Gist of a Thread in Social Network Services Based on Credibility of Wikipedia

Akiyo Nadamoto  
Konan University  
8-9-1, Okamoto, Higashinada,  
Kobe, 658–8501 Japan  
nadamoto@konan-u.ac.jp

Yu Suzuki  
Nagoya University  
Furo, Chikusa,  
Aichi, 464–8601, Japan  
suzuki@db.itc.nagoya-u.ac.jp

Takeshi Abe Kaw  
National Institute Informatics  
Hitotubashi 2–1–2, Chiyoda,  
Tokyo, 101–8430, Japan  
abekawa@nii.ac.jp

Abstract

Users of Social Network Services (SNS) can sometimes enter into heated discussions, which prompt those users to concentrate on a single issue and lose track of the actual theme. We believe that it would be beneficial for users and visitors to present information to help understand the gist of the discussion at a glance. As described in this paper, we propose a system that presents a gist of a thread on an SNS and basic information about it by comparing the comments in the thread with Wikipedia article. Wikipedia articles, however, are not always credible. When we compare a thread on an SNS with Wikipedia, the Wikipedia article must have credible content. We measure the credibility of the article based on the credibility of Editors. We first extract the target passage which is candidate of the gist of a thread in an SNS based on the Wikipedia Table of Contents (TOC). Then we measure the credibility of editors of Wikipedia using the edit history and measure the credibility of the article using results of the credibility of editors. The target passage, which has a high similarity degree with comment in an SNS and has a high credibility rate becomes the gist of the thread in SNS. Consequently, users and visitors can ascertain the gist of an SNS thread by viewing a Wikipedia TOC with credibility.

1. Introduction

Social Network Services (SNS) are a representative technique of Web2.0. An SNS consists of multiple threads; each thread contains comments posted by multiple users who are a member of a community. Users can sometimes enter into heated discussions, prompting them to concentrate on a single issue and lose track of the actual theme. In such cases, each user might want to know how relevant a point is to the discussion. Moreover, it is difficult for visitors who just visit and browse the SNS to grasp the gist of a thread. Therefore, we believe that it would be beneficial for users and visitors to present information to help understand the gist of the discussion at a glance.

Some research has been done on creating summaries from Web pages [18, 19, 7]. The research created summarized sentences from Web using Natural Language Processing. However, it is difficult to use the technique for extracting the gist of an SNS because almost all comments are short and the conversation is frequently confusing. If users and visitors were able to understand the gist of an SNS thread at a glance, it would be convenient for them. The content of an SNS is written from a community viewpoint; it might be a biased viewpoint. When we extract the gist of thread in an SNS, we should compare comments in a thread with other content which has general viewpoint. Wikipedia contents are posted by different users on the basis that a “Neutral point of view is the fundamental principle of Wikipedia.” We therefore consider Wikipedia content on a given theme to be based on a general viewpoint. In this study, we choose an article from Wikipedia and compare each comment in a thread in the community content with the smallest structure in the table of contents (TOC) of the article. We consider the TOC in Wikipedia articles to be the gist and the content of a paragraph on Wikipedia. As described in this paper, we propose a system that presents a gist of a thread on an SNS and basic information about it by comparing the comments in the thread with that of a Wikipedia article. As described herein, we define a “target thread” to be a thread from which the gist is to be extracted, and define a “target article” as an article on Wikipedia that is to be compared with the target thread, and define a “target passage” as a candidate of a gist of comments in target article.

An important feature of Wikipedia is that any user can edit articles. Therefore, even if editors incorrectly edit articles, Wikipedia will not refuse these edits. Consequently, Wikipedia articles are not always credible [9]. When we compare a thread in SNS with Wikipedia article, the

1See http://en.wikipedia.org/wiki/NPOV/
Wikipedia article should have credible content.

To solve this problem, users and administrators revise and re-edit less credible and inadequate edits by hand. However, articles in Wikipedia are becoming increasingly numerous, so controlling the quality of articles by hand is difficult because the increase in the number of edits is not proportional to the quality of articles [14]. In fact, we can easily find clearly wrong sentences in Wikipedia. To resolve this problem, we need an application to assess the credibility of articles. This application would be useful for users who do not know which articles are credible or should be improved. This application would also be useful for users who use Wikipedia as a credible source, and who have insufficient knowledge of the topics. Users use Wikipedia to retrieve information about which they have insufficient knowledge. Therefore, users cannot determine whether the articles are credible. Consequently, they often misunderstand information and receive incorrect information from Wikipedia.

We use a reputation-based credibility degree assessment method. Adler et al. [1, 2, 3] and Hu et al. [8] proposed several methods for calculating article credibility values based on the respective editors’ credibility values. In these methods, if the lifetime of versions is long, then the versions should be treated as credible. However, an important drawback of these reputation-based systems is that all editors are regarded equally, even if an editor edits many articles only once. We believe that we should distinguish among users who frequently edit high-quality versions and those who edit low-quality versions.

The process of extracting the gist and credibility information is the following (Figure 1):

1 Extracting gist candidates for a target thread.
   1.1 Extract keywords from the target thread.
   1.2 Identify target articles in Wikipedia.
   1.3 Compare the target thread and target articles based on the TOC in the latter.
   1.4 Extract the gist of and basic information about the target thread.

2 Calculating the credibility of the target passage.
   2.1 Extract edit histories of each article from Wikipedia archive data.
   2.2 Extract Editors and their statistics from edit histories.
   2.3 Calculate credibility values for each Editor.
   2.4 Calculate credibility values of the target passage in Wikipedia.

3 Display the gist of the target thread with credibility.

This paper is organized as follows: Section 2 discusses related work, Section 3 presents a description of the extraction of the gist of a target thread, Section 4 presents a description of the calculation of the credibility of the target passage, Section 5 presents a prototype system, Section 6 gives a discussion of results of experiments conducted using our system. Section 7 presents the conclusions of our study.

2 Related Work

Summarization

In order to determine the differences between two Web pages, Seung-Jin et al.[12] proposed a semantic change-detection (SCD) algorithm to detect semantic changes between two bodies of HTML data. Their approach was based on the transformation of HTML data from the two sources into trees and the removal of common edges between the two trees. This algorithm was mainly intended for detecting important updates made to a Web page. Nadamoto et al.[13] proposed CWB and B-CWB. In this study, we have focused on examining how the similarities (or differences) between two Web sites can be effectively presented page by page. In order to extract gist of a thread of an SNS, we have compared a thread of an SNS with Wikipedia content. Li et al.[11] proposed a method for enhancing diversity, coverage, and balance in the summarization of content. Their viewpoints were similar to that of our approach; however, their target of summarization of content was different from that of ours.

In the field of natural language processing (NLP), dialogs have mainly been studied using carefully annotated transcription data such as dialog act markup in several layers (DAMSL)[6] and graph-based dialog annotation[16]. This is true of discourse studies as well, and various annotation schemes such as RST-DT[4] and discourse graph-bank[22] have been proposed. In this paper, we do not iden-
tify dialogs in a thread, but simply perform a comparison between a thread and Wikipedia content.

Credibility

There has been much research on the credibility of Wikipedia articles [17]. We focus on the automatic or semi-automatic credibility value calculation methods for Wikipedia. When we extract data from Wikipedia articles, three methods are mainly used, a user voting system of articles, version analysis using natural language processing techniques, addition and deletion of edit history.

A popular method of using user voting was proposed by Kramer et al. [10]. For this method, they developed Mediawiki2 and a user voting system was added. In this system, users can directly vote on which articles have better quality. However, many votes are needed to calculate the credibility of articles. Moreover, this system cannot calculate new articles’ credibility values, which have not been read and not voted on by users. In our system, we do not need a voting mechanism, and our system can calculate credibility values even if the articles have not been read.

Wöhner et al. [21] propose a system for calculating credibility value of article using lifecycle of articles. Wöhner et al. think that editor’s addition and deletion can classify into several patterns. Therefore, using these patterns, Wöhner’s system distinguish which patterns are credible or not. This idea is similar to editor’s credibility value calculation method we used. However, this system do not consider the editors’ behavior, only consider addition and deletion of articles. In our system, we consider addition of deletion of articles for each editor. Therefore, we can analyze the changes of articles deeper than Wöhner’s system.

Adler et al. [1, 2, 3], Hu et al. [8] and Wilkinson et al. [20] proposed a system to calculate credibility values using edit history. These authors implemented this system to FireFox and MediaWiki plug-ins as the WikiTrust module [5] 3. This system calculates the credibility values in real time. Our credibility calculation system is based on Adler’s. However, their system do not calculate equivalent credibility scores to editors. In our system, we improve the accuracy of credibility scores using normalization techniques.

3 Extracting Candidate of Gist of Target Thread

3.1 Extracting Keyword from a Target Thread

When we compare the target thread with a target article, we use keywords from the target thread. Generally, there are few sentences in a comment, and it is difficult to carry out a comparison by using simple keywords. Therefore, we use our topic structure model[13]. The model is based on Oyama’s topic structure model[15]. Our topic structure consists of a subject and content terms. According to our model, the topic structure $t_p$ for a given thread $P$ is simply represented as a pair of a set of subject terms $S$ and a set of content terms $C$. $S$ and $C$ consist of multiple subject terms $s_i, i \in \{1, \ldots, n\}$ and content terms $c_{jj}, jj \in \{1, \ldots, n\}$. Namely, the topic structure $t_p$ of a thread $P$ is represented as follows:

$$t_p = (S, C)$$
$$S = (s_1, \ldots, s_m)$$
$$C = (c_1, \ldots, c_n)$$

Extracting subject terms

The subject term refers to the most dominant term in the thread. It is identified by considering the frequency of terms. We define “subject degree” as the degree of probability that a keyword is a subject term. The subject degree of a word $w_i$ is determined by its term frequency. The subject degree $\text{sub}(w_i)$ of a keyword $w_i$ within a given thread $P$ is defined as follows:

$$\text{sub}(w_i) = tf(w_i) \times \text{weight}(w_i) > \alpha$$  (1)

where $tf(w_i)$ denotes the term frequency of $w_i$ in the target thread, $\text{weight}(w_i)$ denotes the weight of $w_i$, and $\alpha$ denotes the threshold. In our prototype system, we arrange proper nouns, numbers, and common nouns in the increasing order of their weights. In cases where all nouns and proper nouns have the same weight, the weights of nouns comprising numbers and numerical classifiers tend to be higher. The weights assigned to each type of noun in our experiment are proper nouns as 3.0, numbers and numerical classifiers as 0.1, common nouns as 1.0, and other nouns as 0.9. The alpha assigned in our experiment is 2.5. Word vector is defined as the product of word frequency and weight of word.

Extracting content terms

The content term is intuitively a term that has a high cooccurrence relationship with a certain subject term on the thread.

3.2 Identifying Target Articles on Wikipedia

Occasionally, the granularity of information in a thread is different from that in a Wikipedia article. This implies that the information in a thread may sometimes be related to different articles on Wikipedia. We consider three types of target articles. We use all types of target articles when we extract a gist and basic information. When we extract three types of target articles, we create a Wikipedia link graph.

(1) Title-based target articles

We first identify a target article with a theme matching the
keyword of a community. We refer to the target article as a title-based target article. When we extract other target articles, we use this article as a standard.

2) Link-based target articles
Link-based target articles are close to title-based target articles on the link graph of Wikipedia. We first identify articles whose distance from title-similar articles is one path on the link graph. The selected articles become a candidate for link-based target articles. We identify a link-based target by three types of links.

- Interactive linked article:
  Interactive linked article has an inlink and outlink to the title-based target article. We consider it is important to the theme of the thread, we regard the interactive linked article as link-based target article.

- Outlink article:
  Outlink article is linked from the title-based target article. If the title of the outlink article appears in a target thread many times (greater than threshold $\beta$), the article is considered a link-based target article (see Figure 2).

- Inlink article:
  Inlink article links to title-based target article. We consider that it is not important to the title-based article, we ignore it.

3) Content-based target articles
There are three link types: interactive link, outlink, and inlink. Even if the articles are not close to a title-similar article on the link graph, they may have similarities with the thread. We call these articles content-similar target articles. In this case, we first extract candidate of content-similar target articles from the link graph. If the title of an article appears many times (greater than threshold $\beta$) in the thread but is not an immediate neighbor, it is considered a candidate of content-similar target article. We extract content-similar target articles from all the candidates for articles on the basis of the degree of coverage. To calculate the degree of coverage, we calculate the similarity between all paragraphs in a candidate content-similar target article and all comments in a target thread in a round-robin manner. (see Figure 1) The degree of coverage is the ratio of similar paragraphs to all paragraphs. We calculate similarity by using a topic structure and a cosine vector function. Our degree of coverage of a surrounding article $\text{Cov}(SA_i)$ is as follows:

$$\text{Cov}(SA_i) = \frac{SCA_i}{PCA_i} \tag{2}$$

where $SA_i$ is a candidate for content-similar target article and $SCA_i$ is the total number of similar paragraphs in the content-similar target article. $PCA_i$ is a binomial number between the number of comments in the target thread and the number of paragraphs in the content-similar target article. If $\text{Cov}(SA_i)$ is greater than the threshold $\delta$, the article is considered to be a content-similar target article.

3.3 Comparison between Target Threads and Target Articles

After identifying the target article, we compare the comments in the thread with small passages in the target article. The small passages are divided on the basis of the TOC of the article. First, we calculate the weight of each subject term and content term using $tf/idf$. Then, we compare a comment in a thread $CP_a$ with the target passage $TaP_k$, $k \in \{1, \ldots, mn\}$ of the article by using a cosine vector as follows:

$$\text{Sim}(CP_a, TaP_k) = \frac{\overrightarrow{F}(CP_a) \cdot \overrightarrow{F}(TaP_k)}{|\overrightarrow{F}(CP_a)| \cdot |\overrightarrow{F}(TaP_k)|} \geq \gamma \tag{3}$$

where $\overrightarrow{F}(CP_a)$ is a feature vector of $CP_a$ and $\overrightarrow{F}(TaP_k)$ is a feature vector of $TaP_k$. If $\text{Sim}(CP_a, TaP_k)$ is greater than the threshold $\gamma$, the target passage in the article is the candidate of the gist of the comment. Therefore, the target passage is considered to contain basic information about the thread. We call the target passage in Wikipedia which is a candidate of the gist of the comment is a target passage.
4 Calculating the Credibility of the Target Passage

After extracting the target passage, we calculate each credibility of them. We first assess the appropriateness of edits. Next, we assess credibility values of editors. Then, we assess credibility values of versions using the editors’ edit histories. When we assess the appropriateness of edits, we first present our Wikipedia model for analysis. Based on this model, we describe the meaning and measurements of edit appropriateness, editors, and versions.

4.1 Modeling Wikipedia Edit History

We define several notations used in the following sections. Wikipedia includes multiple target passages TaPk, k ∈ {1, . . . , nn} in an article, and when Wikipedia editors write articles, a new articles version vi,j is created. In this paper, we regard the target passages version in the article as same as the article version. i is an article index and j is a version number. vi,1 is the first created version of an article i. That means vi,1 is TaPk in i. We define vi,0 as a target passage version with no content.

We define the Wikipedia editors e = 1, 2, · · · , E who edit article more than once. Ae = {vi,j | vi,j ∈ V and vi,j is edited by e} is the set of article versions edited by e. One article version is edited by one editor. We do not set the editor of the articles with version number j = 0.

4.2 Editor’s Contribution to Article Versions

In our system, we first assess whether the version from vi,j−1 to vi,j is appropriate, and calculate the editor’s contribution values of article edits τ(vi,j). We define the appropriateness values of articles as the ratio of unchanged edits and non-reverted deletions after other editors’ edits. This is because, when the editor adds appropriate words to the article, these words should remain and not be deleted by other editors. In the same way, when an editor removes inappropriate words from articles, these words should not be reverted by other editors.

We show an example of appropriate and inappropriate edits in Figure 3. First, we identify the edits from vi,j−1 and vi,j using diffs. addi,j is the set of added words and deli,j is the set of deleted words. In this example, addi,j consists of “Kunio” and “Prime Minister”, and deli,j consists of “Yukio” and “President”.

Next, we identify the remaining edits of addi,j and deli,j at vi,j+p, where p = 1, 2, · · · , Ni − j. We define δ(addi,j, p) as the set of words addi,j in vi,j+p, and δ(deli,j, p) as the set of words deli,j in vi,j+p. When we identify δ(addi,j), we compare vi,j and vi,j+p, and extract the same parts of articles. However, when we identify δ(deli,j), we have a problem of determining whether the added words are new or reverted. In our system, we identify which articles are reverted because if articles are reverted articles, the same articles appear as past-edited articles.

We define the contribution ratio of additions Radd(i, j, p) and that of deletions Rdel(i, j, p) as follows:

\[ R\text{add}(i, j, p) = \frac{\left| \delta(\text{addi,j}, p) \right|}{\left| \text{addi,j} \right|} \] (4)
\[ R\text{del}(i, j, p) = \frac{\left| \delta(\text{deli,j}, p) \right|}{\left| \text{deli,j} \right|} \] (5)

where \(|\delta(\text{addi,j}, p)|\), \(|\delta(\text{deli,j}, p)|\), and \(|\text{addi,j}|\), \(|\text{deli,j}|\) are the number of words in \(\delta(\text{addi,j}, p)\), \(\delta(\text{deli,j}, p)\), \(\text{addi,j}\), and \(\text{deli,j}\).

The problem is that the contribution ratio of addition \(R\text{add}(i, j, p)\) and that of deletion \(R\text{del}(i, j, p)\) is not equivalent when we change p, because if the number of edits after version vi,j increases, the contribution of addition and deletion should be high. Therefore, we normalize \(R\text{add}(i, j, p)\) and \(R\text{del}(i, j, p)\) using the standard addition and deletion ratio. We define the standard addition/deletion ratio as the ratio of edits that remains/reverts after p versions.

The normalized contribution ratio of addition and deletion, \(\bar{R}\text{add}(i, j, p)\) and \(\bar{R}\text{del}(i, j, p)\), is defined as follows:

\[ \bar{R}\text{add}(i, j, p) = \frac{R\text{add}(i, j, p)}{S\text{add}(p)} \] (6)
\[ \bar{R}\text{del}(i, j, p) = \frac{R\text{del}(i, j, p)}{S\text{del}(p)} \] (7)

where \(S\text{add}(p)\) and \(S\text{del}(p)\) is the average value of \(R\text{add}(i, j, p)\) and \(R\text{del}(i, j, p)\) for each p in all i and j.

Next, we calculate the editor’s contribution degrees of article edits \(\tau(vi,j)\) using \(R\text{add}(i, j, p)\) and \(R\text{del}(i, j, p)\) as
The value of $N_i$ depends on article $i$; however, we do not normalize $\tau(v_{i,j})$ using $N_i$ because we assume that if the number of edits increases, the credibility values of edits should increase.

Finally, we normalize the values $\tau(v_{i,j})$ for converting the average value of all $\tau(v_{i,j})$ to 0. This is because almost all edits are small, and these edits do not change the credibility values of articles. Therefore, if the credibility value of an edit is lower than the average value of $\tau(v_{i,j})$, we should determine that the edit decrease the credibility value of the article. Using this normalization, the credibility values for these edits are converted to negative values. We convert the normalized value of $\tau(v_{i,j})$ follows:

$$\overline{\tau}(v_{i,j}) = \tau(v_{i,j}) - \frac{\sum_{i=1}^{M} \sum_{j=1}^{N_i} \tau(v_{i,j})}{\sum_{i=1}^{M} N_i}$$

Using this credibility value of edits, we calculate the credibility values of editors.

### 4.3 Credibility values of editors

We set $A_e$ as the set of versions edited by $e$. Then, we calculate the credibility values of editors as follows:

$$U_e = \frac{\sum_{v_{i,j} \in A_e} \overline{\tau}(v_{i,j})}{|A_e|}$$

where $|A_e|$ is the number of versions in $A_e$, which means the number of edits by $e$.

We think that this credibility value of editors is useful even if an article is involved in the edit war. In this formula, if $e$ edits $i$ more than twice, $U_e$ as the credibility values of $e$ by $i$ is a summation of each credibility value of the version. Therefore, if first version is deleted by another low credible editor, $e$ will revert this deletion. As a result, $U_e$ which is calculated by only first version is almost the same as $U_e$, which is calculated using the first version, the reverted version by another editor, and the reverted version by $e$.

### 4.4 Credibility values of target passage

Finally, we assess the credibility values of version $v_{i,j}$. We define that the credibility values of a target passage is the weighted averaging value of editors who edits the target passage. The weights are the number of letters in version $v_{i,j}$ written by $e$.

We calculate $T_{i,j}$ as a credibility value of $v_{i,j}$ as follows:

$$T_{i,j} = \frac{\sum_{k=1}^{j} U_e \cdot c_{i,k}}{\sum_{k=1}^{j} c_{i,k}}$$

where $E(v_{i,j}) = \{ e | e \in E \}$ is the set of editors who edit version $v_{i,j}$, and $c_{i,j}$ is the number of letters edited.

If $T_{i,j}$ is greater than threshold, the target passage becomes the gist of the thread.

## 5 Prototype System

We developed a prototype system called “Gist of SNS System (GSS).” Figure 4 shows the user interface of the prototype. We used Microsoft C# to implement the prototype. In Figure 4, the left-hand window in (a) shows the Wikipedia article that contains the keywords input by the user and the left-hand window in (b) shows the results obtained with the prototype. We used the topic of the content structure list to identify an accurate gist of the thread.

An overview of our system is as follows:

1. A user inputs the theme of the thread that he/she wants to use as a keyword for comparison.
2. GSS displays the target Wikipedia article on the left window; the list of candidates in the SNS thread whose theme matches the user’s input is displayed on the right window (Figure 4(a)).
3. The user selects the SNS thread which he/she wants to compare. If there is more than one candidate of Wikipedia article, the user selects target article from the left window.
4. The SNS thread is extracted on the basis of the keyword entered by the user.
5. The GSS extracts keywords from the target thread on the basis of the topic structure.
6. The GSS extracts the target articles that consists of title-based, link-based, and content-based target articles.
7. The GSS then compares the target comment with each small passage in the target articles on the basis of the TOC by using the topic structure and extracts the content discussed in the target thread. When the small passage has high similarity degree, it becomes candidate of target passage.
8. In step 7, the system calculates the number of comments in the target thread.
9. The GSS calculates credibility values for each editors.
10. The GSS calculates the credibility values of target passage.

11. The GSS presents a similar title from Wikipedia as the gist of the thread (Figure 4(b)). It also presents multiple articles in each tab in the left window.

12. In the left window, the user selects the tab which he/she want to get the gist of a information.

13. When the user clicks a heading in the TOC in the left window, the system displays the Wikipedia content.

6 Experiments

We conducted three experiments. One experiment was conducted to examine the extraction of the gist of an SNS thread content. The next was performed to assess the accuracy of the gist of an SNS thread using the credibility value. The third examines the feasibility of our proposed system.

6.1 Experiment1: Accuracy of extracting the gist of an SNS without Wikipedia’s Credibility

We first measured the efficiency of the proposed extraction of the gist of a thread without calculating the credibility of Wikipedia. We select measured keywords: movie titles, location names, and organization names. In the dataset, each thread of number 2 discusses a theme that is wider than No1. Wikipedia is likely to have articles about most such proper nouns in the SNS thread. In our experiment, threshold of subject term is 4.5, content term is 0.01, coverage degree is 0.5, and similarity is 0.1. Column 4 in Table 1 presents results of our experiments. It is apparent that our method is suitable for use with avatars and Japanese baseball teams. However, the location names, which are Kauai Island and the Palace of Versailles, are not encouraging. In these cases, users who are members of the SNS are only “passing through.” They discuss very few topics, which degrades the accuracy of our system. From our experiment, it is apparent that new words and community words pose a major challenge to our system because many new words and community words exist in the SNS thread; our system cannot understand these words. In the future, we intend to develop a system that can understand words of these types.

6.2 Experiment2: Accuracy of extracting the gist of an SNS with Wikipedia’s Credibility

It is difficult to measure the accuracy of credibility of Wikipedia, because we can not extract all missed sentences in articles. In this paper, our purpose is creating the gist of an SNS with Wikipedia’s credibility, then we calculated the accuracy of our proposed system based on comparing the
results of Experiment 1. The dataset was identical to that used for Experiment 1. We normalized the credibility value from 0.0 to 1.0, and its threshold was 0.1 in our Experiment 2. Column 5 in Table 1 shows results of Experiment 2. The results of Palace of Versailles and Japanese baseball team are higher than the results of Experiment 1. Almost target articles have high credibility, the system removes low credibility target passage and the precision becomes high. In this case, our system beneficial for obtain the gist of the target thread.

On the other hand, the precision of Avatar and Kauai is lower than their results of Experiment 1, because their credibility of target passage is very low. For example, Avatar article in Wikipedia is written by many newcomers, because the Avatar is sudden burst word. In this case, our method calculates credibility of target passages which are low. When a target thread discuss about mainly Avatar article and discuss about few another articles, our system removes low credibility value target threads from the gist of a target thread and the precision becomes low. However, our method can return correct results over time, because we can get the credibility value of the newcomers. This is our problem, and near future we should solve it.

6.3 Experiment 3: User experiment

We performed user experiments involving our system. There were ten subjects. Five subjects used an SNS on a daily basis. We regarded these subjects as inside users. The other five subjects did not use an SNS on a daily basis. We regarded these subjects as outside users. Figure 5 shows the results of the experiment. First, the five inside users dis-
Table 1. Accuracy of extracting the gist of an SNS with/without Wikipedia’s Credibility

<table>
<thead>
<tr>
<th>Data</th>
<th>Number of target article</th>
<th>Precision (%) without</th>
<th>Precision (%) with</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avatar1</td>
<td>5</td>
<td>68</td>
<td>46</td>
</tr>
<tr>
<td>Avatar2</td>
<td>6</td>
<td>65</td>
<td>45</td>
</tr>
<tr>
<td>Kauai Island1</td>
<td>5</td>
<td>47</td>
<td>36</td>
</tr>
<tr>
<td>Kauai Island2</td>
<td>7</td>
<td>52</td>
<td>46</td>
</tr>
<tr>
<td>Palace of Versailles</td>
<td>7</td>
<td>43</td>
<td>52</td>
</tr>
<tr>
<td>Palace of Versailles</td>
<td>5</td>
<td>47</td>
<td>52</td>
</tr>
<tr>
<td>Japanese baseball team(Hanshin Tigers)1</td>
<td>6</td>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>Japanese baseball team(Hanshin Tigers)2</td>
<td>6</td>
<td>63</td>
<td>68</td>
</tr>
</tbody>
</table>

cussed Avatar. They generated a total of 80 comments. After one week, the five inside users used our system to obtain the gist of their discussion, and were able to answer our question. At the same time, the five outside users also used our system to successfully answered the same question.

Most of the users said that our system was useful for obtaining the gist of the thread. The five outside users were more enthusiastic about the system that the five inside users.

The users noted the following advantages and drawbacks:

**Inside user**

<Advantages>

- This system is useful when I want to make new comment, because we can get information about the old comments at a glance.
- I can get information that we did not discuss.

<Drawbacks>

- There are so many target pages, it is difficult to show all of them.
- The window is too small, so I cannot view all the information.
- It missed some information that we didn’t discuss.

**Outside user**

<Advantages>

- The system enables me to understand the gist of the thread, when I use the system.
- The thread is too long, and I can’t read and understand it, but the system is very convenient for getting the gist.
- It is good for me to get basic information from Wikipedia, when we obtain the gist of the thread.

<Drawbacks>

- The calculation time is too slow, at about 3 minutes.

Given the feedback, we should study the calculation time in the near future. In this experiment, we evaluated only one content. In the future, we will do the same experiment based on other target titles and analyze the results.

7 Conclusion

As described in this paper, we proposed a system that presents the gist and basic information about a thread on an SNS by comparing the comments in the thread with a Wikipedia article. We designate the system as the “Gist of SNS System (GSS).” The GSS first extracts the candidate of a target passage in Wikipedia that is similar to the comments in the SNS thread; then it calculates the credibility of target passage. When the credibility of target passage is greater than a predetermined threshold, its title becomes the gist of the thread in SNS. We conducted experiments of three kinds to assess the feasibility of our proposed system.

Future work will include the following tasks:

- Extracting community words and new words from the thread and apply to our system.
- Experiment to assess utility using bulk data.
- Consideration of the calculation time.

Acknowledgments.

A part of this study was supported by a Grant-in-Aid for the Information Explosion Project (Number: 21013044, 21013026) and Grants-in-Aid for Scientific Research (Number: 20300036, 20500104).

References


