Assessing Quality Score of Wikipedia Articles using Mutual Evaluation of Editors and Texts

Yu Suzuki
Graduate School of Information Science, Nagoya University
Furo, Chikusa, Nagoya, Aichi 4648603, Japan
suzuki@db.ss.is.nagoya-u.ac.jp

Masatoshi Yoshikawa
Graduate School of Informatics, Kyoto University
Yoshida-Honmachi, Sakyo, Kyoto 6068501, Japan
yoshikawa@i.kyoto-u.ac.jp

ABSTRACT

In this paper, we propose a method for assessing quality scores of Wikipedia articles by mutually evaluating editors and texts. Survival ratio based approach is a major approach to assessing article quality. In this approach, when a text survives beyond multiple edits, the text is assessed as good quality, because poor quality texts have a high probability of being deleted by editors. However, many vandals, low quality editors, delete good quality texts frequently, which improperly decreases the survival ratios of good quality texts. As a result, many good quality texts are unfairly assessed as poor quality. In our method, we consider editor quality score for calculating text quality score, and decrease the impact on text quality by vandals. Using this improvement, the accuracy of the text quality score should be improved. However, an inherent problem with this idea is that the editor quality scores are calculated by the text quality scores. To solve this problem, we mutually calculate the editor and text quality scores until they converge. In this paper, we prove that the text quality score converges. We did our experimental evaluation, and confirmed that our proposed method could accurately assess the text quality scores.

Categories and Subject Descriptors

H.1.2 [**Models and Principals**]: User/Machine Systems

Keywords

Wikipedia; Quality; Peer Review; Vandalism; Edit History

1. INTRODUCTION

Wikipedia[1] is a famous Internet encyclopedia, and is one of the most successful and well-known User Generated Content (UGC) websites. Any user can edit any article, Wikipedia has more and fresher information than existing paper-based encyclopedias. Many experts submit texts in Wikipedia, and the texts should be informative for readers. However, due to huge number of Wikipedia articles, many texts are not reviewed by experts, so the number of poor quality texts has also dramatically increased. On the other hand, many readers cannot easily identify texts which are good quality or not, because not all readers are experts. Therefore, there is a need for automatically identifying which articles in Wikipedia are good quality or not.

In this paper, we use the survival ratio based approach for calculating text quality scores, which is one of the major approaches for measuring text quality scores[3]. We measure the number of times which editors decide the text should remain, which is a key idea of survival based approach. If many readers feel excellence for a text, the quality of this text is good, but if many readers feel that a text should be removed, the quality of this text is poor. Adler et al. [2] found that 79% of poor quality texts are short-lived. We can estimate from this result that if editors find poor quality texts, many editors remove them.

They assumed that the quality of article becomes good according to the number of edits, because all editors delete only poor quality texts. However, this assumption is not always true because of edits by vandals, because these vandals delete not only poor quality texts but also good quality texts. If vandals delete a text, the survival ratio of the text is overly decreased. To avoid the effects by vandals, we need to detect which editors are vandals and which are not, and readjust survival ratios of texts in accordance with the editor quality scores. However, the editor quality score is calculated by the text quality score, and the text quality score is calculated by the editor quality score. Therefore, calculating the text quality score using the editor quality score is the chicken-or-egg problem.

To solve this problem, we propose a method for mutually calculating text quality scores using both survival ratios of texts and editor quality scores. We define an editor quality score as the average text quality scores written by the editor. However, text quality scores are calculated on the basis of editor quality scores. In short, one quality score is calculated by another quality score. Therefore, it is hard to calculate the text quality scores using editor quality scores. To solve this problem, we first set editor quality scores as constant values and calculate text quality scores. Next, we calculate the editor quality scores by using text quality scores. Again, we calculate the text quality scores using the editor quality scores. In this way, we mutually calculate editor and text quality scores. Using this method, we can calculate a text

[1] http://www.wikipedia.org/

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

ACM 978-1-4503-2263-8/13/10 ...$15.00.
http://dx.doi.org/10.1145/2505515.2505610.

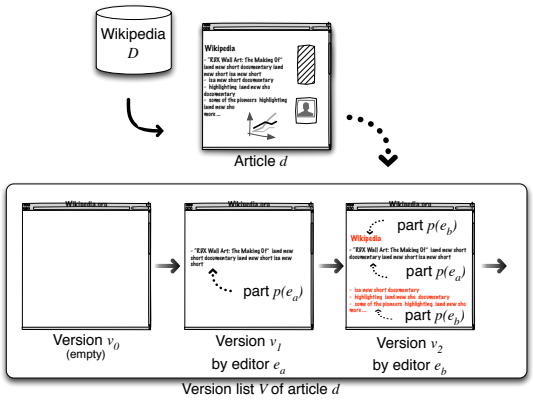


Figure 1: Notations used in this paper.

quality score that takes into consideration its editor quality scores.

2. RELATED WORK

Implicit features are the user's decisions which the system presumes from their behaviors. When the system uses these features, users do not need to input the evaluation of items. Our proposed method uses this method. However, how can users' evaluations be presumed from their behavior?

Adler et al. [3, 2, 1] and Wilkinson et al. [7] proposed a method for calculating quality scores from edit histories. This method is based on survival ratios of texts. Hu et al. [5] also proposed a method for calculating article quality score using editor quality score, which is similar to our proposed method. This method focuses on unchanged content, and they assumed that if an editor unchanged texts, the editor treated the texts as good texts. However, this method does not treat deleted texts, then if texts are deleted by vandals, the quality score of the article is overly decreased. In our system, we focus on deleted texts and editors who delete the texts. Therefore, our proposed system has resistant to vandalism of deletes such as illegitimate blanking.

3. PROPOSED METHOD

Our key idea is that we adjust the text quality scores by using the editor quality scores. We believe that the survival ratio of text is an important factor, but a quality score of editor who deletes a text is also an important factor for calculating text quality score. We assume that vandals rarely write good quality texts, so their text quality scores should be low. Therefore, if an editor deletes a text and has a low quality score, we adjust the decreased survival ratio of this text so that it increases, because this deletion should be considered inappropriate. By using our proposed method, the accuracy of text quality scores should be improved.

3.1 Modeling

In this section, we define notations that are used throughout this paper as shown in Figure 1. On Wikipedia, every article has a version list $V = \{v_i | i = 0, 1, \cdots, N\}$ where i is the version number, and v_N is the latest version. We denote that if $i = 0$, v_0 is a version with empty contents and no editor. When an editor e creates a new article, the system automatically makes two versions, v_0 and v_1, and then the system stores the text of editor e in v_1 which consist

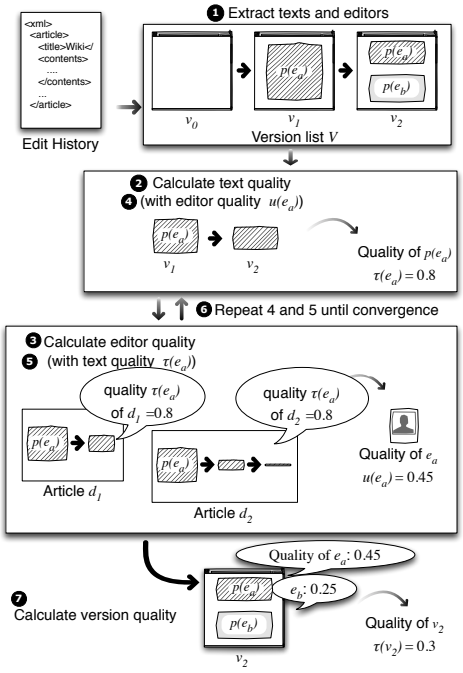


Figure 2: Overview of our proposed method.

of one text $p(e)$. We identify editors using editor names or IP address for anonymous editors. Next, we define version $v_i = \{p(e)|e \in E\}$ as a set of complete texts where E is the set of all Wikipedia editors. $p(e)$ is a text added by editor e. If e deletes all texts from i-th version, v_i is an empty set.

When editors edit one article more than twice consecutively, the system keeps the last version and deletes the other versions created by the editor. That is, the editor of a version and that of next version are always different.

The aim of our proposed method is calculating text quality score $\tau(e)$ of $p(e)$. To accomplish our mission, we should calculate converged text quality score $\tau_K(e)$ by editor e, and converted editor quality score $u'_K(e)$ of editor e. $\tau_0(e)$ is an initial text quality score, and $u'_0(e)$ is an initial editor quality score. K is a number of processes of $\tau_k(e)$ and $u'_k(e)$ converges. In step 6. at section 3.2, we mutually calculate k-th text quality score $\tau_k(e)$ and k-th editor quality score $u'_k(e)$ until convergence. k is the repeated count of these iteration processes.

3.2 Calculation Method of Text and Editor Quality Score

Figure 2 shows the overview of calculating text quality scores. Our proposed system consists of the following seven steps. (1) Extract articles from Wikipedia edit history, and identify texts and their editors from edit history. (2) Calculate initial text quality score using survival ratios of texts. (3) Calculate initial editor quality scores using text quality scores. (4) Calculate adjusted text qualities scores using both the editor and the text quality scores. (5) Calculate the editor quality scores using the text quality scores. (6) Repeat processes (4) and (5) until the text quality score converges. (7) Calculate version quality scores using the converged version quality scores.

3.2.1 Extract Texts and Editors from Edit History

First, we extract all articles from the Wikipedia edit history, and identify which editor edited which texts. Edit history stores the extract title, editor's name, and a snapshot of the article for every version. We extract these data, and store them in a database system. At this time, we identify the editors of the texts using diffs. The texts that editors have added are the texts that differ between the current and previous versions. When a text is not in the previous version but is in the current version, the text must have been written by the editor of the current version. Following this idea, we identify the editor of every text.

When we extract versions, we should consider the versions which are reverted by the other versions. In this case, when we simply use this policy, we identify the editor of the current (reverted) version who wrote the text that differs between the current and previous versions. However, if this reversion is during an edit war, the survival ratio of the text decreases, leading the text quality scores to decrease. To solve this problem, we identify the editors of a reverted version to be the editors of the original version, and the editors who revert the articles back to their previous versions are treated as neither adding nor deleting anything. Using this policy towards reversions, the text quality scores are not affected by vandalism or inappropriate edit warring. In section 3.3, we discuss why we use this method.

3.2.2 Initial Text Quality Score

Next, we define the text quality score $\tau_0(e)$ in article d by editor e as follows:

$$\tau_0(e) = \sum_{p(e)\in\bar{P}} \log_2 \left(|p(e)| + 1\right) \tag{1}$$

where $\bar{P}$ is a set of texts which is not on the version edited by e, and $|p(e)|$ is the number of letters in $p(e)$. We remove the number of letters on the version edited by e himself/herself because of the policy of non-self-evaluation. This equation means the summation of the number of letters on texts that are written by e.

3.2.3 Initial Editor Quality Score

We define the initial editor quality score $u_0(e)$ of editor e as follows:

$$u_0(e) = \frac{\sum^{D(e)} \tau_0(e)}{|D(e)|} \tag{2}$$

where $D(e)$ is a set of Wikipedia articles that e edits, and $|D(e)|$ is the number of articles in $D(e)$. When we calculate $u_0(e)$, we remove texts of articles that are created for specific purposes, such as notes, rules of Wikipedia, editors' private articles, and so on. This is because editors mainly write these texts to express their opinions and do not always delete them. Therefore, the quality scores of these texts tend to be higher than those of general articles.

We normalize $u_0(e)$ to range between 0 and 1 as follows:

$$u'_0(e) = \frac{u_0(e) - \min_{e'\in E} u_0(e')}{\max_{e'\in E} u_0(e') - \min_{e'\in E} u_0(e')} \tag{3}$$

3.2.4 Text Quality Score

Next, we calculate the text quality score using the editor quality score. This phase is derived from initial text quality score calculation method written in section 3.2.2. In this phase, we integrate the survival ratio of texts and those of the editors who delete them using weighted summation, whereas the initial quality score calculation method only use the survival ratio of texts.

We calculate the text quality score $\tau_k(e)$ as follows:

$$\tau_k(e) = \sum_{p(e)\in\bar{P}} \log_2 \left(|p(e)| + 1 + \alpha \sum_{e'\in E'(p(e))} h_{k-1}(e, e') \right) \tag{4}$$

$$h_k(e, e') = |\delta(e')| \cdot (1 - u'_k(e')) \tag{5}$$

where $h_k(e, e')$ is the adjustment of survival ratio by e' to e. $e' \in E'(p(e))$ is an editor who deletes $p(e)$, $\delta(e')$ is the letters in $p(e)$ deleted by e', $|\delta(e')|$ is the number of letters in $\delta(e')$, and α $(0 \leq \alpha \leq 1)$ is the parameter to control the effect of editor quality score. $u'_k(e')$ is the editor quality score of e'. Equation (4) is the initial text quality score which is the same as section 3.2.2 if $\alpha = 0$, and $h_k(e, e')$ means the adjustment of survival ratio, the number of deleted letters with quality scores of editors who delete them. If an editor e' who has a poor quality score deletes a text $p(e)$, then $h_k(e, e')$ is high, the value of $\tau_k(e)$ is almost the same as $\tau_0(e)$. Therefore, if editor quality score is low, the editor quality score does not affect the text quality score. In this case, if the editor who deletes the text has a good quality score, $h_k(e, e')$ has a low value. Thus, the value of $\tau_k(e)$ decreases more than $\tau_{k-1}(e)$.

3.2.5 Editor Quality Score using adjusted Text Quality Score

Using adjusted text quality score $\tau_k(e)$, we define the editor quality scores $u_k(e)$ of e as follows:

$$u_k(e) = \frac{\sum^{D(e)} \tau_k(e)}{|D(e)|} \tag{6}$$

This equation is almost the same as the equation (2) described in section 3.2.3.

We normalize $u_k(e)$ to range between 0 and 1 as follows:

$$u'_k(e) = \frac{u_k(e) - \min_{e'\in E} u_k(e')}{\max_{e'\in E} u_k(e') - \min_{e'\in E} u_k(e')} \tag{7}$$

We repeat the processes in sections 3.2.4 and 3.2.5 until the values of $\tau_k(e)$ and $u'_k(e)$ converge.

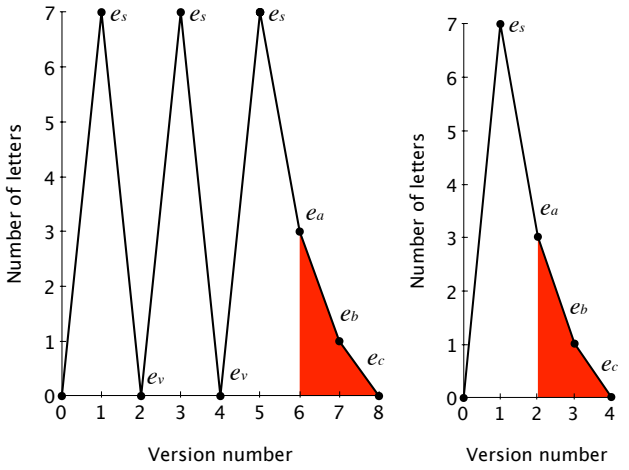
In our experiment, we confirm that the text and editor quality scores converge.

3.2.6 Quality Scores of Versions

Using $u'_K(e)$, we define the version quality score $T(v_i)$ of version v_i as follows:

$$T(v_i) = \frac{\sum_{e\in E(v_i)} u'_K(e) \cdot |p(e)|}{|v_i|} \tag{8}$$

where $E(v_i)$ is a set of editors in v_i, $|v_i|$ is the number of letters in v_i, $|p(e)|$ is the number of letters in $p(e)$. This function means that the version quality score is the weighted average value of text quality score, and the weight is the number of letters in the text.



(a) Edit warring (b) Without edit warring

Figure 3: Example of edit history

3.3 Edit War

Our proposed method is resistant to vandalism. By using an example, we explain that the quality scores of editors are not increased or decreased by vandalism.

In this example, we consider two cases: edit history (a) with edit warring and (b) without edit warring. Figure 3 shows the example of an edit history. In case (a), e_s writes 7 letters, and then a vandal e_v deletes all text of e_s. e_s reverts to version 1, but e_v deletes all text of e_s again. e_s reverts to version 1 again. Then, e_a deletes 4 letters, e_b deletes 2 letters, and e_c deletes 1 letter. Case (b) is the same edit history without the vandal e_v. First, e_s writes 7 letters, then e_a, e_b, and e_c delete 4, 2, and 1 letters, respectively.

From these cases, we calculate the text quality score by e_s using equation (1) described in section 3.2.2. In case (a), e_s leaves 7, 0, 7, 0, 7, 3, 1, and 0 letters at from version 1 to 8, but version 1, 3, and 5 are not verified by the other editors. Then, text quality score by e_s is $\log_2(3+1)+\log_2(1+1)+\log_2(0+1)=3$. In case (b), e_s leaves 7, 3, 1, and 0 letters at from version 1 to 4, but version 1 is not verified by the other editors. Then, text quality score by e_s is $\log_2(3+1)+\log_2(1+1)+\log_2(0+1)=3$, which is the same value as case (a). In short, vandals do not affect the quality scores.

In general, the behavior of vandals, such as inappropriately adding and deleting good quality texts, is not permitted by the other editors, so many non-vandal editors try to counter the behavior of vandals. In our proposed system, if the behavior of an editor is permitted by the other editors, the quality score of the editor increase. As a result, vandals do not affect the quality scores of the other editors.

3.4 Convergence of Text Quality

In this section, we discuss whether the text quality is converged or not by mutual evaluation.

PROOF. From equation (4), (5) and (6), we can calculate $u_{k+1}(e)-u_k(e)$ using $u_k(e')-u_{k-1}(e')$ as follows:

$$
\begin{aligned}
&u_{k+1}(e)-u_k(e)\\
&=\frac{\sum^{D(e)}\tau_{k+1}(e)}{|D(e)|}-\frac{\sum^{D(e)}\tau_k(e)}{|D(e)|}\\
&=\frac{1}{|D(e)|}\sum^{D(e)}\left(\sum_{p(e)\in\bar{P}}\log_2\left(|p(e)|+1+\alpha\sum_{e'\in E'(p(e))}h_k(e,e')\right)\right.\\
&\quad\left.-\sum_{p(e)\in\bar{P}}\log_2\left(|p(e)|+1+\alpha\sum_{e'\in E'(p(e))}h_{k-1}(e,e')\right)\right)\\
&\propto\frac{1}{|D(e)|}\sum^{D(e)}\left(\sum_{p(e)\in\bar{P}}\left(|p(e)|+1+\alpha\sum_{e'\in E'(p(e))}h_k(e,e')\right)\right.\\
&\quad\left.-\sum_{p(e)\in\bar{P}}\left(|p(e)|+1+\alpha\sum_{e'\in E'(p(e))}h_{k-1}(e,e')\right)\right)\\
&=\frac{\alpha}{|D(e)|}\sum^{D(e)}\sum_{e'\in E'(p(e))}\big(h_k(e,e')-h_{k-1}(e,e')\big)\\
&=\frac{\alpha}{|D(e)|}\sum^{D(e)}\sum_{e'\in E'(p(e))}|\delta(e')|\frac{u_{k-1}(e')-u_k(e')}{\max_{e'\in E}u_k(e')-\min_{e'\in E}u_k(e')}
\end{aligned}
\tag{9}
$$

where e' is an editor who deletes texts submitted by e. From this equation, we can find that when at least one editor of e's satisfy the condition of decrease $|u_{k-1}(e')-u_k(e')|$ and all e's satisfy the condition of not increase $|u_{k-1}(e')-u_k(e')|$, the value of $|u_{k+1}(e)-u_k(e)|$ decreases. Then, if at least one e''s $|u'_{k-1}(e')-u'_k(e')|$ decreases, $|u'_{k+1}(e)-u'_k(e)|$ decreases. This means that if $|u'_{k-1}(e')-u'_k(e')|$ does not increase, $u'_k(e)$ is converged to a constant value.

Next, we check how $u'_k(e)$ behaves at initial iteration. From equation (1), initial editor quality $u'_{-1}(e)$ is set to 1. This means that if we set $k=0$ to equation (4), and $u'_{-1}(e)$ to 1, equation (4) equals to the equation (1). Then the range of $u'_0(e)$ is $0\le u'_0(e)\le 1$. In the same way, the range of $u'_0(e)$ is $0\le u'_1(e)\le 1$. Therefore, $|u'_1(e)-u'_0(e)|\le|u'_0(e)-u'_{-1}(e)|$, then $u'_k(e)$ of all editors does not increase.

From these two establishments, we can prove that $u'_k(e)$ converges to a constant value. □

4. EXPERIMENTAL EVALUATION

To determine the accuracy of the article quality score calculated by our proposed system, we did the experimental evaluation. In this evaluation, we tried to confirm that when we use the editor quality score to calculate the text quality score, the accuracy of the text quality score should improve. However, we cannot identify which text is good quality or not, because the unit of text is too small. Therefore, we evaluate the article quality score, a latest version quality score.

4.1 Experimental Setup

We compared four systems: (*baseline 1*) the system based on the proposed method proposed by Adler et al. with penalty to vandals[2], (*baseline 2*) the system based on the method proposed by Hu et al.[5], (*once*) our proposed system using editor quality scores at once, and (*proposed*) our

[2]TextLongevityWithPenalty described at [1]

proposed system using both converged editor and text quality scores. In *once*, we use all steps except step 6. In *proposed*, we use all seven steps. We created article lists, which were ordered by the quality score of the newest versions of the articles.

We set "featured" and "good" articles as a correct answer set. Featured and good articles are selected by the votes of Wikipedia editors, and are evaluated by "Featured article criteria"[3].

We compared four systems using relative precision ratios. We compared the answer set with the list of articles in ascending order of their quality scores. If articles in the answer set are ranked higher, we will be able to confirm that the system calculates accurate quality scores. The key in this evaluation is the appropriateness of answer sets. In current information system retrieval evaluation, observers create answer sets by judging the relevance of articles. However, judging the quality score of articles is difficult, so we cannot confirm the appropriateness of quality score judgments of articles. Therefore, we put only featured and good articles in the answer set.

We set $\alpha = 0.8$ as a parameter of the equation (5). Before these experiments, we set α from 0.1 to 1 in 0.1 increments and calculate averaging precision ratio as preliminary experiment. In this result, when we set 0.8, we got the highest averaging precision ratio of our proposed system.

We used the Japanese version of Wikipedia edit history dumped on January 25, 2013. From these articles, we removed the articles that do not contain links to Wikipedia articles. We also removed the articles for specific purposes, such as redirect pages, notes and rules of Wikipedia. We referred to Wikipedia statistics[4] to decide this definition. These data include $484,146$ articles and $33,743,341$ versions. The number of editors is $2,554,747$ including not registered editors who are identified by IP addresses, and bots which are listed at a list of bots[5].

4.2 Results and Discussions

In this experiment, we did the evaluation using a relative precision ratio per each recall level. Relative precision ratio P is P_t, the number of correct articles selected by the target system, divided by P_b, the number of correct articles selected by the baseline system. Relative precision ratio P is defined as follows:

$$P = \frac{P_t}{P_b} \tag{10}$$

When P is larger than 1, the target system is more accurate than the baseline system. We set the baseline system as *baseline 1*, *baseline 2* and the target system as *once* and *proposed*. Therefore, when we draw a relative recall-precision graph, we first draw a general interpolated 11-pt recall precision graph [4]. Then, we calculate the relative precision ratio P for each recall level. Finally, we draw a relative precision ratio for each recall level.

As we already mentioned in section 4.1, we set the answer set of articles as featured and good articles from Japanese Wikipedia. Since there were 87 featured articles and 499 good articles, we selected 586 articles for the answer set. Next, we calculated articles' quality scores using our proposed method and the baseline method and listed articles in descending order of quality score values. Finally, we calculated the relative precision ratio for each recall level using the article list and the answer set.

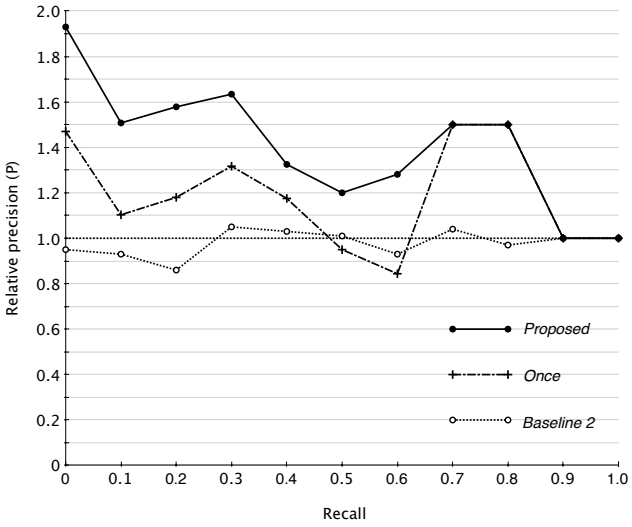


Figure 4: 11-pt relative recall-precision graph

Figure 4 shows a relative precision ratio per each recall level of *once*, *proposed*, and *baseline 2* in comparison with *baseline 1*. From this graph, we discovered that *proposed*, our proposed method calculates article quality scores more accurately than *baseline 1* and *baseline 2*. We also confirmed that when we use editor quality scores for multiple times until convergence, the relative precision ratio increases.

In this experiment, we also confirmed that both text and editor quality scores were converged when we calculate quality scores 18 times. We did not observe diverged and oscillate values. However, if we use the other language version of Wikipedia as a dataset, these quality scores may diverge or oscillate. This is because, when linked graph of editors and texts are separated to multiple graphs, the values may not converge. In our experiment, we do not face this problem. However, if we face this problem, we should develop a method to integrate multiple graphs into a single graph.

In the details of experimental results, we found that our proposed method is effective if vandals attack the articles and cause an edit war, which involves many inappropriate additions and deletions. In the results of *proposed*, there are 36% of articles attacked by vandals in the top 100 positions, whereas there are 5% of articles in *baseline 1*, 0% of articles in *baseline 2*, and 28% of articles in *once*. When we count articles attacked by vandals, we use list of Wikipedia: most vandalized pages". When edit wars happen, vandals delete texts even if they are good quality. Generally, vandals do not indiscriminately delete texts; they delete the texts of specific editors whose opinions they oppose. The articles about religion and politics especially face this kind of edit warring. As a result, vandals decrease quality scores of texts by good quality editors. Using our proposed method, the quality scores of texts by good quality editors increase, and the quality scores of versions that face vandalism increase.

[3] `http://en.wikipedia.org/wiki/Wikipedia:Featured_article_criteria`
[4] `http://en.wikipedia.org/wiki/Wikipedia:What_is_an_article`
[5] `http://ja.wikipedia.org/wiki/WP:BOTST`

On the other hand, the quality scores of versions that do not face vandalism neither increase nor decrease. This is not a problem for our proposed system, because when the articles do not face vandalism, the system can calculate appropriate quality scores. This means that our proposed method is effective for the articles that face vandalism.

From Figure 4, we found that the accuracy of *baseline 1* and *baseline 2* is lower than *once* and *proposed*. This is because, in the list of *baseline 1* and *baseline 2*, we discover that if articles are deleted by vandals, the articles are ranked lower even if the articles are high quality score. In *baseline 1*, as written in the Section 3.7 of [1], if an editor is decided as a vandal, the editor cannot affect positive ratings to the other editors. However, if a high quality text is deleted by vandals, the text is never evaluated by the other editors unless the text is reverted, then the text is treated as low quality score. In our method, if a high quality text is completely deleted by vandals, the system treats that the text is partially remain, then the text is treated as high quality score. As a result, if there are vandals who deletes many good quality articles, our proposed system can calculate appropriate text quality scores.

At recall level from 0.5 to 0.6, relative precision ratio of *once* is lower than 1, which means that the accuracy of *once* is lower than *baseline 1*. This is because of the editors who edit a small number of articles. Even editors who have high quality scores do not always submit good quality texts. Therefore, if there is a good quality text that has survived beyond multiple edits, but the editor's quality score is low, the text is considered low quality score by *once*. However, generally the editor's low quality score is caused by vandalism. Therefore, when we calculate editor and text quality scores, we can recover this problem, so the relative precision ratio of *once* is improved and is higher than 1.

At a recall level from 0.7 to 0.8, relative precision ratios of *baseline 2*, *once* and *proposed* are almost the same, because *baseline 1* cannot find good quality articles at this level. *baseline 1* can find good quality articles when the articles have edit histories long enough for the text quality scores to be calculated. On the other hand, both *once* and *proposed* can find good quality articles with short edit histories, because these systems calculate quality scores of articles using quality scores of editors. Editors generally edit multiple articles, so if a good quality article has a short edit history and if editors obtain high quality scores from the other articles, *baseline 1* and *baseline 2* calculates a low quality score for the article whereas both *once* and *proposed* calculate a high quality score.

5. CONCLUSION

In this paper, we introduced a combination of a survival ratio method and a link analysis method. There are many vandals in Wikipedia, and many vandals attack Wikipedia by deleting good quality texts. In our method, editor quality scores affect deleted text quality scores instead of using an unchanged text survival ratio. Therefore, when the vandals delete good quality texts, they do not affect the survival ratio of the texts, because the editor quality scores of the vandals are low value. As a result, the text quality scores which are attacked by vandals do not decrease. Using this method, we can calculate accurate text quality scores using editor quality scores.

Our proposed approach's strongest point is the resistance to vandalism. In this experiment, 36% of all good quality articles attacked by vandals are identified as good quality articles using our proposed method, but the baseline system identifies all good quality articles as poor quality. From these results, we confirmed that our proposed system could calculate accurate quality score using editor quality scores.

Quality of information is becoming increasingly important in information retrieval research field. An information retrieval system retrieves the documents that are relevant to the user's query, but the system is not concerned about whether the documents are good quality or not. However, if the retrieved documents are poor quality, they should not be retrieved even if they are relevant. Therefore, as Toms et al. [6] already mentioned, when an information retrieval system and a document quality measurement system are integrated, we will develop an information retrieval system more accurate than current information retrieval systems.

6. ACKNOWLEDGMENTS

This research is partly supported by KAKENHI (23700113).

7. REFERENCES

[1] Adler, B., Chatterjee, K., de Alfaro, L., Faella, M., Pye, I., and Raman, V. Measuring Author Contributions to the Wikipedia. In *Proceedings of the 2008 International Symposium on Wikis (WikiSym '08)* (2008).

[2] Adler, B., and de Alfaro, L. A content-driven reputation system for the Wikipedia. In *Proceedings of the 16th international conference on World Wide Web (WWW '07)* (2007), pp. 261–270.

[3] Adler, B. T., Chatterjee, K., de Alfaro, L., Faella, M., Pye, I., and Raman, V. Assigning Trust to Wikipedia Content. In *Proceedings of the International Symposium on Wikis (WikiSym '08)* (2008), ACM.

[4] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval: the concepts and technology behind search*. Addison-Wesley, 2011.

[5] Hu, M., Lim, E., Sun, A., Lauw, H. W., and Vuong, B. Measuring Article Quality in Wikipedia: Models and Evaluation. In *Proceedings of ACM International Conference on Information and Knowledge Management (CIKM 2007)* (2007), pp. 243–252.

[6] Toms, E. G., Mackenzie, T., Jordan, C., and Hall, S. wikiSearch: enabling interactivity in search. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009)* (2009), p. 843.

[7] Wilkinson, D. M., and Huberman, B. A. Cooperation and quality in wikipedia. In *Proceedings of the 2007 international symposium on Wikis (WikiSym '07)* (2007), ACM, pp. 157–164.