Assessing the Quality of Wikipedia Editors through Crowdsourcing

Yu Suzuki and Satoshi Nakamura
Nara Institute of Science and Technology
8916-5 Takayama, Ikoma, Nara 6300192, Japan
{ysuzuki, s-nakamura}@is.naist.jp

ABSTRACT

In this paper, we propose a method for assessing the quality of Wikipedia editors. By effectively determining whether the text meaning persists over time, we can determine the actual contribution by editors. This is used in this paper to detect vandal. However, the meaning of text does not always change if a term in the text is added or removed. Therefore, we cannot capture the changes of text meaning automatically, so we cannot detect whether the meaning of text survives or not. To solve this problem, we use crowdsourcing to manually detect changes of text meaning. In our experiment, we confirmed that our proposed method improves the accuracy of detecting vandals by about 5%.

Keywords

Wikipedia, quality, crowdsourcing, vandalism

1. INTRODUCTION

Wikipedia[1] is one of the most successful encyclopedias on the Internet. Unlike strictly controlled Web-based encyclopedias such as Nupedia[2] or Citizendium[3], anyone can freely edit any article and these edits are immediately reflected in the final version of the articles. Many benign editors submit good-quality articles, but many vandals attempt to damage articles. These vandals are identified by readers and administrators, and then are tagged as "blocked users". As a result, these vandals are prohibited from editing any Wikipedia article. As of September 15, 2015, there were about eleven thousand active editors, [4] including about two thousand blocked editors. Therefore, the ability to assess the quality of Wikipedia editors has become very important[8].

In this paper, we propose a Wikipedia editor-quality assessment method. Here, we define *quality of editors* as an approval rate for texts contributed by Wikipedia editors. When an editor adds a text and many users approve of the text, the editor is assessed as high quality.

Methods based on peer review is the major approach[10], [3],[4] used to detect vandals. In these methods, the quality of an editor is calculated using the edit histories of articles. We assume that low-quality text will be quickly deleted by other editors, whereas high-quality text will remain unchanged for a long time.

Many peer-review methods, however, do not consider the meaning of the text. For example, if the sentence "Wikipedia has good quality articles." is changed to "Wikipedia does not have good quality articles.", the meaning is completely changed, but if the former sentence is changed to "Wikipedia has fine quality articles.", the meaning is not changed. In both cases, several terms are added and deleted, and we cannot decide whether the meaning is actually changed by only considering the quantity of terms changed. Proposed methods based on peer review that rely on systems capturing the addition and deletion of terms are therefore limited.

Automatic detection of changes in sentence meaning is hard, but humans can easily detect these changes. We believe that crowdsourcing techniques can be used to detect changes in sentence meanings that cannot be captured by current natural language processing techniques. This approach should enable us to accurately capture the purpose of edits, and thus improve the accuracy of quality assessment.

In this paper, we therefore propose a method for improving the accuracy of quality assessment for Wikipedia editors. The contributions of this paper are:

- We use crowdsourcing to manually detect changes of text meaning.
- We calculate the quality of Wikipedia editors using the survival time of the text meaning.

2. RELATED WORK

Much research has been done on implicit features regarding user decisions which a system can predict from a user's behavior. When a system uses these features, users do not need to input an evaluation of items. Our proposed method uses this approach. However, how can a user's evaluations be predicted from their behavior?

Adler et al. [1, 2, 3] and Wilkinson et al. [11] propose a method for calculating quality values from edit histories.

[1] https://www.wikipedia.org
[2] http://nupedia.wikia.com/wiki/Category:Nupedia (revived pages)
[3] http://en.citizendium.org/wiki/
[4] https://en.wikipedia.org/wiki/Wikipedia:Size_of_Wikipedia

WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2891113 .

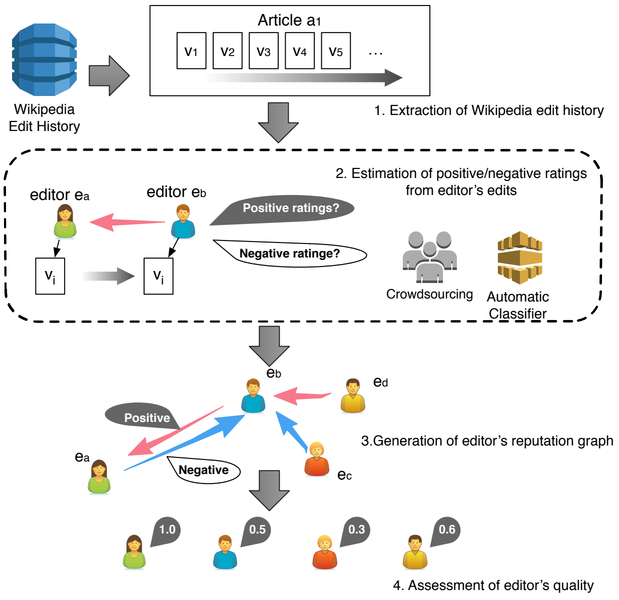


Figure 1: Proposed method

This method is based on the survival ratios of texts. Hu et al. [6] also propose a method for calculating article quality using editor quality, which is similar to our proposed method. This method focuses on unchanged content, and they assume that that an editor considers a text to be good text if the editor does not change that text. However, this method does not consider the original editors. Therefore, for an article which has only one version – i.e., the text of the article has not been edited by other editors – we cannot calculate text quality values using existing methods. In our method, we do consider editors. Therefore, if the editor of a new text edits other texts, and these edited texts are left unchanged or deleted by other editors, we can calculate the quality of the new text.

In these research, edit distance is generally used to detect the differences of two versions. However, if the positions of sentences are changed, or if two sentences are merged into one sentence, edit distance cannot capture actual difference. WikiWho [5] is proposed to solve this issue. However, this method does not always detect reverted texts. Moreover, if the terms in the sentence are dramatically changed but the meaning of the sentences are the same, WikiWho treat these two sentences as different sentences. In our method, we use crowdsourcing to solve this problem.

3. PROPOSED METHOD

Our proposed method consists of the following steps (Figure. 1):

Step 1. Extract all versions of articles in a Wikipedia edit history file

Step 2. Estimate positive/negative ratings from an editor's edits

Step 3. Generate a reputation graph for an editor

Step 4. Assess the quality of the editor

Step 2 is described in more detail below (Figure. 2):

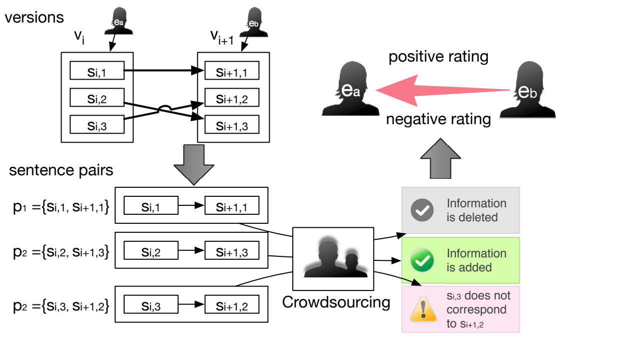


Figure 2: Estimating an editor's peer review (details of Step 2)

1. Extract the text difference between two versions from the edit history
2. Estimate each editor's rating based on the text differences
3. Improve the identification of each editor's rating by other editors using crowdsourcing
4. Predict the ratings of each editor's edits provided by other editors

In this section, we explain these four steps. In particular, we explain Step 2 in detail in Section 3.2.

3.1 Extraction of text differences

First, we input an edit history of Wikipedia articles to our proposed system. In the edit history, all versions of articles are recorded. Each version includes a snapshot of text, a name of an editor, and a timestamp from when the version is created. Here, we define that an article a has a series of versions $V = \{v_1, v_2, \cdots, v_N\}$, where v_i is the i-th edited version.

We then extract which part of the text is edited between an old version and a new version. In our system, we extract a set of sentence pairs $P = \{p_1, p_2, \cdots, p_M\}$, where p_t is a sentence pair which includes two sentences s_j^{old} and s_j^{new}. s_j^{old} is a sentence in an old version, and s_j^{new} is a sentence which may correspond to s_j^{old} in a new version.

To generate sentence pairs P from the series of versions A, we first make all combinations of the versions, and generate version pairs $V = \{(v_1, v_2), (v_1, v_3), \cdots, (v_{N-1}, v_N)\}$. To reduce the calculation time, we generate pairs (v_i, v_j), where $0 < i - j \leq \alpha$. We then split the text of v_i and v_j into sentences using a period as a delimiter. As a result, we get a list of sentence $v_i = \{s_{i,1}, s_{i,2}, \cdots, s_{i,l(v_i)}\}$ and $v_j = \{s_{j,1}, s_{j,2}, \cdots, s_{j,l(v_j)}\}$, where $l(v_i)$ is the number of sentences in v_i. In this process, we remove Wiki-style symbols like "[" and "{" from the sentences. Moreover, we remove sentences where the ratios of symbols and numbers are more than 50% of the sentence, because these sentences are typically parts of tables.

Next, we calculate which sentences in an old version correspond to sentences in a new version. We use a vector space model to measure the similarity of sentences. We divide sentences of v_i and v_j into terms using word segmentation tools, such as POS taggers or morphological analysis tools. We then represent the sentence $s_{i,k}$ as a term vector $\mathbf{t}(s_{i,k})$ as follows:

$$\mathbf{t}(s_{i,k}) = [f(t_1, s_{i,k}), f(t_2, s_{i,k}), \cdots f(t_K, s_{i,k})] \qquad (1)$$

where t_i is a separate term, and $f(t_i, s_{i,k})$ is a tf/idf value of t_i in $s_{i,k}$. When we calculate an idf value of t_i, we use an article as a document unit. Therefore, if a term occurs multiple times in one article, we set the document frequency of the term to 1. Using cosine similarity as $sim(s_{i,k}, s_{j,m}) = \frac{s_{i,k} \cdot s_{j,m}}{|s_{i,k}||s_{j,m}|}$, we find the sentence $s_{j,m}$ which is the most similar to $s_{i,k}$ in v_j. If $sim(s_{i,k}, s_{j,m}) = 1$, $s_{i,k}$ and $s_{j,m}$ are the same, so we should add the sentence pair of $s_{i,k}$ and $s_{j,m}$ to P. If $sim(s_{i,k}, s_{j,m})$ is not 1, but exceeds the threshold β, the sentence should be categorized as partially changed. We then put the pair of sentences $p_t = \{s_{i,k}, s_{j,m}\}$ into P. If $sim(s_{i,k}, s_{j,m})$ is lower than β, $s_{i,k}$ does not correspond to $s_{j,m}$, so we do not add this sentence pair.

3.2 Assignment of Six Types of Label

Next, we assign six labels – "EQUAL," "ADD," "DELETE", "ADD+DELETE", "NO CORRESPONDENCE", and "NOT MAKE SENSE" – to sentence pairs in $p_t = \{s_j^{old}, s_j^{new}\} \in P$. "EQUAL" means that s_j^{old} and s_j^{new} have the same meaning. If s_j^{old} and s_j^{new} are written using different terms and different grammatical structures yet have the same meaning, the label should be "EQUAL." "ADD" means that a new sentence contains all of the information from an old sentence and some added information. "DELETE" means that a new sentence contains only part of the information from an old sentence. "ADD+DELETE" means that a new sentence contains part of the information from an old sentence and adds some information. We assign this label if an old sentence is partially changed, but the old and new sentences have some of the same information. "NO CORRESPONDENCE" means that an old sentence and a new sentence have completely different meanings. "NOT MAKE SENSE" means that either an old sentence or a new sentence does not make sense.

We assign these six labels to the sentence pairs in P. As we stated in the Introduction, this task is difficult to process automatically for all sentence pairs. However, it would be expensive to process this task for all sentence pairs by crowdsourcing because there are many sentence pairs. To reduce the cost and increase the accuracy of assigning labels, we categorize the sentence pairs into two groups: sentence pairs which should be processed by crowdsourcing, and sentence pairs which should be processed by the huristic rules.

When we browse the sentence pairs, it is difficult to label pairs by the huristic rules if an editor both adds and deletes terms to and from an old sentence to make a new sentence. Labeling is easier if the edits between an old sentence and a new sentence are only additions or deletions of terms, but not both. Therefore, we categorize a set of sentence pairs P into two groups P_m and P_a, where P_m is a set of sentence pairs containing edits of both addition and deletion, and P_a is a set of the other sentence pairs.

3.2.1 Labeling of Edits through Crowdsourcing

The goal of this task is to assign one of the six labels to each sentence pair in P_m. To accomplish this, we have constructed a web-based system for crowdsourcing that provides the sentence pairs in P_m to crowdsourcing workers and then aggregates the responses of the workers.

First, the system provides a sentence pair and the following two questions to the workers via a Web interface (Figure 3):

Q1: From the old sentence to the new sentence, how has the content been modified? (Multiple-answer question)

1. The new sentence has more information than the old sentence.
2. The old sentence has more information than the new sentence.
3. The meanings of the old and new sentences are slightly different.
4. The old sentence does not correspond to the new sentence.
5. The old sentence does not make sense.
6. The new sentence does not make sense.

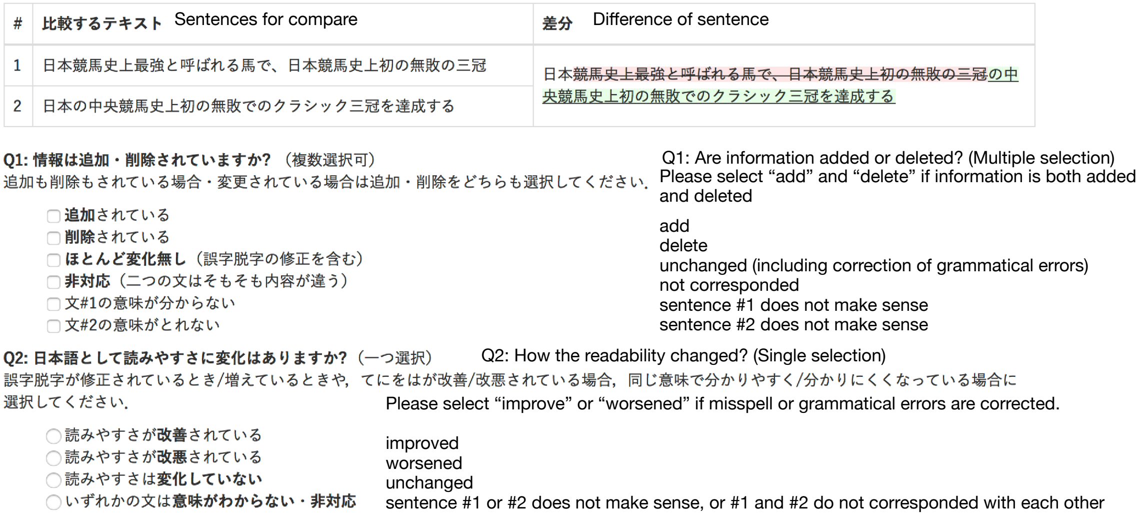
When some information is deleted and other information is added, we expect that the workers will choose both (1) and (2). When both the old and new sentences do not make sense, the workers should choose both (5) and (6).

Q2: From the old sentence to the new sentence, how has the readability been modified? (Single-answer question)

1. Improved. The editor has corrected some misspelled words or grammatical errors in the old sentence.
2. Unchanged.
3. Worsened. The editor has created some misspelled words or grammatical errors in the old sentence.
4. The old sentence does not correspond to the new sentence, or the old sentence or new sentence does not make sense.

Q1 is about the modification of sentence meanings, and Q2 is about the modification of vocabulary and grammatical errors. We use these two questions because we have to deal in different ways with two kinds of modification – the modification of content and that of readability. If we use a question like, "How has the old sentence been changed?", the workers may not distinguish these two types of modification. We only observe the differences in sentence meanings, not the differences in readability. Therefore, although we ask Q2, the question about readability, the Q2 responses are ignored.

We set the condition that at least two workers must assign labels for each sentence pair. If more than half of the workers select the same options for a sentence pair, we assign labels to the sentence pair using the rules described at Table 1. However, if workers select different options from each other, we add workers. If more than 10 workers are assigned to one sentence pair, and no option is selected by more than half of the workers, we assign the label "NO CORRESPONDENCE" to the sentence pair.



#	比較するテキスト Sentences for compare	差分 Difference of sentence
1	日本競馬史上最強と呼ばれる馬で、日本競馬史上初の無敗の三冠	日本~~競馬史上最強と呼ばれる馬で、日本競馬史上初の無敗の三冠~~の中央競馬史上初の無敗でのクラシック三冠を達成する
2	日本の中央競馬史上初の無敗でのクラシック三冠を達成する	

Q1: 情報は追加・削除されていますか?（複数選択可）
追加も削除もされている場合・変更されている場合は追加・削除をどちらも選択してください.

Q1: Are information added or deleted? (Multiple selection)
Please select “add” and “delete” if information is both added and deleted

□ **追加**されている
□ **削除**されている
□ **ほとんど変化無し**（誤字脱字の修正を含む）
□ **非対応**（二つの文はそもそも内容が違う）
□ 文#1の意味が分からない
□ 文#2の意味がとれない

add
delete
unchanged (including correction of grammatical errors)
not corresponded
sentence #1 does not make sense
sentence #2 does not make sense

Q2: 日本語として読みやすさに変化はありますか?（一つ選択）
誤字脱字が修正されているとき/増えているときや，てにをはが改善/改悪されている場合，同じ意味で分かりやすく/分かりにくくなっている場合に選択してください.

Q2: How the readability changed? (Single selection)
Please select “improve” or “worsened” if misspell or grammatical errors are corrected.

○ 読みやすさが**改善**されている
○ 読みやすさが**改悪**されている
○ 読みやすさは**変化していない**
○ いずれかの文は**意味がわからない・非対応**

improved
worsened
unchanged
sentence #1 or #2 does not make sense, or #1 and #2 do not corresponded with each other

Figure 3: A system interface for crowdsourcing workers

3.2.2 *Automatic Labeling of Edits*

The goal of this task is to assign labels to sentence pairs in P_a. In P_a, there are three types of sentence pair: 1) all terms in an old sentence are included in a new sentence and terms are added in the new sentence, 2) all terms in a new sentence are included in an old sentence and some terms from the old sentence are deleted in the new sentence, and 3) an old sentence and a new sentence are the same. We automatically assign the “ADD” label to sentence pairs of type 1), the “DELETE” label to those of type 2), and the “EQUAL” label to those of type 3).

3.3 Editor’s Reputation

From the sentence pairs with labels assigned, we set the ratings of editors. We assume that editor $e_a \in E$ gives positive ratings to $e_b \in E$ if a text of e_a is not deleted by e_b, and e_a gives negative ratings to e_b if a text of e_a is deleted by e_b. Using this assumption, we assign editor’s ratings by aggregating the labels of sentence pairs.

First, we set $r_p(p(s_i, s_j))$ as follows:

$$r_p(p(s_i, s_j)) = \begin{cases} 1 & \text{if } n(s_i) \supset n(s_j) \\ 0 & else \end{cases} \tag{2}$$

where $r_p(p(s_i, s_j)) = 1$ means that if s_i is changed to s_j, part of the information in s_i is deleted. $n(s_i)$ and $n(s_j)$ are the amounts of information in s_i and s_j, respectively. Therefore, if a sentence pair is assigned a label of “DELETE” or “ADD+DELETE” by the manual or automatic labeling described in section 3.2, $r_p(p(s_i, s_j))$ is set to 1. Otherwise, $r_p(p(s_i, s_j))$ is set to 0.

Selection of Q1	label
1. and 2.	ADD+DELETE
1.	ADD
2.	DELETE
3.	EQUAL
4.	NO CORRESPONDENCE
5. and 6.	NOT MAKE SENSE
5.	NOT MAKE SENSE
6.	NOT MAKE SENSE

Table 1: Rules for assessing labels

Next, we define $r(e_a \to e_b, v_i)$, a rating from e_a to e_b at version v_i, as follows:

$$r(e_a \to e_b, v_i) = \begin{cases} 1 & \text{if } e_a \text{ deletes } e_b\text{'s information at } v_i \\ 0 & \text{else} \end{cases} \tag{3}$$

where $r(e_a \to e_b, v_i) = 1$ means that e_a deletes e_b’s information in version v_i more than once. To calculate this equation, we collect sentence pairs where the old sentence is edited by e_a and the new sentence is edited by e_b.

3.4 Assessment of an Editor’s Quality

Finally, we calculate a quality score $q(e_a)$ for editor e_a as follows:

$$q(e_a) = 1 - \left(\frac{\sum_{v_i \in V} \sum_{e_k \in E} r(e_k \to e_a, v_i)}{|E(e_a)|} \right) \tag{4}$$

where $|E(e_a)|$ is the number of all sentence pairs with e_a as an editor of the edited sentence. If e_a adds a version v_i and the added information is deleted by other editors, the value of $r(e_k \to e_a, v_i)$ increases; thus, the value of $q(e_a)$ decreases.

4. EXPERIMENTAL EVALUATION

We experimentally evaluated the accuracy of assessing editor quality through the method presented in this paper. In our experiment, we measured how our method can extract low quality editors, and calculate recall and precision ratio.

We used a *baseline* method as our proposed method without using crowdsourcing. In the baseline method, when we processed Step 2 described in section 3.2.1, we used only automatically labeled sentence pairs to P_m, and we did not use the labels by crowdsourcing. We assigned a label "DELETE" to sentence pairs if more than one term was deleted. Otherwise, we assigned a label "NOT_DELETE" to sentence pairs. In section 3.3, we only considered sentence pairs labeled "DELETE" or "ADD+DELETE"; we did not use sentence pairs labeled "NOT_DELETE."

4.1 Experimental Setup

We used the edit history data of Japanese Wikipedia as of May 12, 2015. In this data, there were 1,523,561 articles, 3,016,675 editors, and 45,207,644 versions. If we assessed quality values for all editors, though, the crowdsourcing cost would be too high. Therefore, we selected target articles from four categories: "Sports," "Islam," "Bird," and "Hawaii". The articles in these categories are maintained by active user groups, so we expected the articles to be well maintained. The numbers of articles and editors are shown in Table 2.

To calculate the recall and precision ratio, we need to prepare a correct answer set. As far as we know, there is no good-quality editor list for Wikipedia. If we were to manually create a list of good-quality editors, it would be very hard for us to create appropriate, unbiased editor sets. Therefore, we instead used the blocked user list provided by Wikipedia[5] to identify low-quality users. As shown in Table 2, the number of blocked editors for the target articles we used was $1,601$.

In the step described in section 3.1, we set a threshold $\beta = 0.7$, because this value proved to be the most accurate value when we did our preliminary experiment.

4.2 Crowdsourcing

In our experiment, we used crowdsourcing to assign labels to sentence pairs. There are many crowdsourcing platforms such as Amazon Mechanical Turk[6] and CrowdFlower[7]. We choose to use Crowdworks[8], one of the major crowdsourcing platforms in Japan, because the target articles were written in Japanese and the workers would have to read, understand, and assign labels for sentence pairs written in Japanese.

The crowdsourcing statistics are shown in Table 3. To ensure the accuracy of labeling through crowdsourcing, each sentence pair was evaluated by more than two workers. When two workers for one sentence pair assigned different labels, our system assigned one more worker to the sentence pair. If more than five workers were assigned to one sentence pair, and no label was selected by more than 50% of the workers, the sentence pair was labeled "UNKNOWN."

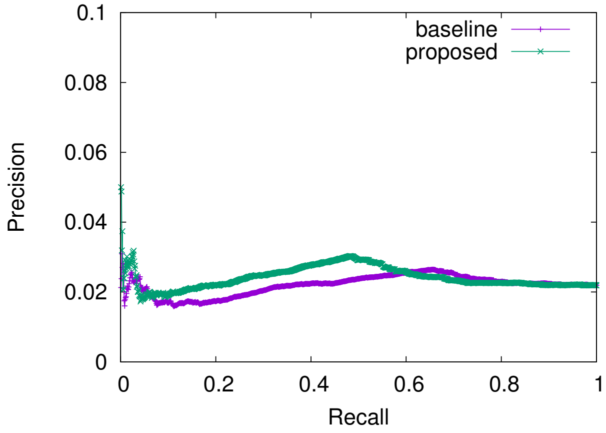


Figure 4: Recall-Precision curve

To collect the evaluation results from workers using crowdsourcing, we constructed a Web-based system using Ruby on Rails 4.2 and Oracle Database Server 12*c*. Using this system, we showed five sentence pairs to each of the workers and then the workers input the evaluation results through the system. When a worker had assessed 100 sentence pairs, we paid 50 JPY (about 0.5 USD) to the worker; however, if a worker assessed fewer than 100 sentence pairs, we did not pay anything. For most workers, one assessment took about 30 seconds per sentence pair. As a result, we paid $15,000$ JPY to collect $24,884$ evaluations. We collected these assessments over a period of three weeks.

4.3 Experimental Results and Discussion

Figure 4 shows a recall-precision curve demonstrating that our proposed method can calculate more accurate editor quality values than the baseline method. At any recall ratio, the precision ratio of our proposed method was about 5% greater than that of the baseline system. At the lowest recall ratio, in particular, the precision ratio of our proposed method was 5% while that of the baseline method was 3%.

However, several sentences were not appropriately labeled by workers. There seem to have been two reasons for this: some workers did not consistently assign appropriate labels, and for several sentence pairs it was very difficult to assign appropriate labels. To solve the first issue, we should measure the accuracy of each worker's assessment.

The second issue is a serious problem. For several articles, readers required some subject knowledge to understand the articles. Therefore, if workers lacked the required knowledge, they were likely to misjudge in their evaluations. Moreover, if changes in the sentences were complex, the

target articles	#
articles	4,412
editors	78,340
blocked editors	1,601
sentence pairs	759,190

Table 2: Experimental Setup.

crowdsourcing	#
workers	227
evaluations	24,884
sentence pairs	8,040
cost	0.5 JPY/evaluation (≈ 0.005 USD)

Table 3: Crowdsourcing Statistics

[5] `https://en.wikipedia.org/wiki/Category:Blocked_Wikipedia_users`

[6] `https://www.mturk.com/mturk/`

[7] `http://www.crowdflower.com`

[8] `https://crowdworks.jp/`

workers could have been confused when selecting options. For example, if a large amount of information is deleted and a small amount is added, a worker may be uncertain as to whether to ignore the small part added. To solve this issue, we should use an unsupervised method, such as a majority vote, because we cannot create an answer set for all cases.

5. CONCLUSION

In this paper, we have proposed a method for assessing the quality of editors through crowdsourcing techniques. In current assessment methods based on the peer review of Wikipedia editors, text that survives multiple edits is assessed as good-quality text. However, these methods do not consider the changes in sentence meanings, because it is difficult to automatically capture these changes. In our method, we use crowdsourcing techniques to solve this problem. As a result, the precision ratio increases by about 5%.

In this work, we only aimed at detecting vandals. Therefore, we only considered the negative ratings of peer reviews. However, we also have positive ratings such as ”ADD” and ”EQUAL” available. By using both positive and negative ratings, we will be able to identify good-quality editors.

Moreover, we should be able to identify many types of vandal, which we cannot do through our current method. With our method, we can identify vandals who delete information from Wikipedia, but we cannot identify vandals who write grammatically bad sentences or who enter many misspelled terms. In **Q2** from section 3.2.1, though, we ask crowdsourcing workers about grammatical errors and misspelled terms. Therefore, we will be able to also identify these vandals and good editors using crowdsourcing techniques.

Regarding our future work, there are many vandals changing the Wikipedia content, and there are also many vandals among crowdsourcing workers. If there are too many bad-quality workers, and these workers assess large quantities of sentence pairs, the assessment accuracy will deteriorate. Methods for assessing the quality of crowdsourcing workers have been developed, such as those of Raykar et al.[9] and Ipeirotis et al. [7]. Applying such techniques should improve the accuracy of our proposed assessment.

In our experiments, we used the Japanese Wikipedia dataset. Therefore, we should confirm this method to the other language versions of Wikipedia datasets.

6. ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 23700113, and NAIST Bigdata Project.

7. REFERENCES

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

[2] B. T. Adler, K. Chatterjee, L. de Alfaro, M. Faella, I. Pye, and V. Raman. Assigning trust to wikipedia content. In *Proceedings of the 4th International Symposium on Wikis*, WikiSym '08, pages 26:1–26:12, New York, NY, USA, 2008. ACM.

[3] B. T. Adler, L. de Alfaro, I. Pye, and V. Raman. Measuring author contributions to the wikipedia. In *Proceedings of the 4th International Symposium on Wikis*, WikiSym '08, pages 15:1–15:10, New York, NY, USA, 2008. ACM.

[4] L. De Alfaro, A. Kulshreshtha, I. Pye, and B. T. Adler. Reputation systems for open collaboration. *Commun. ACM*, 54(8):81–87, Aug. 2011.

[5] F. Flöck and M. Acosta. Wikiwho: Precise and efficient attribution of authorship of revisioned content. In *Proceedings of the 23rd International Conference on World Wide Web*, WWW '14, pages 843–854, New York, NY, USA, 2014. ACM.

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

[7] P. G. Ipeirotis, F. Provost, and J. Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 64–67, New York, NY, USA, 2010. ACM.

[8] S. Kumar, F. Spezzano, and V. Subrahmanian. Vews: A wikipedia vandal early warning system. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 607–616, New York, NY, USA, 2015. ACM.

[9] V. C. Raykar and S. Yu. An entropic score to rank annotators for crowdsourced labeling tasks. In *Proceedings of the 2011 Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, NCVPRIPG '11, pages 29–32, Washington, DC, USA, 2011. IEEE Computer Society.

[10] Y. Suzuki and M. Yoshikawa. Assessing quality score of wikipedia article using mutual evaluation of editors and texts. In *Proceedings of the 22nd ACM international conference on Conference on information and knowledge management*, CIKM '13, pages 1727–1732, New York, NY, USA, 2013. ACM.

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