Finding Missing Tweets using Topic Structure and Browsing time

Yu Suzuki
Nara Institute of Science and Technology
Ikoma, Nara, Japan
ysuzuki@is.naist.jp

Hiromitsu Ohara, Akiyo Nadamoto
Konan University
Kobe, Hyogo, Japan
m1424003@center.konan-u.ac.jp
nadmoto@konan-u.ac.jp

ABSTRACT

Microblogging services such as Twitter and Facebook become popular in recent years. In these services, many users post short messages which correspond to many topics such as daily activities, opinions, and new events. Therefore, users need a system to summarize messages if the users receive tons of messages. If the following users tweet about important things which the user does not know, these tweets should be noticed. However, which tweets should be noticed is one important problem. Users should need which topics are on their timeline. However, if the summarization method does not consider topics of tweets, the summarized tweets do not contain rarely tweeted topics. To solve this problem, we propose a method for automatically extracting missing tweets based on topic granularity and missing time of the users. In this study, we map the missing tweets to the Wikipedia category tree by considering topic structure granularity; then we present the topic structures of missing tweets using our proposed visualization interface. In our experiments, we confirmed the effectiveness of our proposed hierarchal topic structure.

CCS CONCEPTS

• **Information systems** → *Social networks*; *Service discovery and interfaces*; Personalization;

KEYWORDS

twitter, browsing time, uninformed information

ACM Reference Format:
Yu Suzuki and Hiromitsu Ohara, Akiyo Nadamoto. 2017. Finding Missing Tweets using Topic Structure and Browsing time. In *iiWAS '17: The 19th International Conference on Information Integration and Web-based Applications & Services, December 4–6, 2017, Salzburg, Austria.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3151759.3151798

iiWAS '17, December 4–6, 2017, Salzburg, Austria

ACM ISBN 978-1-4503-5299-4/17/12…$15.00
https://doi.org/10.1145/3151759.3151798

1 INTRODUCTION

Microblogging services, such as Twitter[1], Facebook[2] and Instagram[3], have become popular in recent years. In these microblogging services, many users post short texts called *tweets* in Twitter. Their texts are correspond to many topics such as daily activity, opinions, and news.

One of the features of Twitter is that users follow the other users if they have something in common with her. In this paper, a user whom the user follows as *followees*. If a followee post tweets, the user receive the tweets. When a user follows many followees and the follwees post many tweets, s/he receives numerous and diverse tweets. However, when a user does not browse tweets for some periods of times, the user should lose interesting and important information in tweets which are posted at not browsed time, which we call *missing tweets*.

In Twitter, "while you were away" feature is implemented and used, and the output of our method is similar to the output of this feature. The method of selecting summarized tweets is not open to the public, but from the results of presented tweets, we can predict that this summarization method does not consider the topics of missing tweets. Therefore, if there is a topic which is rarely tweeted but important, users cannot find the topic.

There are many kinds of information which are important, unimportant, known, and unknown topic. However, it is hard for users to obtain interesting and important information from missing tweets because s/he has to search for the tweets that were posted while the user was unable to browse. We define *missing information* as the tweets that the user lost and important information, and *browsing time* as a time span which our target user do not browse tweets. We propose a system that extracts tweets which are important or unknown from lost information automatically.

One important issue of tweet summarization is for ascertaining which tweets are interesting and important for readers. For understanding many users' thoughts, it is essential for developing techniques to grasp interests of users from tweets as a tree structure. For example, there is user u_a who is interested in a presidential election in the US, especially the political opinions of Hillary Clinton, and there is another user u_b who is interested in especially the political opinions of Donald Trump. To grasp these interests, we used a clustering method for the tweets by user A and B. One typical solution to this issue is using techniques of hierarchical clusterings, such as Ward's method[16] and DIANA[7], and tweet

[1] http://twitter.com/
[2] htto://facebook.com/
[3] https://www.instagram.com

topics based method[11]. Here we count how many users are interested in the presidential election. If we do not know that Hillary Clinton and Donald Trump put up as a president in this election, we cannot count these two users. Therefore, we should use external knowledge for grasping the interests of users.

However, when we use this user's profile, we cannot understand a different hierarchy level of interests. We can say that she is interested in Real Madrid players and serial drama, or just soccer and TV program. Therefore, if we construct a user's profile as a hierarchal structure, we can understand a different level of interests. The other user follows the user because she also likes the same baseball player. However, if the user finds that the following user also interested in the soap opera by browsing the topics of the following user, the user should talk about the baseball player and even the soap opera. Therefore, it is essential for visualizing what kind of topics twitter users are interested in. When we visualize the topics of this user without considering granularities of the topics, a summary should be related to a major theme, the baseball player. However, the topics about soap drama and the actor should be unanticipated topics for this twitter user.

If we consider several tweets about the baseball player as topics for summarizing, then the actor and the soap drama are picked up and present as a set of summarized tweets. However, if the actor appears on the drama, then the tweets related to the actor and the drama should be presented at once. Tweets about the actor and the baseball player should be presented at once if the actor and the baseball player are mutual friends. Moreover, if the readers do not feel interested in the baseball player, the readers want to ignore the tweets about the baseball player. Therefore, we will make a better summary by considering the topic structures of the tweets from the unstructured tweets.

When we use these techniques for generating a summary, the systems will generate a structure from the set of twitter itself. However, when we use these existing hierarchal clustering methods, we cannot measure the difference between two user profiles accurately. Let us consider how to compare profiles of user A and B. User A's profile consists of a soccer player and serial drama, and user B's profile consists of sports and TV program. When there is no external knowledge about the fact that soccer player and sports, serial drama and TV program are related to each other respectively, we cannot find that user A and B have similar topics and interests. For example, we assume that there are a set of tweets about baseball players and basketball players. In this case, we expect the summarization system to categorize the tweets into two categories: baseball and basketball players. However, if there is no information about baseball and basketball players, we cannot categorize as we expected. We use the Wikipedia category tree as the external knowledge base to solve this problem.

In this paper, first we extract missing tweet based on missing time, and we propose a method for mapping a set of extracted missing tweets to the Wikipedia category tree by considering topic structure granularity. We can use any taxonomies as an external knowledge base for our system. However, there are many new words in tweets, a knowledge base we use should include these new words. Therefore, we select the Wikipedia category tree as an external knowledge base.

Many researchers studied about to grasp topics of tweets[2, 4, 6, 8, 9, 13, 14, 18]. These researchers do not consider about a period, but we propose the method to grasp topics structure of tweets by considering user's browsing period. A critical feature of our proposed method is that we can grasp not only which topics are heterogeneous or homogeneous with each other but also considering a missing time when extracting topics. Another feature is that the topic structures for multiple users are easy to compare with each other. We did our experiments to confirm the effectiveness of our proposed hierarchal topic structure. Therefore, in this paper, we propose a method for mapping tweets to the Wikipedia category tree considering missing time. In our experimental results, we confirmed that the averaging precision ratios for commonly known themes are about 72%, which is a sufficient accuracy for practical use. However, our proposed method is inadequate for personal, private topics, because there is no category corresponds to these topics. In fact, precision ratios for general themes, such as politics, is about 72%, and them for particular themes, such as computer and sports, are about 42-44%. Moreover, we developed a visualization interface to present the topic structures of missing tweets for quickly understanding the missing tweets.

The contributions of this paper are the following:

(1) Generate topic structures of tweets using the Wikipedia category tree considering browsing time
(2) Visualize the topic structure of tweets using a network graph
(3) Confirm that our proposed method is effective for commonly known topics

As a first step of presenting a summary of missing important tweets, we formerly have proposed a method [11] which corresponds to a part of 1. to extract missing tweets. In this paper we introduce and modify 1. and we propose 2. and 3.

2 RELATED WORK

Many researchers have studied about topic detection from Twitter[8][14]. Hong et al. [4] propose a method for topic modeling in Twitter using LDA (Latent Dirichlet Allocation) [1] and the Author–Topic model. Michelson et al. [9] detect topics using author information of tweets with categories in Wikipedia. Kasiviswanathan et al. [6] detect topics from Twitter using the dictionary learning method. Zao et al.[18] propose Twitter-LDA which is dedicated LDA for Twitter. Sasaki et al. [13] emphasize the study of variation in topic trends by time, proposing a topic model improve on Twitter-LDA. Cataldi et al.[2] propose a topic detection method with the relation of topics by author information of tweets and the topic life cycle. However, their extracted topics do not have a structure; then the topics are not related to each other. Our method constructs structured topic based on Repeated-Bisection[15] and the Wikipedia category tree.

In recent years, there are several studies about visualization of topic graphs based on topic structures. Daniil et al.[10] proposed a method of visualizing topic structures of academic search results. They use the Wikipedia category tree structure when they extract topics from results. On the other hand, we use Repeated-Bisection to extract topics from tweets, and we use the Wikipedia category tree to create a topic graph. Michael et al.[17] proposed topical semantics of twitter topic graph based on following and tweet-retweet

relationship. Paula et al.[12] proposed a method for constructing a topic graph by using URLs on the tweet, open DNS, and DBpedia. They purpose a method for constructing user's profile from Twitter. We construct a topic graph by using Repeated-Bisection, and the Wikipedia category tree as an external knowledge.

3 GENERATION OF TOPIC GRAPH FOR CONSIDERING TOPIC GRANULARITY

In this paper, we propose a method for visualizing topics in a set of missing tweets. We first extract missing tweet, and given a set of missing tweets and the Wikipedia category tree, our proposed system categorizes them by topics, constructs a topic graph, and visualizes the topic graph.

Figure 1 shows an overview of our proposed method, which consists of three steps as shown below:

(1) *Extracting missing tweet*: Extracting which tweets are submitted during user's browsing time and it is before and after.
(2) *Clustering Tweets into Categories and extracting topics*: Using Repeated Bisection as clustering tools, we divide a set of tweets into clusters and extract topics in each cluster.
(3) *Generate a topic graph*: Using the topics of tweets and the Wikipedia category tree, we generate a topic graph of the tweets.
(4) *Classify topics* Classify the topics which are nodes of the topic graph as known topics and unknown topics.
(5) *Visualization of topic graphs*: We visualize the topic graph and the corresponding tweets using our implemented Web user interface.

3.1 Clustering Tweets into Categories and extracting topics

First, given a set of tweets during user's un-browsing time and their before and after. Then, we use Repeated–Bisection [5], an extension of k-means clustering. There are two reasons to select this method: 1) in our preliminary experiments[3], this method is the most accurate for clustering especially short texts, and 2) this method can put several keywords to the clusters. For using this clustering method, we first extract feature vectors of the terms.

Given a set of tweets T, we extract a feature vector for each tweet. First, we divide a tweet into the terms using morphological analysis or POS tagger. Then, we select noun and unknown terms as feature terms. The reason of using unknown terms is that these terms consist of slang and newly invented words which are not recognized by the morphological analysis. To clean the feature terms, we select the terms which are included in more than two tweets. Feature vector $\boldsymbol{f}(t_i)$ of tweet t_i ($t_i \in T$) is defined as follows.

$$\boldsymbol{f}(t_i) = [tf(t_i, w_1) \cdot idf(w_1), tf(t_i, w_2) \cdot idf(w_2), \cdots, tf(t_i, w_m) \cdot idf(w_m)] \quad (1)$$

$$tf(t_i, w_j) = \begin{cases} 1 & \text{if } w_k \text{ appears at } t_i \\ & \text{more than once} \\ 0 & \text{else} \end{cases} \quad (2)$$

$$idf(w_j) = -\log \frac{df(w_j)}{|T|} \quad (3)$$

where w_j is a term in T, $|T|$ is the number of tweets in T, $tf(t_i, w_j)$ indicates whether w_j appears at t_i or not, $df(w_j)$ is the number of tweets which have w_j, and $idf(w_j)$ is an IDF (Inverted Document Frequency) value of w_j where a document is a tweet. When we extract feature vector, we only use IDF and do not use TF (Term Frequency). This is because each tweet has less than 140 characters, there is almost no term which appears more than twice in one tweet. Then, we input the feature vectors $\boldsymbol{f}(t_i)$ into the clustering method. In this method, we divide the tweets in T into the clusters C, where one tweet belongs to one cluster. For example, if we input a set of tweets about two baseball teams to this clustering system, then the outputs are two clusters of tweets related to baseball teams, and the name of a baseball team. However, we did not understand the relation between two baseball teams.

Finally, we prune sparse clusters because of noisy clusters. When one cluster has tweets about various kinds of topics and each topic has a small number of tweets, we should prune the cluster. Because, there is no appropriate label for the cluster. For this decision, we calculate a density of each cluster, the average value of cosine similarity values for any of two tweet feature vectors in the cluster. We define $d(c)$ as a density of $c \in C$ as follows:

$$d(c) = \frac{1}{|c|^2 - 1} \sum_{t_p \in c} \sum_{t_q \in c, q \neq p} len(t_p, t_q) \quad (4)$$

$$\text{where } len(t_p, t_q) = \frac{\boldsymbol{f}(t_p) \cdot \boldsymbol{f}(t_q)}{|\boldsymbol{f}(t_p)| \cdot |\boldsymbol{f}(t_q)|} \quad (5)$$

where $|c|$ denotes the number of tweets in c. We prepare the threshold σ, and we remove c from C if $d(c)$ is less than σ. We do this process for all clusters in C. We call each cluster as *tweet cluster*.

After we make clusters, we extract topics from each tweet cluster. If a tweet cluster has multiple topics, we regard the value of topics in a cluster is more than threshold α as a topic.

3.2 Generation of a Topic Graph

In this process, we construct an undirected graph G called *topic graph* from the clustered tweets and the Wikipedia category tree, which is shown in the upper right side of Fig.1. G has a set of nodes N and a set of undirected edges E. A node $n \in N$ represents a topic. There are two types of nodes: *topic node* and *semantic node*. *Topic node* is a topic of tweet cluster which is extracted in section 3.1. *Semantic node* is a parent node of the topic nodes and it is the high-level semantics of the topic node which is extracted from Wikipedia. Edge $e \in E$ represents a relation between topic nodes and semantic nodes. The semantic nodes can be connected to the other semantic and topic nodes; the topic nodes can be connected with the semantic nodes.

3.2.1 Generation of a topic node. First, we generate a topic node for each topic of tweet cluster. We construct *topic nodes* by identifying an article $w(c) \in W$ which corresponds to c, where W is a set of all articles in Wikipedia. c has a set of keywords with related degrees $L(c) = \{(l_1, d_1), (l_2, d_2), \cdots, (l_N, d_N)\}$, where l_i is a keyword and d_i is a related degree of l_i with c. Related degree of keyword l_i for category c means how l_i is adequate as a keyword for expressing the category c. For example, if there is a category about baseball team, the keyword "baseball" has a high related degree and

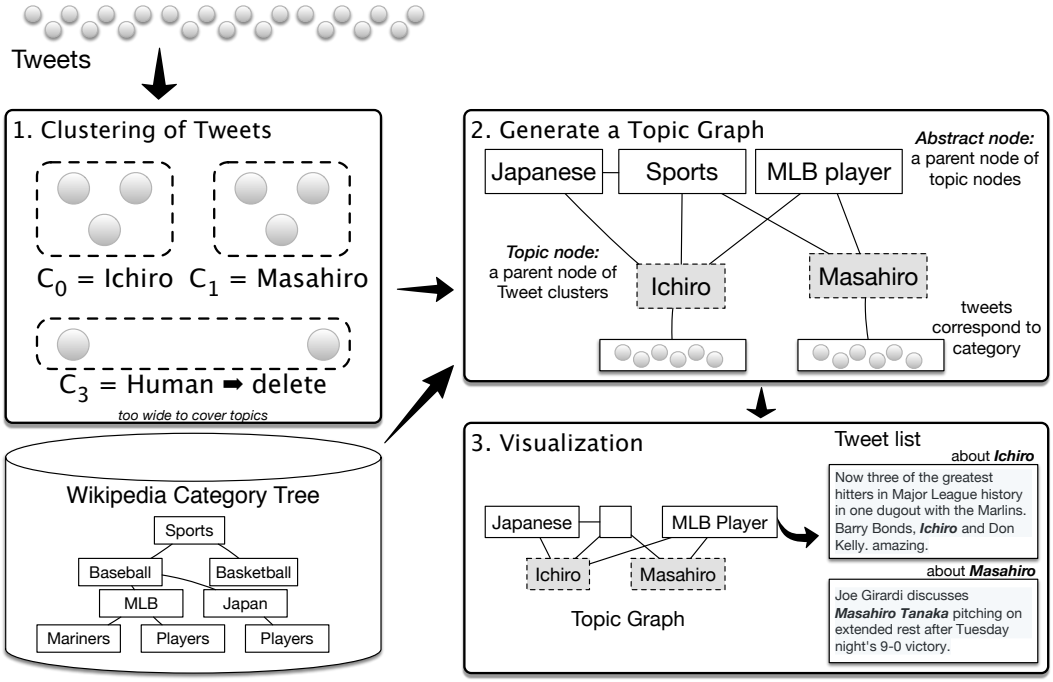

Figure 1: Overview of our proposed method.

the keyword "human" has a low related degree. These values are extracted using the clustering method we used. Then, we calculate similarity value $s(c, w)$ between the weighted keyword sets $L(c)$ with article $w \in W$ as follows:

$$s(c, w) = \sum_{i=1}^{N} d_i \cdot r(l_i, w) \tag{6}$$

$$r(l, w) = \begin{cases} 1 & \text{if } l_i \text{ is in the title of } w \\ 0 & \text{else} \end{cases} \tag{7}$$

These equations show that if there are common terms l in a list of keywords in $L(c)$ and a title of the article w, we add the related degree d which corresponds to l, to the score of w. For example, we select a category $c \in C$ which has a list of keywords with the related degrees $L(c) = \{(\text{“}baseball\text{”}, 1), (\text{“}player\text{”}, 0.5)\}$. In addition, there are two Wikipedia articles $w_p, w_q \in W$. The title of w_p is "baseball team" and w_q is "baseball player." In this case, the category scores are calculated such as $s(c, w_p) = 1.0$ and $s(c, w_q) = 1.5$. Then, w_p is selected as the corresponding article with c. As a result, we select w_p as a *topic node* $w(c)$ of c.

For example, we have two clusters c_0 and c_1. A cluster c_0 is tagged by "Ichiro Suzuki," and c_1 is tagged by "Kenta Maeda." In c_0, many tweets are related to Ichiro Suzuki, a famous Japanese baseball player who played Major League Baseball (MLB). Using the steps presented above, the corresponding Wikipedia article $w(c_0)$ of c_0 is the article "Ichiro Suzuki[4]." Then, we put a node $w(c_0)$ to $G(c_0)$.

[4]https://en.wikipedia.org/wiki/Ichiro_Suzuki

3.2.2 Generation of an semantic node. Next, we construct *semantic nodes* by mapping a topic node $w(c)$ to the Wikipedia category tree. For example, we assume that there are two topic nodes $w(c_0)$ of cluster c_0 and $w(c_1)$ of c_1, where $c_0, c_1 \in C$. c_0 is about "Ichiro Suzuki" and c_1 is about "Kenta Maeda." Both Ichiro Suzuki and Kenta maeda are a famous Japanese baseball player currently playing MLB The goal is to add nodes related to these two topics respectively.

$w_i^a(c_j)$ means the i-th semantic node of $w(c_j)$. The Wikipedia article that corresponds to $w(c_0)$, which is the article of "Ichiro Suzuki," belongs to two categories $w_0^a(c_0)$, "Yankees Players," and $w_1^a(c_0)$, "Baseball Players." Then, we put two nodes "Yankees Players" and "Japanese Baseball Players" into graph $G(c_0)$ as semantic nodes. As a result, we generate two graphs $G(c_0)$ and $G(c_1)$ presented in the left side of the graph in Figure 2.

c_1 is related to "Kenta Maeda." $w(c_1)$ is the article of "Kenta Maeda," and this article belongs to two categories "Los Angeles Dodgers" and "Japanese Baseball Player." As a result, we generate two graphs $G(c_0)$ and $G(c_1)$ presented into the left side of the graph in Figure 2.

Finally, we connect a topic node with semantic nodes. In $G(0)$, there is one topic node $w(c_0)$ and two semantic nodes $w_0^a(c_0)$ and $w_1^a(c_0)$. We respectively connect $w(c_0)$ to $w_0^a(c_0)$ and $w(c_0)$ to $w_1^a(c_0)$.

(a) Creating the smallest topic graph

First, we transform a topic in a topic cluster into the smallest graph which consists of a topic of a cluster, and its high-level semantics. If a topic cluster has multiple topics, we transform a topic cluster into multiple topic graphs. In this paper, we call the topic graph as "the smallest topic graphSTG_j". j is a number of a topic. A leaf node

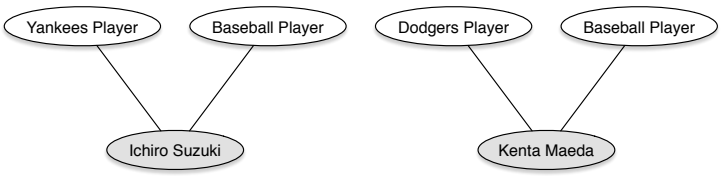

Figure 2: Example of network graph $G(c_0)$ and $G(c_1)$.

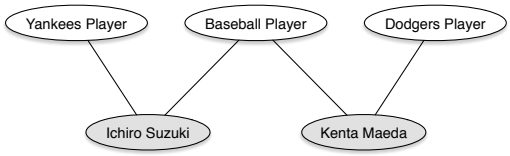

Figure 3: Example of connected network graph $G(c_0, c_1)$, a connected graph of $G(c_0)$ and $G(c_1)$.

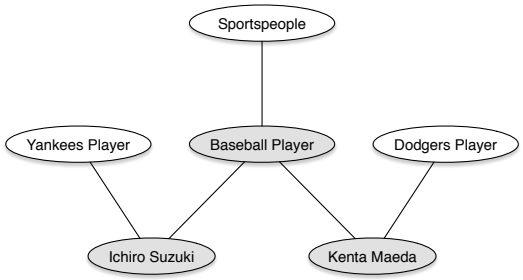

Figure 4: Example of connected network graph $G(c_0, c_1, c_2)$, a connected graph of $G(c_0)$, $G(c_1)$, and $G(c_2)$.

of all STG_j is the topic$C = \{c_1, c_2, \ldots, c_n\}$ (show Figure **??**). STG_j is a labeled graph. The label of the leaf node is a topic of the topic cluster. Non-leaf nodes in STG_j are high-level semantics of a leaf nodec_x. We regard the semantics of c_x is a category of Wikipedia, that means label of non-leaf nodes in STG_j is the semantics of c_x. When we extract heigh-level semantic of a c_x from Wikipedia, we use Wikipedia category link information database[5]. We regard category s_{xi} as a high-level semantic of c_x. In addition, i is a number of category of c_x. We also extract high-level semantic s_{xim} of s_{xi} from Wikipedia category link information database. We create $n-th$ high-level semantic to create the smallest topic graph. As described in this paper, n is 2.

In this time, the category words which contain "Wiki", "stub" and "user" are deleted from the category, because these words are used to manage Wikipedia, and these are not appropriate for using high-level semantics. We also delete "Living people" and "XX-language surnames", which have no vital meaning. Then, the leaf node of STG_j is a topic of the topic cluster, and non-leaf nodes of STG_j are high-level semantics of a leaf node(topic). For example, there are two topics of "Ichiro" and "Masahiro Tanaka". First, we create the smallest topic graph about "Ichiro" as c_1. We search article about "Ichiro" from Wikipedia. We extract categories, which are "American League stolen base champions" as s_{11}, and "Japanese Major Leaguers" as s_{12}, as high-level semantics from the Wikipedia category link information database. We also extract two-hop high-level semantics of topic from Wikipedia category link information database. Then we create the smallest topic graph related to "Ichiro"(c_1). Next, we search each article about "Masahiro Tanaka"(c_2) from Wikipedia. We also extract categories which are "Japanese Major Leaguers"(s_{21}) and "Olympic baseball players of Japan" (s_{22}) as high-level semantics and extract two-hop high-level semantics of topic Then we create the smallest topic graph related to "Masahiro Tanaka".

The number of the smallest topic graph is a total number of topics of all topic clusters. After we create all smallest topic graphs, we create topic graphs based on joining at the same nodes.

3.2.3 Merge Multiple Graphs. Next, we merge the same nodes into one node. In Figure 2, two graphs $G(c_0)$ and $G(c_1)$ share the same node "Baseball Player." Therefore, we merge these two nodes into one node. A merged graph $G(c_0, c_1)$ is presented in Figure 3.

We assume that there is a graph $G(c_2)$, and that there are two nodes $w(c_2)$ and $w^a(c_2)$. $w(c_2)$ is "Japanese Baseball Player" which $G(c_0, c_1)$ also shares, and $w^a(c_2)$ is "Sportspeople." In this case, we merge $G(c_0, c_1)$ and $G(c_2)$ by merging the node "Baseball Player." The node "Baseball Player" in $G(c_0, c_1)$ is an semantic node, but that in $G(c_2)$ is a topic node, then this node is treated as a topic node in $G(c_0, c_1, c_2)$ because of the definition of topic node.

We can consider all ancestors of the nodes. However, if we use these ancestor nodes, our method will connect unrelated nodes. For example, if there is a node about "Steve Jobs," a founder of the Computer company, and the parent node is "Computer company," this node will not connect to $G(c_0, c_1, c_2)$. However, if we use all ancestor nodes, this node will connect to the node "People in the U.S.," and this connection will not be useful for many readers. To avoid this problem, we only use a parent node. How many ancestor nodes are useful for readers is still an open problem.

3.3 Determining type of topic node based on time interval

We consider that there are two types of missing information that are partially-known information and complementary-unknown information. We define the former as *known topic* and the latter as *unknown topic*. The definition of known topics and unknown topics as follows.

[5]http://dumps.wikimedia.org/jawiki

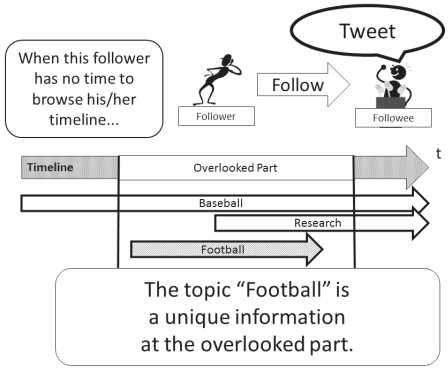

Figure 5: Image of the known topic and unknown topic.

- Known topic
  A known topic is a topic of tweets. Some tweets are posted during a browsing user browsing time; some tweets are posted during a user non-browsing time. Then, a browsing user knows part of the topic.
- Unknown topic
  An unknown topic is a topic of tweets posted only during the user non-browsing period. A browsing user does not know information at all.

Figure 5 presents an image showing the known topic and unknown topic. In Figure 5, the followee tweets about part of "baseball" and "research" during a browsing user's browsing time and the browsing user know part of "baseball" and "research" topics. They become known topics. On the other hands, the followee tweets about "football" during a browsing user's un-browsing time, the browsing user does not know about information of "football" which is tweeted by followee at all. The topic of "football" becomes an unknown topic. We consider a browsing user can understand roughly known topic information that is tweeted his/her un-browsing time because he/she already browsed the same topic of tweets posted during the user browsing time. In this case, we consider it is suitable for the browsing user to present an outline of the topics; then we present the topic structure of the missing information to him/her. On the other hands, for an unknown topic, it is difficult for a browsing user to understand all contents of topic clearly because the tweets were posted during a time when the browsing user cannot browse. We consider that it is necessary to present information that has a greater detail that a user can discover the full breadth of the topic than in the case of a known topic.

We determine topic nodes in a topic graph; they are the known topic or unknown topic. First, we check time stamp of tweets in the topic nodes. If a time stamp of a tweet is browsing time of a browsing user, the tweet becomes known. On the other hands, if a time stamp of a tweet is a non-browsing time of a browsing user, the tweet becomes unknown tweet. After we extract each time stump of a tweet, we next determine a type of topic node. When at least one known tweet includes a topic node, the topic node becomes the known topic. On the other hands, when all tweets in a topic node are an unknown tweet, the topic node becomes the unknown topic.

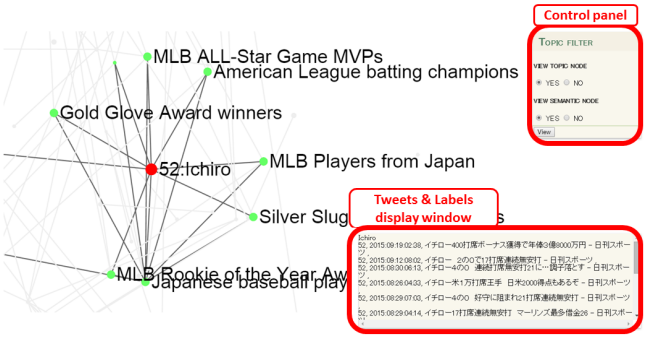

Figure 6: Visualization of a topic structure.

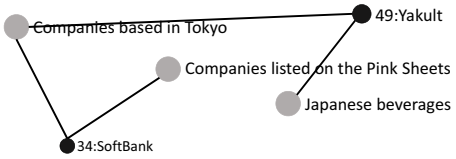

Figure 7: Example of cluster-by-topic.

3.4 Visualization of cluster-by-topic

Finally, we visualize a generated topic graph using a Web interface. Figure 6 shows a part of a topic graph. In this graph, a green node represents a semantic node. A label beside a node is the name of Wikipedia articles or categories in the Wikipedia category tree. Topic nodes represent multiple colors based on their type. Unknown topic node presents light blue color, and know topic node is changed based on missing time about the topic. Purple nodes present long missing time about the topic, pink nodes present middle missing time, and red nodes present short, missing time about the topic

Using the control panel in the upper right part of this figure, a user can select whether topic/semantic nodes appear or not. There is a window for viewing tweets and labels in the lower left part of this figure. When a user clicks a topic node, the user can browse a set of tweets corresponding to the topic node. When a user clicks a semantic node, the user can browse a set of tweets which correspond to topic nodes connected to the semantic node.

If all nodes present to the browsing user, it is difficult to grasp all topics. Then, we prepare control panel that browsing user can select presenting node types such as semantic node and topic node(See right upper side of Figure 6). When browsing user click topic node, the system presents tweets in the node (See right lower side of Figure 6).

When a user uses our system, the user uploads a set of tweets into our system. Then, our system automatically generates the topic graph and presents it to the users. There are many nodes and their related labels. Then if the user is interested in a topic of a node, the user clicks the node. Then the user can browse a set of tweets related to the topic.

4 EXPERIMENTAL EVALUATION

As described in this section, we conducted experiments to assess the accuracy of mapping tweet clusters to Wikipedia category.

4.1 Experimental Setup

We set the following five themes: Politics, Music, Computer, Sports, and Animation/Games for collecting tweets for evaluations. We prepared 10, 000 tweets using the Twitter Search API, 2, 000 tweets for each theme. All tweets are written in and by Japanese. We conducted our experiments using five steps as follows:

(1) Clustering 2,000 tweets for each theme, and extracting topics of each cluster
(2) Generate the topic graph using our proposed method
(3) Give clusters and their corresponding Wikipedia article titles to the observers.
(4) Observers evaluate whether the article titles are appropriate or not for representing the clusters using the following five degrees (5: appropriate, 4: almost appropriate, 3: cannot say, 2: almost inappropriate, 1: inappropriate).
(5) Summarize the observer's evaluations, and analyze whether our proposed method has good accuracy or not

We implemented our proposed system as a Web application using PHP5. We use "bayon[6]" as an implementation of Repeated Bisection, a clustering method for tweets, as described in section 3.1. The parameter of divided point for bayon is 1.0. The threshold of cosine similarity σ is 0.5. We set these parameters by preliminary evaluation.

At step 4., observers are collected using Crowdworks[7], a popular crowdsourcing platform in Japan. All of these observers know about the themes and their topics. The number of observers for each theme is presented in the second column in Table 1.

We give a set of tweets and titles of their corresponding Wikipedia articles to the observers; then they select whether the titles are appropriate as a label of tweets. The policy of this decision is that the observers should assign higher degrees if they can presume the correspondence category or article name from the set of tweets. Therefore, if an assigned correspondence category widely covers the topics, the category should earn high evaluation score. For example, if a set of tweets are about several Major League baseball players, and the title is "MLB Player," an observer should select 5: appropriate. In this step, the observers do not understand whether the category is the most appropriate or not, then there may be more appropriate categories in the Wikipedia category tree. Therefore, if the correspondence category is "Baseball," an observer should also select 5, but if the correspondence category is "Basketball Team," an observer should select 1.

Finally, we calculate precision ratio p^t of theme t using the following equation:

$$p^t = \frac{|C^t_{corr}|}{|C^t|} \tag{8}$$

where C^t is a set of categories in theme t, and C^t_{corr} is a set of categories in C^t which receive the evaluation scores more than 3.0 by the observers. $|C^t|$ is number of clusters in C^t, and $|C^t_{corr}|$ is number of clusters in C^t_{corr}.

We predicted before we did this experiment that our proposed method is useful for commonly known themes that are commonly submitted by many Twitter users. Indeed, such tweet clusters will correspond to some Wikipedia articles. For example, if user posts about baseball or basketball, there are many articles about baseball and basketball, then the cluster will successfully connect to the appropriate article. However, we also predicted that our proposed method is ineffective for private themes which are submitted by a few users. For example, if a cluster is on someone's website, and if the website is not popular, then there is no corresponding Wikipedia article. In this case, we cannot generate a node and their appropriate parent nodes.

We prepared the following three assumptions. We confirmed which assumptions are correct based on the experimentally obtained results.

- Effective for the themes "Politics," "Music," and "Computer" because these tweets invariably have topics that have an adequate size.
- Not effective for the theme "Animation/Games," because these tweets invariably have overly specific topics, such as the names of the characters and their abbreviated forms.
- Sometimes effective for the theme "Sports," but sometimes ineffective because several topics of sports have an adequate size of topic granularity but several topics do not.

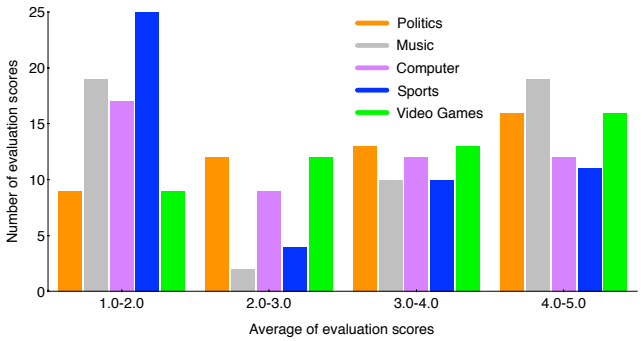

Figure 8: Number of evaluation scores vs. Average of evaluation scores.

Table 1: Numbers of evaluation scores for respective bins.

Theme	# observers	Precision ratio
Politics	8	0.72
Music	11	0.56
Computer	5	0.44
Sports	5	0.42
Animation/Games	4	0.52

4.2 Experiment Results

Table 1 shows the precision ratio of our proposed method for each theme. Figure 8 presents results obtained using our proposed method. We discuss which cases are effective and ineffective by analyzing the detail of results.

4.2.1 Effective Case. In theme "Politics," our proposed method works well. The main reason is that there are many technical terms about politics in the Wikipedia categories and articles. We found

[6]https://code.google.com/p/bayon/
[7]http://www.crowdworks.jp

that many of the terms about politics do not have multiple meanings, multiple semantics.

In theme "Animation" and "Games," our proposed method works well for several clusters. Especially, the clusters related to characters in animations and games earn appropriate correspondence Wikipedia articles. This is because there are many articles about Animation and Games in Japanese Wikipedia. There are many redirect pages in Wikipedia, and these pages solve the problem about disambiguation of topics.

4.2.2 Ineffective Case. In theme "Sports," our proposed method does not perform well. The main reason for this case is that there is many people's name as topic names. For example, there is a famous baseball player "Ichiro Suzuki," but there is also a famous car and motorcycle company "Suzuki Motor Corp." The name of two objects share the term *Suzuki*, but there is no relationship with each other. Therefore, if there is a topic cluster about Ichiro Suzuki, our system may assign differen Wikipedia article about Suzuki Motor Corp. To solve this problem, we should consider the semantics of topic words and the Wikipedia category tree.

In several themes, the nodes with little or no relationship are connected with each other. This is because heterogeneous clusters are integrated into the Wikipedia category tree which covers a wide range of topics. For example, there are two categories of two different persons, and these persons do not have a relationship with each other. However, almost all Wikipedia articles about persons are included in the category *Living People*. This case occurs in the other categories, such as *Name of Area* and *Japanese*. To solve this problem, we should select which categories are appropriate or not for topic/semantic nodes.

5 CONCLUSION

In this paper, we proposed a method for automatically extracting user's missing tweets based on topic granularity and missing the time of browsing user. Specifically, first we extract missing tweet based on missing time, and we propose a method for mapping a set of extracted missing tweets to the Wikipedia category tree by considering topic structure granularity. Therefore, one crucial feature of our proposed method is that we can grasp not only which topics are heterogeneous or homogeneous with each other but also considering a missing time when extracting topics. Another feature is that the topic structures for multiple users are easy to compare with each other.

We did our experiments to confirm the effectiveness of our proposed hierarchal topic structure. From our experiments, we confirmed that our proposed method is effective for commonly known themes such as politics, music, and computer. In the experiments, a precision ratio for these themes is about 52-72%. This is because there are Wikipedia categories and articles which correspond to the tweet topics. However, sometimes users post tweets about their friends and colleagues, which do not correspond to the Wikipedia category, and this decreases the precision ratios.

One important feature of our proposed method is that Twitter users can capture heterogeneous topics, because we handle extra knowledge bases, the Wikipedia category tree, to construct a hierarchical structure from an unstructured textual data source. Many methods of generating a summary of unstructured data use a structure from a data source itself by using hierarchical clustering method. For example, if there are many tweets about baseball players and a few tweets about basketball players, we cannot distinguish who is a baseball player or basketball players without an external knowledge base. Using the knowledge base, we can identify which players are baseball players.

Our proposed system will be effective for summarizing not only tweets but also the semantics of papers. When we adopt our proposed method for summarizing papers, we will discover implicit relations of papers. However, we should prepare the other external knowledge base for this case, because the Wikipedia category tree does not always have technical terms. Therefore, we should develop a method to select an appropriate external knowledge base automatically. For example, if a user summarizes papers about biology, the system automatically selects Gene Ontology (GO).

A problem of our method is we did not use synonyms of the terms; we sometimes assign inappropriate Wikipedia articles and categories to tweet categories. Therefore, if "Kobe," a name of the city in Japan, is a topic word for a tweet cluster, our proposed method may assign the Wikipedia article "Kobe Bryant," a famous basketball player in the U.S. and of course not a name of the city. To solve this problem, we should capture the semantics and synonyms of topic words and the title of Wikipedia articles.

Another problem is that we did not use posting time of tweets. Thus we cannot capture the transition of topics. For example, if a user tweets about baseball player A at any time, and she also tweets about baseball play B at a specific time, we should not integrate these tweet clusters even if the clusters of these topics are the same. To solve this problem, we should use posting time information for clustering, and we also develop a method for visualizing the transitions of tweet topics.

In our experiments, the target language is Japanese. However, our proposed method is language independent. Therefore, we will do our experiment with the other languages. We assume that if we use a large size of external knowledge bases, such as an English version of Wikipedia category tree, the accuracy of our proposed method will increase.

ACKNOWLEDGMENT

This work was partly supported by NAIST Bigdata Project. The research results have been achieved by "Research and Development on Fundamental and Utilization Technologies for Social Big Data", the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* 3 (March 2003), 993–1022. http://dl.acm.org/citation.cfm?id=944919.944937

[2] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. 2010. Emerging Topic Detection on Twitter Based on Temporal and Social Terms Evaluation. In *Proceedings of the Tenth International Workshop on Multimedia Data Mining (MDMKDD '10)*. ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/1814245.1814249

[3] S. Hanai and A. Nadamoto. 2014. Clustering for Similar Recipes by using cooking ingredient. *IEICE technical report* 114, 204 (2014), 47–52.

[4] L. Hong and B. D. Davison. 2010. Empirical Study of Topic Modeling in Twitter. *Proceedings of the First Workshop on Social Media Analytics (SOMA)*. (July 2010), 80–88.

[5] Michael Steinbach George Karypis, Vipin Kumar, and Michael Steinbach. 2000. A comparison of document clustering techniques. In *TextMining Workshop at KDD2000 (May 2000)*.

[6] Shiva Prasad Kasiviswanathan, Prem Melville, Arindam Banerjee, and Vikas Sindhwani. 2011. Emerging Topic Detection Using Dictionary Learning. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM '11)*. ACM, New York, NY, USA, 745–754. https://doi.org/10.1145/2063576.2063686

[7] L. Kaufman and Peter J. Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley.

[8] Michael Mathioudakis and Nick Koudas. 2010. TwitterMonitor: Trend Detection over the Twitter Stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10)*. ACM, New York, NY, USA, 1155–1158. https://doi.org/10.1145/1807167.1807306

[9] Matthew Michelson and Sofus A. Macskassy. 2010. Discovering Users' Topics of Interest on Twitter: A First Look. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data (AND '10)*. ACM, New York, NY, USA, 73–80. https://doi.org/10.1145/1871840.1871852

[10] Daniil Mirylenka and Andrea Passerini. 2011. Navigating the topical structure of academic search results via the Wikipedia category network. In *Proceedings of the 22nd ACM international conference on Information and Knowledge Management(CIKM '13)*. 891–896.

[11] Hiromitsu Ohara, Yu Suzuki, and Akiyo Nadamoto. 2015. Followee Recommendation Based on Topic Extraction and Sentiment Analysis from Tweets. In *International Conference on Information Integration and Web-based Applications and Services(iiWAS2015)*. 215–225.

[12] Paula Peñas, Rafael del Hoyo, Jorge Vea-Murguía, Carlos González, and Sergio Mayo. 2013. Collective Knowledge Ontology User Profiling for Twitter – Automatic User Profiling. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT) - Volume 01 (WI-IAT '13)*. IEEE Computer Society, Washington, DC, USA, 439–444. https://doi.org/10.1109/WI-IAT.2013.62

[13] K. Sasaki, T. Yoshikawa, and T. Furuhashi. 2014. Online Topic Model for Twitter Considering Dynamics of User Interests and Topic Trends. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. (October 2014), 1977–1985.

[14] Bharath Sriram, Dave Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas. 2010. Short Text Classification in Twitter to Improve Information Filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10)*. ACM, New York, NY, USA, 841–842. https://doi.org/10.1145/1835449.1835643

[15] M. Steinbach, G. Karypis, and V. Kumar. 2000. A comparison of document clustering techniques. *In 6th ACM SIGKDD,World Text Mining Conference* (2000).

[16] Joe H. Ward. 1963. Hierarchical Grouping to Optimize an Objective Function. *J. Amer. Statist. Assoc.* 58, 301 (1963), 236–244. http://www.jstor.org/stable/2282967

[17] Michael J. Welch, Uri Schonfeld, Dan He, and Junghoo Cho. 2011. Topical Semantics of Twitter Links. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining (WSDM '11)*. ACM, New York, NY, USA, 327–336. https://doi.org/10.1145/1935826.1935882

[18] W. X. Zhao, J. Jiang, J. He J. Weng, E. Lim, H. Yan, and X. Li. 2011. Comparing Twitter and Traditional Media using Topic Models. In *European Conference on Information Retrieval (ECIR 2011)*.