desai, L. Wang, and J. Han. Amethyst: A system for mining and exploring topical hierarchies of heterogeneous data. In *KDD '13*, pages 1458–1461. ACM.

[6] I. Davidson and S. Ravi. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *PKDD'05*, pages 59–70. Springer.

[7] J. Edmonds. Optimum branchings. *Journal of Research of the National Bureau of Standards B*, 71:233–240, 1967.

[8] R. Fu, J. Guo, B. Qin, W. Che, H. Wang, and T. Liu. Learning semantic hierarchies via word embeddings. In *ACL'14*, volume 1.

[9] B. C. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *SDM'03*, volume 3, pages 59–70. SIAM.

[10] X. Han and J. Zhao. Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In *ACL'10*, pages 50–59. ACL.

[11] J. He, V. Hollink, and A. de Vries. Combining implicit and explicit topic representations for result diversification. In *SIGIR'12*, pages 851–860. ACM.

[12] A. C. König and E. Brill. Reducing the human overhead in text categorization. In *KDD '06*, pages 598–603. ACM.

[13] X. Liu, Y. Song, S. Liu, and H. Wang. Automatic taxonomy construction from keywords. In *SIGKDD'12*, pages 1433–1441. ACM.

[14] O. Medelyan, S. Manion, J. Broekstra, A. Divoli, A.-L. Huang, and I. Witten. Constructing a focused taxonomy from a document collection. In P. Cimiano, O. Corcho, V. Presutti, L. Hollink, and S. Rudolph, editors, *ESWC'13*, volume 7882, pages 367–381. Springer Berlin Heidelberg.

[15] D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with pachinko allocation. In *ICML'07*, pages 633–640. ACM.

[16] Z.-Y. Ming, K. Wang, and T.-S. Chua. Prototype hierarchy based clustering for the categorization and navigation of web collections. In *SIGIR '10*, pages 2–9. ACM.

[17] R. Navigli, P. Velardi, and S. Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *IJCAI'11*, pages 1872–1877. AAAI Press.

[18] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, M. Gatford, et al. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, pages 109–109, 1995.

[19] M. Sanderson and B. Croft. Deriving concept hierarchies from text. In *SIGIR '99*, pages 206–213. ACM.

[20] U. Scaiella, P. Ferragina, A. Marino, and M. Ciaramita. Topical clustering of search results. In *WSDM'12*, pages 223–232. ACM.

[21] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *ACL'06*, pages 801–808. ACL.

[22] C. Wang, M. Danilevsky, N. Desai, Y. Zhang, P. Nguyen, T. Taula, and J. Han. A phrase mining framework for recursive construction of a topical hierarchy. In *SIGKDD' 13*.

[23] J. Wang, C. Kang, Y. Chang, and J. Han. A hierarchical dirichlet model for taxonomy expansion for search engines. *Urbana*, 51:61801.

[24] X. Wang, S. Liu, Y. Song, and B. Guo. Mining evolutionary multi-branch trees from text streams. In *KDD '13*, pages 722–730. ACM.

[25] H. Yang and J. Callan. A metric-based framework for automatic taxonomy induction. In *ACL'09*, pages 271–279. ACL.

[26] J. Yu, Z.-J. Zha, M. Wang, K. Wang, and T.-S. Chua. Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In *EMNLP '11*, pages 140–150. Association for Computational Linguistics.

[27] H.-J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma, and J. Ma. Learning to cluster web search results. In *SIGIR'04*, pages 210–217. ACM.

[28] Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM'02*, pages 515–524. ACM.

[29] X. Zhu, Z.-Y. Ming, X. Zhu, and T.-S. Chua. Topic hierarchy construction for the organization of multi-source user generated contents. In *SIGIR '13*, pages 233–242. ACM.