Topic Hierarchy Construction for the Organization of Multi-Source User Generated Contents

Xingwei Zhu[1]*, Zhao-Yan Ming[2]†, 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, zxy-dcs@tsinghua.edu.cn,
chuats@nus.edu.sg

ABSTRACT

User generated contents (UGCs) carry a huge amount of high quality information. However, the information overload and diversity of UGC sources limit their potential uses. In this research, we propose a framework to organize information from multiple UGC sources by a topic hierarchy which is automatically generated and updated using the UGCs. We explore the unique characteristics of UGCs like blogs, cQAs, microblogs, *etc.*, and introduce a novel scheme to combine them. We also propose a graph-based method to enable incremental update of the generated topic hierarchy. Using the hierarchy, users can easily obtain a comprehensive, in-depth and up-to-date picture of their topics of interests. The experiment results demonstrate how information from multiple heterogeneous sources improves the resultant topic hierarchies. It also shows that the proposed method achieves better F_1 scores in hierarchy generation as compared to the state-of-the-art methods.

Categories and Subject Descriptors

H.0 [**Information Systems**]: General; H.3.2 [**Information Storage and Retrieval**]: Information Storage

Keywords

User Generated Contents, Information Organization, Topic Hierarchy

1. INTRODUCTION

With the rapid development of Web 2.0, user generated contents (UGCs) on the internet are becoming important sources for information navigation and knowledge acquisition. For example, when a user wants to know Barack Obama's performance in the 2012 presidential campaign, he may refer to blogs on Blog sites like *Blogger*[1] for authoritative criticisms, ask questions on community question answering (cQA) sites like *Yahoo! Answers*[2] for specific information and read tweets from *Twitter*[3] to follow up with his friends' opinions.

However, the volume of UGCs is huge and increasing every day. Even for a specific topic, it is usually impossible for users to go through all the contents and manually identify the newly emerging and important sub-topics. On the other hand, though in some UGC sources like *Wikipedia*[4], the data is well organized into structured format which can be easily accessed, they cannot catch up with the ever changing internet since they rely on human to compile and update.

The characteristics of different kinds of UGCs are diversified. Information in any single source is always limited and domain specific. For example, if a user requires a technical report for "IPhone 5", it is better to refer to blogs or cQAs instead of tweets in Twitter. But when he wants to know how his friends like "IPhone 5", Twitter turns out to be a better place. As a result, in order to have an overall picture of the topic, users have to refer to multiple UGC sources, which further increases their burdens to integrate these heterogeneous contents together.

To address the above problems, in this paper, we propose to organize information from multiple UGC sources using an automatically generated topic hierarchy. The task is not trivial due to the following three challenges:

- Instead of merely constructing a hierarchy [8] or organizing contents according to an existing hierarchy [12], our task requires the two problems to be solved at the same time, which requires a seamless integration of state-of-the-art techniques in both fields.
- Though there have been many works proposed to organize information from individual UGC sources like blogs [4], cQAs [12] or tweets [14], *etc.*, integrating information from multiple sources is a new and difficult problem since they are heterogeneous and may contain different kinds of noise and errors.
- In order to keep up with the ever changing internet, a novel framework is required to enable real time update on the hierarchies with newly obtained data. The real

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

[†] Corresponding author.

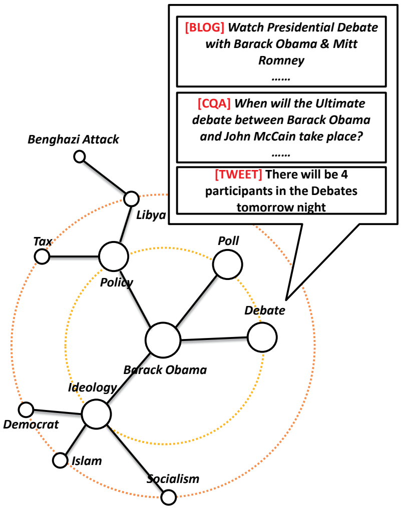
SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.
Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...$15.00.

[1] http://www.blogger.com/

[2] http://answers.yahoo.com/

[3] https://twitter.com/

[4] http://en.wikipedia.org/wiki/Main_Page



[BLOG] *Watch Presidential Debate with Barack Obama & Mitt Romney* ......

[CQA] *When will the Ultimate debate between Barack Obama and John McCain take place?* ......

[TWEET] There will be 4 participants in the Debates tomorrow night

Figure 1: The topic hierarchy for "Barack Obama". It organizes 119 blogs, 476 cQAs and 49749 tweets. Due to the space limit, we only illustrate the title of one blog, one cQA and one tweet for one of its sub-topics, *debate*.

time requirement is very different from previous approaches [8] [13], where a hierarchy is built using static data set and requires no further update.

In view of the above challenges, a three step framework is proposed to organize UGCs using automatically generated topic hierarchies. Given a collection of UGCs such as blogs, cQAs and tweets on a specific topic, we first identify potential topic terms from these sources. Then assisted by external knowledge from Wikipedia, *WordNet*[5] and search engine results, we propose a novel scheme to identify subtopic relations between topic terms using multiple evidences. Finally, by treating topic terms as nodes and sub-topic relations between them as edges, a graph-based algorithm is used to incrementally generate a topic hierarchy. Each time new data is available, the same framework can also be used to update the previously generated hierarchy with newly emerging sub-topics. Using the resultant topic hierarchy, the UGCs can be organized to nodes on the hierarchy according to their relevant sub-topics. In Figure 1, we show an example topic hierarchy for the topic "Barack Obama". It contains 11 sub-topics, under which a total of 119 blogs, 476 cQAs and 49749 tweets are organized. The contributions of our work can be summarized as:

- We propose a novel framework to organize UGCs by automatically generated topic hierarchies with real time update. The experimental results demonstrate its superior performance over the state-of-the-art methods.
- We leverage heterogeneous information from multiple sources by exploring their unique characteristics, and propose a novel scheme to combine the evidences extracted from these sources to enhance the framework.

The remainder of this paper is organized as follows: In section 2 we present the related work and in Section 3 we formally define the problem. Section 4 presents the framework of our approach, followed by detailed description on the three main modules, *i.e.*, topic term identification, topic relation identification and topic hierarchy generation including the real time hierarchy updating algorithm. In section 5, we discuss our evaluation method, experiments and results. Finally Section 6 concludes the paper.

2. RELATED WORK

Topic Term Detection: TF-IDF weighting has been found to be very effective [2] [6] for topic term extraction from documents. To avoid the need to estimation IDF, which requires huge document collections, Matsuo et al.[11] employed a sentence-level word co-occurrence matrix to find the keywords in a single document. NLP tools such as TextRunner [20] has also been applied to extract keyword terms, however, since these tools are linguistics-based, it is hard to use them to analyze UGCs which usually contain multiple languages and are ill grammar. In some recent works [10], the hierarchical cluster structure of documents (e.g., Wikipedia) is also adopted to improve the keyword selection performance.

Taxonomy Induction: Given a small document set, Lawrie et al. [9] proposed to extract its intrinsic topic hierarchy by estimating the topicality of terms and co-occurrence probability between them using only the given documents. Given a candidate term set, Navigli et al. [13] trained classifiers to detect is-a relations between terms and utilized a graph-based algorithm to optimize the term taxonomy. Besides, Snow et al.[16] introduced a probabilistic model to determine the most possible hierarchy for a set of concepts. Using both statistics-based and pattern-based features, Yang et al.[19] further proposed the metrics of information function and created hierarchies by an insert process, in which nodes are inserted onto the hierarchy to minimize the change of information functions. A recent work [21] extended this approach by employing more objective functions, *i.e.*, minimum Hierarchy Discrepancy and minimum Semantic Inconsistency to achieve a better insertion decision.

Approaches using Multiple Sources: Information from multiple sources provides researchers clues in different views and helps to achieve better results by overcoming the bias of any single information source [3] [5] [17]. Han et al.[3] used information from Wikipedia, WordNet and a NE cooccurrence corpus to measure the semantic relatedness between words. Although they just chose the conditionally most confident source to estimate the relatedness, instead of integrating the three sources together, the results have already outperformed the methods which only used single source. On the other hand, Hoffart et al.[5] proposed to integrate information from Wikipedia, WordNet, Geo-Name corpus, *etc.*, to build Yago2, an open domain structured knowledge base.

3. PROBLEM FORMULATION

We define the *root topic* $\mathcal{C}$ as a word or phrase which indicates the users' search intends. It can be an entity (*e.g.*, "Barack Obama"), an event (*e.g.*, "Benghazi Attack") or other informative concept. Given a root topic, we define its *information source set* S_c as $S_c = \{s_i\}_{i=1}^N$, in which s_i indicates a collection of documents from the i^{th} information source. They are collected for $\mathcal{C}$ and can be automatically updated when new data is available.

[5] http://wordnet.princeton.edu

The data in S_c is usually unstructured and contains noise and errors. We define a *topic hierarchy* as $H = \{T, M, R\}$, in which all useful information in S_c is organized using the following three components:

- **Topic Set** $T = \{t_1, t_2, ..., t_i, ...\}$, where t_i indicates a topic term. T includes the root topic $\mathcal{C}$ and the potential sub-topics of $\mathcal{C}$ for the documents in S_c.
- **Document-Topic Mapping** $M : d \to 2^T$, where d indicates a document in S_c. M assigns each document with terms in T as its relevant topics. For some noisy documents, the mapping results may be ϕ.
- **Sub-topic Relation Set** R: Denote $r(t_A, t_B)$ as a sub-topic relation, which means t_B is a sub-topic of t_A. $R = \{r_1, r_2, ..., r_i, ...\}$ is a set of sub-topic relations between the topics in T. It links all the topics into a hierarchy rooted at $\mathcal{C}$.

Formally, we define our task as follows:
Information Organization Task: Given a root topic $\mathcal{C}$ and its information source set S_c, we aim to build and continuously update a topic hierarchy H for $\mathcal{C}$ in order to organize the information in S_c according to their relevant topics.

4. APPROACH

4.1 Information Sources

Inevitably, every single UGC source has flaws. Take blog as an example, although well-written, blog usually focuses on narrow points (*e.g.*, technical reports for "IPhone 5" or big events for "Barack Obama") and takes a long time work before being published online. On the other hand, tweets can provide timely information (*e.g.*, release date for "IPhone 5" in Europe), while they also contain a huge amount of noises (*e.g.*, "*should I buy a IPhone 4S or IPhone 5?*"). These drawbacks limit the potential uses of the UGCs.

In this paper, we will tackle the above problem by combining the power of three prevailing UGCs, *i.e.*, Blogs, cQA and Twitter as our information sources. we also utilize domain-independent factoid knowledge from Wikipedia, WordNet, *etc.* to supplement the limited and specific UGC sources.

4.2 Framework

To address the information organization task, we introduce a three step framework as illustrated in Figure 2. For a given root topic $\mathcal{C}$, we first identify potential sub-topic terms from S_c. Then we identify the sub-topic relations between the sub-topics. Finally we generate the topic hierarchy and assign UGCs in S_c onto the hierarchy. The following three sections will describe the details for each module. We will also show how the framework can be used to update an existing hierarchy when new data is available.

4.3 Topic Term Identification

We collect potential sub-topics for $\mathcal{C}$ in two steps: (1) we extract keywords from documents in S_c to obtain an initial topic set; and (2) we extend the topic set with more abstract and general topics using knowledge from external sources.

4.3.1 Grounding Topic Extraction

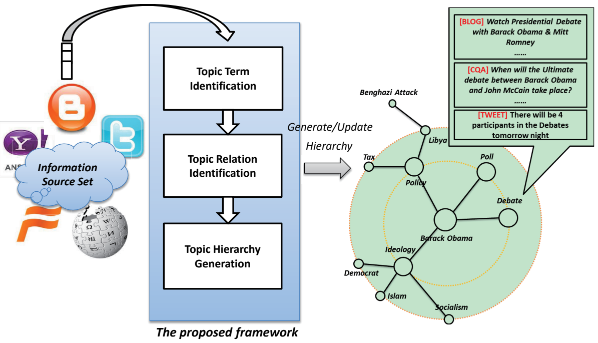

Information Source Set
ANS
Topic Term Identification
Topic Relation Identification
Topic Hierarchy Generation
The proposed framework
Generate/Update Hierarchy
[BLOG] Watch Presidential Debate with Barack Obama & Mitt Romney ......
[CQA] When will the Ultimate debate between Barack Obama and John McCain take place? ......
[TWEET] There will be 4 participants in the Debates tomorrow night
Benghazi Attack
Libya
Tax
Policy
Poll
Debate
Barack Obama
Ideology
Democrat
Islam
Socialism

Figure 2: The proposed method: maintaining a Topic Hierarchy using real-time data from multiple heterogeneous UGC sources by a three step framework.

Heuristic	Details
POS-tag	A topic term must be a NP[6]("battery") or a NP phrase("battery life").
Length	A topic term contains at least 1 word and at most 3 words.
Frequency	A topic term must occurs in at least 3 documents in the corpus.
Linguistic-Alteration	If a phrase is like "NP_1 of NP_2"("life of battery") or "NP_2's NP_1"("battery's life"), it will be converted to "NP_2 NP_1"("battery life"). Rules about *plural* and *abbreviation* are also used.

Table 1: Heuristic rules used in topic identification

The documents in S_c contains many potential sub-topic terms. Since these terms directly reflect people's focuses and interests, we denote them as *grounding topics*, t_g. To characterize a potential grounding topic, we first adopt four heuristic rules from [7], *i.e.*,*the pos-tag heuristic*, *length heuristic*, *frequency heuristic* and *linguistic-alteration heuristic*. In Table 1, we present a brief introduction of them.

Next, for each source in S_c, we separately estimate the *TF-IDF* scores for the potential grounding topics and obtain topic terms for each document as follows.
Blogs: We use the blog title and content to compute the TF-IDF scores for terms in blogs. Since the title of a blog usually indicates its main topics, we double the weights of terms in titles. For each blog, the top 5 ranked terms by TF-IDF are selected as its topic terms.
cQAs: We use the question title, description and the best answers to compute the TF-IDF scores. For each cQA, the top 5 ranked terms are selected as its topic terms.
Tweets: We use the content and the words surrounded by hash tags to compute the scores. Since tweets are short and cover fewer topics, we only collect the top ranked term as their topic terms.

Simply putting the topic terms of documents together may bring lots of noise. In order to select high quality and popular topic terms for the root topic, we adopt the following two assumptions:

Assumption 1: A topic term is of ***high quality*** *only if it can be extracted from documents in different information sources.*

Assumption 2: A topic term is ***popular*** *only if it can be extracted from many different documents.*

[6] We use the Stanford CoreNLP toolkit (http://nlp.stanford.edu/software/corenlp.shtml) to get the POS-tags.

We thus only collect the topic terms that are among the top 200 frequent topic terms in at least 2 different information sources into the *grounding topic set*, $T_G = \{t_{g1}, t_{g2}, ...\}$.

4.3.2 Topic Set Extension

Although T_G contains many low-level topic terms (*e.g.*, ***siri***, ***battery*** for "IPhone 5"), it lacks middle level topic terms (*e.g.*, ***software*** and ***device***) partly because they are too abstract and general for UGCs. Since these middle level terms are also useful in topic hierarchy generation, we investigate external information sources to obtain them.

Search Engine: For each term in T_G, we use two patterns, *i.e.*, "* such as <slot>" and "<slot> of *" to obtain its higher level topic terms. We first fill the term into the slot and submit it to a search engine like *Bing*[7]. Then we collect the noun phrases in the position of the wildcard on the returned search pages as middle level topics.

WordNet: WordNet contains hypernym relations between concepts. If a topic term in T_G is included in WordNet, we collect the terms in its direct hypernym synset as middle level topics.

Wikipedia: If a topic term in T_G is a Wikipedia title, its category tags are also collected as middle level topics.

The middle level topics shared by at least 2 sources are collected into the *extended topic set*, T_E. Then the final candidate topic set $T = \{\mathcal{C}\} \cup T_G \cup T_E$.

4.4 Topic Relation Identification

The sub-topic relation between topic terms provides essential information to organize the topics. Take "Barack Obama" as an example, if we know that there is a sub-topic relation between two of its sub-topics, *e.g.*, ***policy*** → ***tax***, which indicates that ***tax*** is a sub-topic of ***policy***, it will be very probable that there is a path between the two topic terms on the topic hierarchy.

To infer a sub-topic relation $r(t_A, t_B)$, we need to estimate the probability that the topic t_B is a sub-topic of t_A. Inspired by [18], we approximate this probability with an empirical score $e(r(t_A, t_B))$ which is estimated by evidences from multiple sources. The evidences are summarized in Table 2 and the details are listed as follows:

Evidences from the Information Source Set: If there is a sub-topic relation between two topic terms, they must be highly related, where contextual distribution can be used to measure this relatedness. More specifically, for each topic term, the nouns, verbs and adjectives that co-occur with it in the documents/sentences in S_c are collected as its document/sentence level contexts. For each term pair t_A and t_B, the document and sentence level contextual evidences, *i.e.*, $e_{distr_{doc}}(t_A, t_B)$[8] and $e_{distr_{sen}}(t_A, t_B)$ are defined as the cosine similarity between the corresponding contexts of them.

Evidences from Wikipedia: *Pointwise Mutual Information (PMI)* is also a measurement for the relatedness between two terms. We compute the PMI over a subset of Wikipedia corpus, which only includes pages that are not redirect pages and contain more than 1000 words. For t_A and t_B, we compute the normalized PMI as $npmi(t_A, t_B) = \frac{pmi(t_A,t_B)}{-log[p(t_A,t_B)]}$, where $pmi(t_A, t_B) = log\frac{p(t_A,t_B)}{p(t_A)p(t_B)}$. Then the pmi evidence $e_{pmi}(t_A, t_B) = \frac{1+npim(t_A,t_B)}{2}$.

We also use the category and sub-title evidences from Wikipedia, *i.e.*, $e_{wcate(t_A,t_B)}$ and $e_{wtitle}(t_A, t_B)$, which can be estimated as follows, respectively:

$$e_{wcate}(t_A, t_B) = \begin{cases} 1 & : \text{ wikipage } t_B \text{ has category tag } t_A \\ 0 & : \text{ otherwise.} \end{cases} \tag{1}$$

$$e_{wtitle}(t_A, t_B) = \begin{cases} 1 & : \ t_B \text{ is a subdirectory of } t_A \text{ on a wikipage} \\ 0 & : \text{ otherwise.} \end{cases} \tag{2}$$

Evidences from WordNet: If t_A and t_B can be found in WordNet and t_A is an ancestor of t_B, then it is very probable that t_B is a sub-topic of t_A. In practice, if t_A is an ancestor of t_B in WordNet, we use the WordNet similarity [15] to estimate the WordNet evidence as $e_{wnet}(t_A, t_B) = \frac{1}{WNDis(t_A,t_B)}$, where $WNDis(t_A, t_B)$ denotes the length of the shortest path between t_A and t_B in WordNet. Otherwise we just let $e_{wnet}(t_A, t_B) = 0$.

Evidences from Search Engine Results: We also use patterns like "*<topic>* such as *<subtopic>* and" to collect evidences from the internet. For each term pair t_A and t_B, we generate a query by filling them into the pattern (*e.g.*, "t_A *such as* t_B and") and submit it together with the root topic [9] to the search engine. Denote $pattern_i$ as the i_{th} pattern, we can obtain a 0/1 pattern-based evidence $e_{spattern_i}(t_A, t_B)$, whose value is set to 1 only if the search engine returns more than ζ results that contain this query; otherwise it is set to 0. In practice, we select 6 patterns as listed in Appendix A and ζ is set to 10 empirically.

To combine the above evidences, instead of simply adding or multiplying them together like [19][21], we first divide the evidences into two sets as shown in Table 2, *i.e.*, the **directed-evidence** set E_{dir} which includes the evidences that can determine the direction of the sub-topic relation and the **undirected-evidence** set E_{und} in which the evidences bear no directionality information.

For the directed-evidences, we use a linear combination to combine them together as follows:

$$e_{dir}(t_A, t_B) = \sum_{e_k \in E_{dir}} w_k \cdot e_k(t_A, t_B) \tag{3}$$

where w_k is the weight of the k^{th} evidence in determining the direction of the sub-topic relation and $\sum_k w_k = 1$.

Since the training data is not always available for open domain topics, supervised methods used in previous works [21] on weight estimation cannot be used to calculate w_k. In this paper, we propose an unsupervised method to meet this challenge. The basic idea of our method is that since sub-topic relation is directed, it is not possible that both $r(t_A, t_B)$ and $r(t_B, t_A)$ exist. Therefore, if a directed-evidence e_k supports both $r(t_A, t_B)$ and $r(t_B, t_A)$, which happens when the difference between $e_k(t_A, t_B)$ and $e_k(t_B, t_A)$ is not significant, e_k should be less weighted. More specifically, we estimate the weight w_k for each directed-evidence e_k as $w_k = \frac{dif_k}{\sum_k dif_k}$, in which dif_k is calculated as follows:

$$dif_k = \frac{1}{p_k} \sum_{\substack{t_A, t_B \in T \\ t_A \neq t_B}} |e_k(t_A, t_B) - e_k(t_B, t_A)| \cdot l_{k,t_A,t_B}$$

[7] http://www.bing.com

[8] We do not use tweets in generating document level contextual evidence since they are relatively too short.

[9] The root topic is used to filter out irrelevant search results.

directed-evidences in E_{dir}	**Source**
$e_{spattern_0}$ - $e_{spattern_5}$	*search engine*
e_{wtitle}	*wikipedia*
e_{wcate}	*wikipedia*
e_{wnet}	*wordnet*
undirected-evidences in E_{und}	**Source**
$e_{distr_{doc}}$	*information set*
$e_{distr_{sen}}$	*information set*
e_{pmi}	*wikipedia*

Table 2: The sources of evidences we use to estimate the probability of a sub-topic relation

where $l_{k,t_A,t_B} = \max\{e_k(t_A,t_B), e_k(t_B,t_A)\}$ gives higher weights to the evidences with high values and $p_k = \sum_{t_A,t_B \in T, t_A \neq t_B} |e_k(t_A,t_B) + e_k(t_B,t_A)|$ punishes the evidences that are too general. Then the final score $e((r(t_A,t_B))$ is estimated using the following equation:

$$e(r(t_A,t_B)) = e_{dir}(t_A,t_B) \cdot \prod_{e_s \in E_{und}} e_s(t_A,t_B) \qquad (4)$$

where $\prod_{e_s \in E_{und}} e_s(t_A,t_B)$ combines all the undirected-evidences for the two topic terms.

4.5 Topic Hierarchy Generation

By treating the sub-topics in T as nodes, sub-topic relations between them as edges, we can generate a sub-topic graph. Next, we estimate the weight for each edge using the scores estimated in section 4.4 using the following equation:

$$w(r(t_A,t_B)) = \begin{cases} |e(r(t_A,t_B)) - e(r(t_B,t_A))| \cdot e(r(t_A,t_B)), \\ \qquad \text{if } e(r(t_A,t_B)) > e(r(t_B,t_A)) \\ \qquad \text{and } t_A \neq t_B \\ 0, \qquad \text{if } e(r(t_A,t_B)) < e(r(t_B,t_A)) \\ \qquad \text{and } t_A \neq t_B \\ 1, \qquad \text{if } t_A = t_B \end{cases} \qquad (5)$$

in which $w(r(t_A,t_B))$ is proportional to both $e(r(t_A,t_B))$ and the difference between $e(r(t_A,t_B))$ and $e(r(t_B,t_A))$, indicating the strength of the edge from the node t_A to t_B. By removing the zero weighted edges, equation 5 also guarantees that there is only one directed edge between two nodes, which is essential for a valid hierarchy.

Next, we need to prune this graph into a hierarchy. Inspired by [13], we employ a graph based method to tackle this problem. Different from this work, we take an iterative approach: at each step we only add one sub-topic into the hierarchy, which makes our method amendable to incrementally update the hierarchies. In general, the proposed hierarchy generation algorithm can be formalized as Algorithm 1.

Given a candidate topic set T for $\mathcal{C}$, the algorithm functions as follows. Let T_i and R_i be the resultant topic set and sub-topic relation set of the hierarchy after the i^{th} iteration, we initialize them as $T_0 = \{\mathcal{C}\}$ and $R_0 = \phi$ (line 1). In the i^{th} iteration, we first add a topic term t in $T - T_{i-1}$ into T_{i-1} (line 3 - 9) using the following function:

$$t = \underset{t_s \in T - T_{i-1}}{\operatorname{argmax}} \sum_{t_k \in T_{i-1}} (w(r(t_k,t_s)) + w(r(t_s,t_k))) \qquad (6)$$

where t is the topic term that maximizes $score(t) = \sum_{t_k \in T_{i-1}} (w(r(t_k,t_s)) + w(r(t_s,t_k)))$, which indicates the overall relatedness between t_s and topic terms in T_{i-1}. Note that *NIL* will be returned if $T - T_{i-1} = \phi$ or $score(t)$ is 0.

Algorithm 1 Hierarchy Generation Algorithm

```
Input:
  $T$: the candidate topic set;
Output:
  $R_{ret}$: the sub-topic relation set of the resultant hierarchy;
  $T_{ret}$: the topic set of the resultant hierarchy;
  Initialize $T_0 = \{\mathcal{C}\}$, $R_0 = \phi$
  for i = 1 TO $\infty$ do
    $t \leftarrow$ selectTermFrom($T - T_{i-1}$)
    if $t$ is NIL then
      $R_{ret} \leftarrow R_{i-1}$
      $T_{ret} \leftarrow T_{i-1}$
      break
    end if
    $T_i \leftarrow t \cup T_{i-1}$
    $R_i \leftarrow R_{i-1}$
    for all $t_k$ IN $T_{i-1}$ do
      if edgeBetween($t_k$, $t$) exists then
        $R_i \leftarrow R_i \cup$ edgeBetween($t_k$, $t$)
      end if
    end for
    edgeWeighting($R_i$)
    $R_i$ = hierarhcyPruning($R_i$)
  end for
```

Once we have added t into T_{i-1} to form T_i, we also add all the edges between t and topics in T_{i-1} into R_{i-1}, resulting in R_i (line 10 - 15). In order to guarantee that edges in R_i make up a valid hierarchy, next we use a graph based method to prune the edges in R_i as follows.

Edge Weighting using Topic Relatedness (line 16): Edge weighting assigns score to each edge in R_i, indicating its importance in constructing the hierarchy. Our method generalizes that of [13] by weighting each edge in R_i with the estimated weights instead of simple 0/1 values. The process is as follows:

- If a topic term is related to many grounding topics in T_G, it could be important in the hierarchy. Let $w_t(t_k)$ denotes the weight of a topic term t_k in T_i, we estimate it as $w_t(t_k) = \sum_{t_g \in T_G} w(r(t_{root},t_g)) \cdot w(r(t_k,t_g))$, in which $w(r(t_{root},t_g))$ gives the importance of the grounding topics, and $w(r(t_k,t_g))$ reflexes the relatedness between t_k and the grounding topics.
- Then for each node $t_k \in T_i$, by denoting $L = \{t_u \rightarrow t_{u+1}\}_{u=0}^{|L|}$ as a path that connects t_{root} and t_k, its score is calculated as: $score_L = \sum_{u=0}^{|L|-1} w_t(t_u) \cdot w(r(t_u,t_{u+1}))$. Next, for each edge $t_s \rightarrow t_k$, its weight is estimated as follows:

$$w_r(t_s \rightarrow t_k) = \max_{\substack{L \text{ ends} \\ \text{with } t_s \rightarrow t_k}} score_L \qquad (7)$$

Hierarchy Pruning using Optimum Branching (line 17): Given the edge weighting results, we can use the *Chu-Liu/Edmond's optimum branching algorithm* [1] to find a subset of the current edge set R_i, which is the optimized hierarchy for the given topic terms where *every non-root node has only one parent and the sum of the edge weights are maximized.*

When the iteration terminates at the N^{th} iternation, the resultant topic set T_{N-1} and sub-topic relation set R_{N-1} will be collected for the final topic hierarchy. To guarantee that the hierarchy is specifically related to the given topic $\mathcal{C}$ and the documents in S_c, we further remove (1) the nodes that are not reachable for the root topic and (2) the leaf nodes that are not in the grounding topic set. Finally, for each topic term t in the topic set, we assign the documents in S_c whose topic terms contain t to the corresponding node, resulting in M, the document-topic mapping function of the resultant topic hierarchy.

4.6 Topic Hierarchy Update

For a given topic, the information on the internet is ever-changing. Different from many previous approaches that work on static corpus, we find it useful and necessary to dynamically sketch evolving topic hierarchies for users. For example, assume we have built a topic hierarchy for "Barack Obama" before the presidential campaign; when people start to talk about his inauguration ceremony on the internet, we need to detect the corresponding new sub-topic ***inauguration*** and insert it and its relevant documents to the correct place on "Barack Obama"s topic hierarchy.

To this end, we make it a key function for our proposed framework to be able to incrementally update the topic hierarchy by using the newly obtained data. Let $H_{old} = \{T_{old}, M_{old}, R_{old}\}$ be the existing hierarchy and S_{new} the newly obtained data set, a new hierarchy $H_{new} = \{T_{new}, M_{new}, R_{new}\}$ can be obtained by updating H_{old} using the following process:

```
Update the candidate topic set:
    $T_{add} \leftarrow$ topic term identified from $S_{new}$;
    $T_{new} \leftarrow T_{old} \cup T_{add}$;
Update the topic hierarchy:
    $R_{new}$, $T_{new} \leftarrow$ generate hierarchy using Algorithm 1 in
    which $R_0 = R_{old}$, $T_0 = T_{old}$ and $T = T_{new} - T_{old}$, thus
    adding new nodes in $T_{add}$ and edges between terms in
    $T_{add}$ and $T_{old}$ into $T_{old}$ and $R_{old}$.
Update the document-topic mapping function:
    $M_{add} \leftarrow$ new mapping results between terms in $T_{new}$ and
    documents in $S_{new}$;
    $M_{new} \leftarrow M_{old} \cup M_{add}$;
```

This update process is robust. This is evidenced from the fact that, when there are mistakes in the existing hierarchy, the newly obtained information can be used to correct the wrong relations. For example, the sub-topic relation set $R_{old} = \{$***barack obama*** $\rightarrow$ ***tax***$\}$ contains a very ambiguous relation[10]. When a new topic term ***policy*** is discovered from the new data set, instead of just adding it as a child to any of the two existing nodes, our method can further break the original ambiguous relation and create a better structure for the three topic terms; thus $R_{new} = \{$***barack obama*** $\rightarrow$ ***policy***, ***policy*** $\rightarrow$ ***tax***$\}$.

5. EVALUATION

5.1 Experimental Setup

We collect UGCs from the internet to form a corpus of four categories: Digital Products, Politicians, Cosmetics and Corporations. In each category, two to three topics are selected, resulting in 11 root topics. For each topic, blogs, cQAs and tweets are collected. We crawl blogs for a topic by submitting the topic name as the query into the Google Blog search engine[11] and collect the first 200 returned blogs. For cQAs and tweets, we use the Yahoo! Answer API and Twitter API to obtain the data stream on the target topic. A brief statistics of our corpus can be found in Table 3.

Category	Topic	blog	cQA	tweet
Digital Products	IPhone 5	182	1,523	411,826
	IPad mini	192	797	2,782
	Xbox kinect	195	1,476	30,261
Politicians	Barack Obama	193	1,043	426,811
	Mitt Romney	188	1,024	2,759
	Hillary Clinton	190	1,050	811
Cosmetics	Chanel	179	985	2,782
	Estee Lauder	194	1,049	17,816
Corporations	Facebook Inc.	190	973	429,979
	Microsoft Corp.	175	1,034	429,483
	Blizzard Inc.	192	999	2,022

Table 3: Statistics on the collected data sets in each information source

Topic	Sub-topics
IPhone 5	game, siri, battery, cost, software, jailbreak, wifi, update, ...
Barack Obama	policy, debate, law, islam, benghazi, poll, democrat, tax, ...
Chanel	perfume, watch, bags, sunglass, toiletry, mall, event, advert, ...
Microsoft Corp.	bing, surface, os, partner, outlook, windows phone 8, office 2013, ...

Table 4: Resultant candidate topic sets for four exemplar root topics

Note that the data in all the three information sources are tagged with time stamps. For blogs, they can be obtained from the snippets in Google Blog search results which indicate when the blogs are published. The data from Yahoo! Answer and Twitter APIs also contains time stamps on when the questions are asked or the tweets are created.

5.2 Topic Term Identification

In this section, we will first analysis the candidate topic sets generated by the method described in section 4.3. Next we will demonstrate how the use of different UGC sources improve its performance.

Table 4 shows a portion of identified topic terms for four exemplar root topics. From the results we can see that UGCs contain very rich and diversified information. For "Barack Obama", sub-topics on different aspects like politics (e.g., ***law***), personal information (e.g., ***islam***) and latest events (e.g., ***debates***) can be precisely extracted. We even obtain more comprehensive product-related topics (e.g., ***sunglass***) for "Chanel" than those in Wikipedia.

Define the ***coverage*** of the candidate topic set T as the ratio of the documents in the information source set that contain at least one term in T as its relevant topics, the average coverage is 18.4% for all root topics. Although twitter contains a huge amount of noise (more than 85% are noise from statistics on a randomly selected data set that contains 550 tweets), the proposed method can effectively filter out most of them, hence only 18.2% high-quality tweets are covered. On the other hand, higher coverage is observed for blogs (54.7%) and cQAs (40.4%).

From the covered documents, we sample 20 blogs, 20 cQAs and 50 tweets for each topic. Three annotators are asked to

[10] It may means ***barack obama***'s ***policy*** on ***tax*** or ***barack obama***'s own ***tax***.

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

determine whether the topic terms we found are relevant to the corresponding documents. Only the documents that have agreement among all annotators are used for the statistics. The result shows that for all UGC sources and topic categories, our method can achieve a precision of 73.2% to 90.4%.

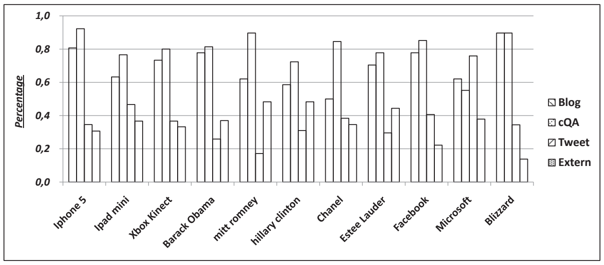


Figure 3: The percentage of topic terms supported by each information source on the topic set

Figure 3 shows the contribution of different UGC sources in topic term identification. We can see that all sources are indispensable. Although cQAs and blogs provide the majority of the topic terms, tweet tends to contribute timely and popular topics such as ***release date*** of ***iphone 5*** and ***price*** of ***ipad mini***. Though very limited, external sources provide middle-level topics such as ***os*** of ***microsoft*** and ***policy*** of ***barack obama***, which are essential when organizing the topics into a hierarchy.

5.3 Topic Hierarchy Generation

5.3.1 Evaluation Metrics

We evaluate our results against manually created gold standards. For each topic, three annotators are employed to create hierarchies independently using terms from the candidate topic set according to the following three rules.

Rule 1: Relevancy. All nodes on the hierarchy must be reasonable sub-topics of the root topic. It guarantees that users won't get noisy information from the topic hierarchy.

Rule 2: Maximum coverage. All relevant nodes should be included into the hierarchy. Hence users won't miss any useful information about the root topic.

Rule 3: Hierarchical structure. Each two connected nodes must be directly related that no other nodes in the candidate topic set can be inserted between them. This makes the hierarchy completely structured for users to quickly find their interested contents.

For example, given the candidate topic set {***barack obama***, ***policy***, ***tax***, ***medic care***, ***friday***} for "Barack Obama". First, only ***friday*** will be removed according to rule 1 and 2. Then according to rule 3, the gold standard hierarchy should be {***barack obama*** → ***policy***, ***policy*** → ***tax***, ***policy*** → ***medic care***}, which provides users a completely hierarchical view of the available UGCs about "Barack Obama".

Next, the three annotators compare their resultant candidate hierarchies and come up with the gold standards through discussions. On average, the gold standard hierarchies contain 22.3 nodes and 21.3 edges with an average depth of 3.64.

We use the precision, recall and F_1 scores to measure the hierarchy generation performance. Denote R and R_{gold} as the sub-topic relation sets of our output and the gold standard, respectively, the metrics can be calculated as follows:

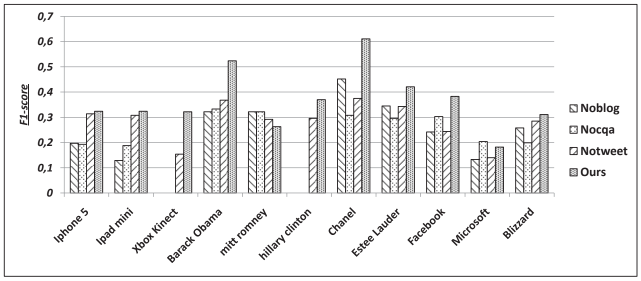


Figure 4: Performance on topic hierarchy generation when using different information source set

$$\text{precision (pre.)} = \frac{|R \cap R_{gold}|}{|R|)}$$

$$\text{recall (rec.)} = \frac{|R \cap R_{gold}|}{|R_{gold}|}$$

$$F_1 \text{ score } (F_1) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

5.3.2 Ablation Study on the Contribution of Information Sources

In Figure 4, we first compare the performance of our proposed methods when using different information source set, which consists of (1) only cQAs and tweets (*noBlog*), (2) blogs and tweets (*noCQA*), (3) blogs and cQAs (*noTweet*), and (4) all the three sources (*All*). The results show that the more the sources we use, the better the performance we can achieve, where the improvements of around 31.2% - 74.8% are observed on the average F_1 scores of *All* against those of the other three combinations. There are mainly two reasons. First, since multiple sources provide more topic terms (as shown in Figure 3), the recall is improved by 39% - 124%. Second, though not for all the topics, the average precision is also improved. This is partly because more UGCs help to estimate better evidences which result in building more optimal topic hierarchies.

On the other hand, when comparing the results of *noBlog*, *noCQA* and *noTweet*, we can see that *noBlog* performs the poorest for most topics (for topics like "Xbox Kinect", the F_1 score is even 0.). It indicates that blogs play a critical role in our method, partly because blogs are usually well-written and contain rich contents. For topics like "Chanel", the performance does not drop that much when removing the blog data. This is partly because many of the crawled blogs are about fashion events and product comparison instead of focusing on real sub-topics like ***perfume*** and ***lipstick***. Once we can collect more appropriate blogs for these topics, even higher performances can be expected.

5.3.3 Evaluation on Effectiveness of Evidence Combination Schemes

The proposed evidence combination scheme plays an important role in our method. In this section, we compare it with three baseline methods: (1) *Linear*, which simply combines all evidences linearly and has been used in [19] [21]. For each topic, we adopt ridge regression [19] to optimize the weight vector using the training data from the other topics in the topic category. (2) *No undirect*, a variation of the proposed scheme which does not use undirected-evidences. (3) *Equal direct*, a variation of the proposed scheme in which

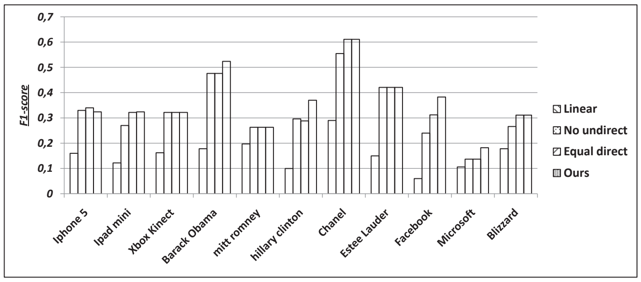


Figure 5: Performance on topic hierarchy generation when using different evidence combination schemes

the weights of all the directed-evidences are set to be equal. All methods run on the same topic sets as those used in generating the gold standards [12].

The result is shown in Figure 5. Compared to the *Linear* method, all the three methods that adopt our proposed scheme perform significantly better on the F_1 scores (t-test, p-value<0.05). This indicates that considering the directed and undirected evidences separately is a better way to estimate the probability of sub-topic relations than putting all the evidences together undistinguishedly. The use of undirected-evidences also significantly improve the performance (t-test, p-value<0.05). Since we only adopt nine carefully selected directed-evidences, the effect of weights is limited. However, as shown in Table 5, the weight vector indeed reflexes the quality of different evidences. For example, the Wikipedia-based evidences (e.g., e_{wcate}, e_{wtitle}) usually obtain higher weights since they are from high quality sources. Stricter patterns (e.g., $e_{spattern0}$) are also higher weighted than general patterns (e.g., $e_{spattern2}$). As the result, more improvement could be observed when more evidences of varied qualities [21] are introduced.

evidence	$e_{spattern_0}$[a]	$e_{spattern_2}$[b]	e_{wcate}	e_{wtitle}
weight	0.116	0.099	0.129	0.129

[a] <*topic*> such as <*subtopic*> and

[b] <*subtopic*> on <*topic*> and

Table 5: Part of the weight vector of directed-evidences for "IPhone 5"

5.3.4 Comparison with State-of-the-art Methods

Based on the results in previous two sections, our method performs the best when using all information sources and adopting the proposed combination scheme. We now compare our best method with three state-of-the-art methods: (1) *Yang's Method* [19], which organizes concepts into a hierarchy according to a information function. For each topic, the information function employs all the evidences in Table 2 and is trained using the gold standard of this topic. (2) *Navigli's Method* [13], a graph based method which only employs a classifier-based 0/1 evidence and does not support real time update. Since the evidence used in this paper only provides clues for *is-a* relation, we extend it with extra directed-evidences from our evidence set for the sake of fair comparison. (3) *Snow's Method* [16], which uses a probability model to obtain the most probable hierarchy for the concepts in a given concept set. For fair comparison, we use the probability of a sub-topic relation approximated for our method here.

In Table 6, we can see that our proposed method achieves the best performance on all metrics for most topics and significantly outperforms the state-of-the-art methods on the averaged F_1 score (t-test, p-value<0.05). Compared to Yang's method, the proposed graph based method makes better use of the relations among three or more topics. Taking the root topic "Chanel" as an example, given its sub-topic set {***chanel***, ***product***, ***perfume***, ...}, while Yang's method returns the relation set {***chanel*** → ***product***, ***chanel*** → ***perfume***, ...} by considering only the strongness of pairwise sub-topic relations, our method can further detect the sub-topic chain along the three topics, *i.e.*, ***chanel*** → ***product*** → ***perfume***, ,which better captures the global relation. Moreover, the proposed method outperforms Snow's method in term of F_1 score by about 12%. It is partly because the results of Snow's method are strongly affected by the insertion order of the topics. Once an insertion error occurs in one step, it cannot be corrected in the following steps. But our incremental updating mechanism can naturally solve this problem. Navigli's method tends to generate very deep hierarchies [13], where errors occur sometimes (e.g., ***iphone 5*** → ***camera*** → ***cost***, where the cost of IPhone 5's camera does not make sense). This type of errors can be solved by our method through the use of multiple evidences. As for the iphone 5 example, we will determine that the relatedness between ***iphone 5*** and ***cost*** is much stronger than that between ***camera*** and ***cost*** in term of evidences like PMI, thus a direct connection is established between ***iphone 5*** and ***cost***.

5.3.5 Case Study on Topic Hierarchy Generation

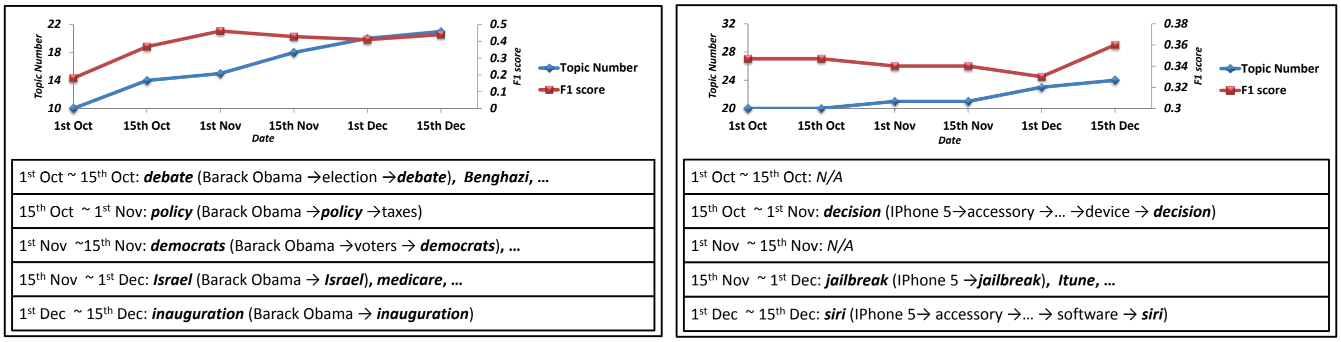
In this section, we analyse the strength and weakness of our method using a few examples. First, rather than simply assigning all sub-topics as direct children of the root topic, our method tends to generate completely structured hierarchies (average depth is 5.27). For example, (***facebook*** → ***application*** → ***business*** → ***marketing*** → ***ads***) is a sub-topic chain in the resultant hierarchy of "Facebook Inc.". Evidences from all sources contribute to this result. Pattern-based evidences support the relations that frequently occurs on web pages like ***facebook*** → ***application***. On the other hand, Wikipedia-based and WordNet-based evidences help to find many novel relations (rather than traditional *is-a* or *has-a* relation), such as ***marketing*** → ***ads*** for the given example and ***policy*** → ***tax*** for "Barack Obama". The undirected-evidences are also important. For the given example, both contextual-based and pmi-based evidences suggest that ***marketing*** and ***ads*** are more relevant than ***business*** and ***ads***. Based on all these evidences, our method can best capture the relations among all the sub-topics of a given root topic.

However, multiple sources also bring in different errors. For pattern-based evidences, when a pattern "<*topic*> such as <*subtopic*> and" matches a string " *... as well as other companies/brands/**games** such as **Bing** and Gears of War ...*", a sub-topic relation between ***game*** and ***bing*** will be detected. This is obviously wrong but is hard to be corrected by NLP tools. On the other hand, for an error sub-topic relation (***event*** → ***lipstick***) for "Chanel", we found that

[12] In fact, since these methods (including those presented in the following section) use the same topic identification process, they all share the same candidate topic sets.

Topic	*Yang's Method*			*Navigli's Method*			*Snow's Method*			*Our Method*		
	pre.	rec.	F_1	pre.	rec.	F_1	pre.	rec.	F_1	pre.	rec.	F_1
Iphone 5	0.16	0.15	0.16	0.36	0.30	**0.33**	0.30	0.27	0.28	0.36	0.31	**0.33**
Ipad mini	0.14	0.16	0.15	0.41	0.20	0.27	0.26	0.20	0.22	0.50	0.24	**0.33**
Xbox Kinect	0.10	0.14	0.12	0.50	0.23	**0.32**	0.36	0.24	0.28	0.50	0.24	**0.32**
Barack Obama	0.19	0.23	0.20	0.50	0.45	0.47	0.55	0.50	**0.52**	0.55	0.50	**0.52**
Mitt Romney	0.07	0.10	0.08	0.29	0.23	**0.26**	0.26	0.23	0.25	0.30	0.24	**0.26**
Hillary Clinton	0.21	0.40	0.28	0.33	0.27	0.30	0.41	0.33	**0.37**	0.42	0.33	**0.37**
Chanel	0.16	0.22	0.18	0.55	0.55	0.55	0.61	0.61	**0.61**	0.61	0.61	**0.61**
Estee Lauder	0.23	0.33	0.27	0.40	0.44	**0.42**	0.35	0.38	0.37	0.40	0.44	**0.42**
Facebook Inc.	0.20	0.22	0.21	0.32	0.36	0.34	0.32	0.36	0.34	0.36	0.41	**0.39**
Microsoft Corp.	0.18	0.19	0.18	0.18	0.19	**0.21**	0.14	0.18	0.15	0.14	0.15	0.18
Blizzard Inc.	0.18	0.22	0.20	0.26	0.27	0.26	0.22	0.22	0.22	0.30	0.32	**0.31**

Table 6: Performance comparison between our method and state-of-the-art methods. The bold face indicates the best F_1 performance for each topic. Our method achieves significant improvements on F_1 scores (t-test, p-value<0.05) compared to all three baseline methods.



1st Oct ~ 15th Oct: ***debate*** (Barack Obama →election →***debate***), ***Benghazi, …***
15th Oct ~ 1st Nov: ***policy*** (Barack Obama →***policy*** →taxes)
1st Nov ~15th Nov: ***democrats*** (Barack Obama →voters → ***democrats***), …
15th Nov ~ 1st Dec: ***Israel*** (Barack Obama → ***Israel***), ***medicare, …***
1st Dec ~ 15th Dec: ***inauguration*** (Barack Obama → ***inauguration***)

1st Oct ~ 15th Oct: *N/A*
15th Oct ~ 1st Nov: ***decision*** (IPhone 5→accessory →… →device → ***decision***)
1st Nov ~ 15th Nov: *N/A*
15th Nov ~ 1st Dec: ***jailbreak*** (IPhone 5 →***jailbreak***), ***Itune, …***
1st Dec ~ 15th Dec: ***siri*** (IPhone 5→ accessory →… → software → ***siri***)

Figure 6: Hierarchy update on "IPhone 5" and "Barack Obama" between 1^{st} Oct and 15^{th} Dec: The trend map shows how the F_1 score changes when the topic set grows along time; the table illustrates some examples of the newly detected topics (indicated by *Bold Italic* face) and their position in the hierarchies (shown in brackets) in each period.

the only evidence that supports it is from WordNet, since there is a WordNet path[13] between the two words.

5.4 Hierarchy Update

Online UGCs increase every minute. In this section, we demonstrate how the proposed method can be used to update the topic hierarchies incrementally with real time information. We use the data of blogs, cQAs and tweets published before 15^{th} Dec, 2012 on two example topics, *i.e.*, "Barack Obama" and "IPhone 5". According to their published date, we split the data into 7 sub sets. We initiate the hierarchy using the data before 1^{st} Oct and update it using the data in the other 6 sub sets, each indicates a duration of 15 or 16 days. As our baselines are not designed for real time data, we only show our results for the hierarchy updating experiments here.

The result in Figure 6 offers us a close look of this process. We can see that as time passes and new topics emerge, our method can effectively detect these topics and merge them into the hierarchy. As the results, topics like ***jailbreak*** for "IPhone 5", ***debate***, ***inauguration*** for "Barack Obama" are updated onto the appropriate positions of the topic hierarchy shortly after they become popular on the internet. From the results, we can also find that "Barack Obama" is a more time-sensitive topic, which brings in new sub-topics in each period. On the contrary, the change of topics on "IPhone 5" is very small during the two and a half months.

6. CONCLUSION

In this paper, we proposed an automatic method for incremental information organization for multiple UGC sources. Given a root topic, we used evidences from multiple UGCs to identify topic terms and sub-topic relations between them. With these topic terms, a graph-based algorithm was applied to generate and update the topic hierarchies, on which the UGCs can be organized according to their relevant topics. Comprehensive experiments on 11 root topics demonstrated the effectiveness of our method. For future work, we will explore more UGC sources such as forums and try to find available initial topic hierarchies to enhance our system. It is also interesting to apply the generated topic hierarchy in more sophisticated text analysis tasks.

7. ACKNOWLEDGMENTS

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office and The National Key Basic Research Program (also called 973 Program), No.2012CB316301 and NSFC under project No.61272227.

[13] The WordNet path: ***event*** → ***makeup*** → ***lipstick***.

8. REFERENCES

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

[2] Y. HaCohen-Kerner, Z. Gross, and A. Masa. Automatic extraction and learning of keyphrases from scientific articles. *Computational Linguistics and Intelligent Text Processing*, pages 657–669, 2005.

[3] X. Han and J. Zhao. Structural semantic relatedness: a knowledge-based method to named entity disambiguation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 50–59. Association for Computational Linguistics, 2010.

[4] C. Hayes, P. Avesani, and U. Bojars. An analysis of bloggers, topics and tags for a blog recommender system. *From Web to Social Web: Discovering and Deploying User and Content Profiles*, pages 1–20, 2007.

[5] J. Hoffart, F. Suchanek, K. Berberich, and G. Weikum. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 2012.

[6] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 216–223. ACL, 2003.

[7] S. N. Kim and M.-Y. Kan. Re-examining automatic keyphrase extraction approaches in scientific articles. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 9–16. Association for Computational Linguistics, 2009.

[8] Z. Kozareva and E. Hovy. A semi-supervised method to learn and construct taxonomies using the web. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1110–1118. ACL, 2010.

[9] D. J. Lawrie and W. B. Croft. Generating hierarchical summaries for web searches. In *Proceedings of the 26th annual international ACM SIGIR conference*, pages 457–458. ACM, 2003.

[10] X. Mao, Z. Ming, Z. Zha, T. Chua, H. Yan, and X. Li. Automatic labeling hierarchical topics. 2012.

[11] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[12] Z. Ming, K. Wang, and T. Chua. Prototype hierarchy based clustering for the categorization and navigation of web collections. In *Proceedings of the 33rd international ACM SIGIR conference*, pages 2–9. ACM, 2010.

[13] R. Navigli, P. Velardi, and S. Faralli. A graph-based algorithm for inducing lexical taxonomies from scratch. In *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Three*, pages 1872–1877. AAAI Press, 2011.

[14] K. Nishida, R. Banno, K. Fujimura, and T. Hoshide. Tweet classification by data compression. DETECT '11, pages 29–34. ACM, 2011.

[15] T. Pedersen, S. Patwardhan, and J. Michelizzi. Wordnet:: Similarity: measuring the relatedness of concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41. ACL, 2004.

[16] R. Snow, D. Jurafsky, and A. Ng. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808. ACL, 2006.

[17] X. Wang, K. Zhang, X. Jin, and D. Shen. Mining common topics from multiple asynchronous text streams. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 192–201. ACM, 2009.

[18] H. Yang and J. Callan. Feature selection for automatic taxonomy induction. In *Proceedings of the 32nd international ACM SIGIR conference*, pages 684–685. ACM, 2009.

[19] H. Yang and J. Callan. A metric-based framework for automatic taxonomy induction. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 271–279. ACL, 2009.

[20] A. Yates, M. Cafarella, M. Banko, O. Etzioni, M. Broadhead, and S. Soderland. Textrunner: Open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of NAACL: Demonstrations*, pages 25–26. ACL, 2007.

[21] J. Yu, Z. Zha, M. Wang, K. Wang, and T. Chua. Domain-assisted product aspect hierarchy generation: towards hierarchical organization of unstructured consumer reviews. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 140–150. ACL, 2011.

APPENDIX

A. SEARCH ENGINE-BASED EVIDENCES

In Table 7, we list the patterns we use in estimating the pattern-based evidences.

<*topic*> such as <*subtopic*> and
<*topic*>'s <*subtopic*>
<*subtopic*> on <*topic*> and
<*topic*> including <*subtopic*> and
<*subtopic*> of <*topic*>
<*subtopic*> and other <*topic*>

Table 7: Patterns used to estimate the pattern-based evidences