Electronic Preprint for Journal of Information Processing Vol.21 No.2

Regular Paper

Effects of Implicit Positive Ratings for Quality Assessment of Wikipedia Articles

Yu Suzuki[1,a)]

Received: May 22, 2012, Accepted: December 7, 2012

Abstract: In this paper, we propose a method to identify high-quality Wikipedia articles by using implicit positive ratings. One of the major approaches for assessing Wikipedia articles is a text survival ratio based approach. In this approach, when a text survives beyond multiple edits, the text is assessed as high quality. However, the problem is that many low quality articles are misjudged as high quality, because every editor does not always read the whole article. If there is a low quality text at the bottom of a long article, and the text has not seen by the other editors, then the text survives beyond many edits, and the text is assessed as high quality. To solve this problem, we use a section and a paragraph as a unit instead of a whole page. In our method, if an editor edits an article, the system considers that the editor gives positive ratings to the section or the paragraph that the editor edits. From experimental evaluation, we confirmed that the proposed method could improve the accuracy of quality values for articles.

Keywords: wikipedia, reputation, quality, edit history

1. Introduction

Wikipedia [*1] is one of the most successful and well-known User Generated Content (UGC) websites. It has more and fresher information than existing paper-based encyclopedias, because any user can edit any article. Many experts submit texts, and texts submitted by them should be informative for all who read it. Therefore, as well as being very large, Wikipedia is also very important. However, a dramatic increase in the number of editors causes an increase in the number of low-quality articles. The paper by Kittur et al. [1] (Table 1) showed that about 78.6% of 147,360 articles had not reached "B-class" status [*2]. Therefore, automatic or semi-automatic systems should be developed to identify which part of article is high-quality and which is not.

In this paper, we propose a method to identify high quality texts. Here we define the word "quality" as a degree of excellence. The definition of quality has many aspects such as credibility, expertise, and correctness. Therefore, measuring excellence is a difficult task. To solve this problem, we measure the number of editors who consider the article excellent, which is one of the important aspects of quality. When many editors consider the article excellent, the quality of this article is high, but when a small number of editors consider the article excellent, the quality is low. In the latter case, even if only a small number of readers read the article, and these readers consider the article excellent, we decide that the quality of the article is low. This is because there is minimal evidence to decide whether the quality of the article is high or low.

[1] Information Technology Center, Nagoya University, Nagoya, Aichi 464–8601, Japan
[a)] suzuki@db.itc.nagoya-u.ac.jp
[*1] http://www.wikipedia.org/
[*2] http://en.wikipedia.org/wiki/Template:Grading

If editors find low quality texts, the editors generally reject and delete them. Adler et al. [2] investigated that 79% of bad-quality texts are short lived. This means that if a text survives beyond multiple edits by the other editors, the text should be high quality. Therefore, using the *survival ratio* of texts, the system calculates the quality value of a text.

Example 1: Consider the following example. One editor e_a writes a part $p(e_a)$ of an article. Then, another editor e_b edits another part of this article, but keeps $p(e_a)$ intact. Then we assume e_b remains $p(e_a)$ as it is because s/he judged $p(e_a)$ to be high quality. Next, another editor e_c deletes $p(e_a)$. We assume that e_c judged $p(e_a)$ to be low-quality, hence s/he deleted the text. As a result, the part $p(e_a)$ is confirmed by e_b, but not confirmed by e_c. In this case, the *survival ratio* of $p(e_a)$ is 1 when e_b edits, and 0 when e_c edits. Therefore, the overall survival ratio of $p(e_a)$ is 0.5.

In this method, when a text survives beyond multiple edits, the text is judged as high quality. However, as not every editor reads the whole article, even if there is low-quality text on long articles, the text is treated as high-quality. To solve this problem, we use section and paragraph as a unit instead of whole page. This means that if an editor edits an article, the system treats that the editor gives positive ratings to the section or the paragraph which the editor edits. This is because we believe that if editors edit articles, the editors should read whole sections or paragraphs, and delete low-quality texts.

The rest of this paper is organized as follows. First, in Section 2, we summarize related works about measuring quality of Wikipedia, which use explicit and implicit features. In Section 3, we describe how to measure the quality value of parts, editors, and versions. In Section 4, we discuss the evaluation results. Finally, in Section 5, we close with conclusions and future work.

2. Related Work

Much current research on Web document quality or credibility uses either evidence based techniques [3], feature extraction techniques [4] or link analysis based techniques [5]. One main advantage of these techniques is that they are used mainly for general Web pages, not specific domains. However, a weak point is that these methods cannot treat information about edit history, editors who write a text, and so on. Using these information, we can calculate more accurate quality of texts than existing Web document quality assessment systems.

There has been much research in calculating quality degrees of products, people, and objects [6], [7] using reputation based methods. A key concept for evaluating Wikipedia articles is *the peer review process*. Wikipedia is not thought to have a peer review system because most texts are instantly made and saved, though no one reviews these texts. However, Stvilia et al. [6] mentioned that the open edit system was a kind of peer review system where editors of the system vote on explicit or implicit features of the texts.

In these investigations, many features are extracted from editors' behaviors in many studies, and they can be divided into two types: explicit and implicit. Explicit features are directly input to the system by users; for example, voting. Implicit features are not clearly input to the system by users. Instead, the system presumes the users' decisions from their behaviors. In our system, we use implicit features for calculating quality values. In this section, we describe the studies that have used explicit and implicit features and also describe why we choose to use implicit features.

2.1 Explicit Features

Explicit features are commonly used to evaluate quality of information, products, and objects. For example, Amazon.com [*3], a major online shopping site, has a voting system for users to evaluate products by giving them 1–5 stars. When users want to show how satisfied or not they are with a product they have bought, they give the product 1–5 stars and a review. Then, the system presents the average number of stars along with the reviews. If the other users do not know the quality or the satisfaction of the products, they refer to the number of stars and reviews and decide whether to buy it or not. This system has been implemented by not only online shopping sites but also by many online Web services, such as YouTube and Google, because it is easy to implement and the process of calculating the number of stars is easy and clear.

In this method, the system administrators provide an evaluation form that has voting evaluations and review functions. Kramer [8] implemented the voting system on MediaWiki [*4], and also implemented it at the English version of Wikipedia as the Article Feedback Tool [*5]. Using these systems, users easily understand which articles are high-quality by referencing votes by other users. However, one problem with this system is that not every user always appropriately evaluates or reviews. In fact, according to statistics about the video streaming service YouTube [*6], almost all users who vote give five stars, the highest score, to almost all videos they votes on. Moreover, the survey of the Article Feedback Tool by Wikipedia [*7] shows that 90.9% rates articles as "useful," whereas 78.6% of articles are still rated "start."

The advantages of using explicit features are that the system is easy to implement, and users can directly evaluate quality of articles. The disadvantages are that users rating are subjective, and vandals can easily affect the ratings of articles.

2.2 Implicit Features

Implicit features are those implicitly input by users. When the system uses these features, users do not need to input the evaluation of items. Instead of explicit features, the system presumes users' evaluation of items from their behavior. Our proposed method uses this method. However, how to presume users' evaluations from their behavior?

Lifecycles of texts are used for calculating quality values of texts or articles. Wöhner et al. [9] proposed a credibility value calculation method using the lifecycle of a text. In this method, the quality value and lifecycle of a text have a relationship. Halfaker et al. [10] proposed a method for calculating contribution degrees for editors. In this method, they proposed six assumptions about why editors contribute to Wikipedia. These ideas are appropriate when the articles are frequently edited. However, many articles are not frequently edited, so the lifecycle of a text is different for every article. In addition, when edit warring occur, this method cannot calculate appropriate quality values. Our method can calculate appropriate quality values if articles are not only infrequently edited but also suffering an edit war because we consider editor quality values.

Mutual reinforcement model such as HITS, PageRank, and SALSA is used to calculate quality values of Wikipedia articles. Lim et al. [11], [12] and Suzuki et al. [13] proposed a method to calculate quality values using this model. In these methods, the system generates a graph where the nodes corresponds to the editors, and the edges are correspond to the amount of contribution, and then analyzes the graph for calculating quality values. However, these methods assume that all editors browse all parts of articles, which is not appropriate for long articles. In general, many editors do not always browse a whole article. In our method, we consider a section or a paragraph instead of a whole article to calculate quality values, which is more realistic assumption than the existing approach. Therefore, if we combine this model with our proposed method, we will improve the accuracy of calculating quality values.

Adler et al. [2], [14], [15], Hu et al. [12], and Wilkinson et al. [16] proposed a method for calculating quality values from edit histories. These methods are based on survival ratios of texts and is similar to the basic idea described in Section 3.2. In these methods, edit distance is used for measuring difference between old and new versions. In this case, the impact for text quality by deletion and that by remaining are the same. However, these

[*3] http://www.amazon.com/
[*4] http://www.mediawiki.org/
[*5] http://en.wikipedia.org/wiki/Wikipedia:Article%20Feedback%20Tool
[*6] http://www.youtube.com/
[*7] http://www.mediawiki.org/wiki/Article

impacts of these two operations should be different, because the editors can delete a text only once whereas they can remain a text many times. Therefore, if we treat deletion and remaining texts as equivalent, we should separately calculate the impacts from these operations, and then integrate after normalizing.

3. Proposed Method

Our goal is to assess quality value of articles by mutually evaluating quality values of texts and editors. In this section, we describe a method to calculate quality values of texts by positive ratings, quality values of editors, and quality values of articles. The overview of the process is as follows (**Fig. 1**):

(1) The system extracts edit histories of articles from dumped data, and identify editors of texts.
(2) It calculates positive ratings for texts from edit histories.
(3) It calculates editor quality values by combining positive ratings.
(4) It calculates version quality values using editor quality values.

3.1 Modeling

In this section, we define notations that are used throughout this paper as shown in **Fig. 2**. 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. If editor e in all editors E creates a new article, the system makes two versions, v_0 and v_1, and then the system stores the text of editor e in v_1 which consist of one part $p(e)$. We identify editors using editor names, but anonymous editors have no editor name. In this case, we use the IP address instead of editor name. Then, we define version $v_i = \{p(e) | e \in E\}$ as a set of complete parts that is stored at i-th edit and that consists of a text by $1, 2, \cdots, i$-th editors. $p(e)$ is a part of article by editor e. If e deletes all texts from i-th version, v_i is an empty set.

Editor e creates a set of parts $P(e) = \{p(e)\}$ where $p(e)$ is a part created by e for all articles. If e does not add any texts to any articles, $P(e)$ is an empty set. When editors edit one article by same user more than twice consecutively, the system keeps the last version and deletes the other versions created by the users. Therefore, the editor of a version and that of next version are always different.

The aim of our proposed method is calculating version quality value $T(v_i)$ of version v_i. To accomplish our goal, we calculate a parts quality value $\tau(e)$ of parts by editor e, and an editor quality value $u'(e)$ of editor e.

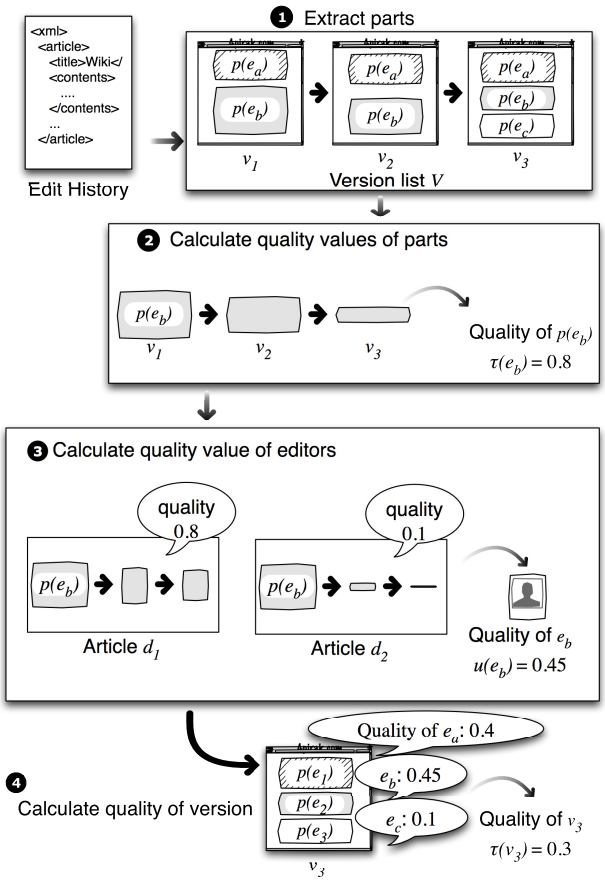

Fig. 1 Overview of our proposed method.

3.2 Key Idea

We show how to calculate quality values of articles using an example of edit history in **Fig. 3**. Using this example, we explain how to calculate quality values of parts $p(e_a)$ that are added by editor e_a in version v_1. First, we identify the texts that are added in version v_1. In this example, the editor e_s writes all texts of v_0, and the editor e_a adds the texts "Ueshima" and "Prime Minister" as $p(e_a)$ to version v_2. At this time, we suppose that e_a gives positive ratings to $p(e_s)$. e_a remains 21 letters written by e_s ("Yoshihiko", "is", "the", "of Japan"), then e_a gives 21 letters of positive ratings to $p(e_s)$. e_a deletes 13 letters written by e_s ("Noda", "president"), then e_a gives 8 letters of positive ratings to $p(e_s)$.

However, the problem of this method is that if e_a edits small edits, e_a writes only a small number of letters and remain all texts,

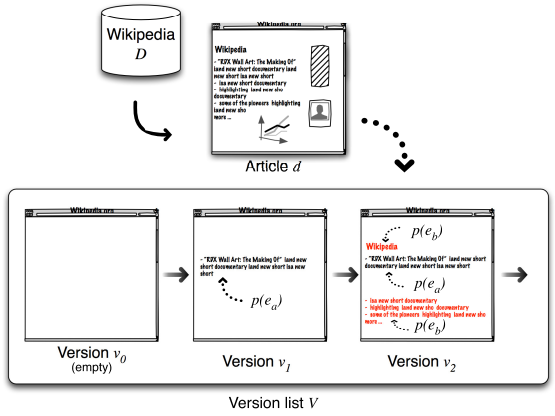

Fig. 2 Notations of this paper.

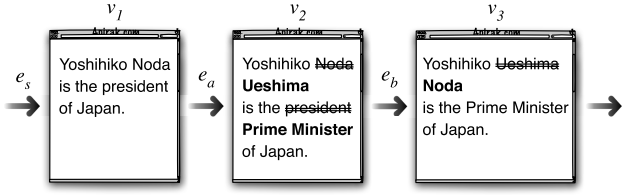

Fig. 3 Example of edit history.

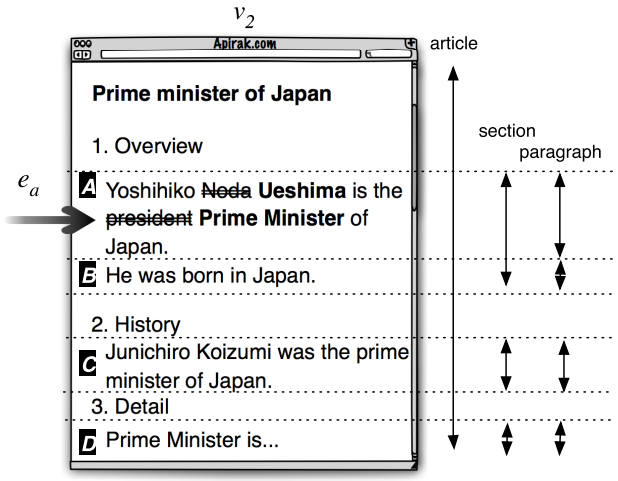

Fig. 4 Implicit Positive Ratings using the whole article, sections, and paragraphs.

the system decides that e_a permits to remain almost all texts. We believe that not all editors read whole articles. Therefore, when there are many editors who do not read whole articles, the accuracy of quality values should decrease. To solve this problem, we use section and paragraph as a reading unit.

3.3 Quality Value

In this section, we describe how to calculate positive ratings from edit history. **Figure 4** shows an example to explain how to calculate positive ratings. First, we define a unit as article, section, or paragraph. A unit of whole article is defined as texts in the whole article. A unit of section is defined as the texts divided by symbols which indicate separation of sections. In this example, A and B belong to the same section, C and D belong to different section. A unit of paragraph is defined as texts divided by special, not linguistic characters, such as HTML tags and line break code. In this example, A and B belong to different paragraph because A and B is divided by line break.

We describe intuitive explanation of positive ratings. In this example, editor e_a edits a part of article A. When we use whole article as a unit, we assume that e_a permits to remain parts A, B, C, and D. Therefore, e_a gives positive ratings to A, B, C, and D. When we use section as a unit, we assume that e_a permits to remain parts A and B. In this case, we suppose that e_a do not read C and D because e_a does not edit. Therefore, e_a gives positive ratings to A and B. When we use paragraph as a unit, we assume that e_a permits to remain only A. In this case, we suppose that e_a does not read B, C, and D. Therefore, e_a gives positive ratings to A.

3.3.1 Positive Ratings

Next, we calculate the quality values of parts using quality values of editors.

3.3.1.1 A Whole Article as a Unit

We calculate the positive ratings using whole article $\tau^a(e)$ as follows:

$$\tau^a(e) = \sum_{p(e)\in P_a} (\log|p(e)| + 1) \tag{1}$$

where P_a is a set of texts which is permit to remain texts written by e and permitted by not e, and $|p(e)|$ is the number of letters in $p(e)$.

3.3.1.2 Section as a Unit

We calculate the positive ratings of parts using section $\tau^s(e)$ by Eq. (1). P_s is used instead of P_a, and is a set of texts which is permit to remain texts written by e and permitted by not e.

3.3.1.3 Paragraph as a Unit

We calculate the positive ratings of parts using paragraph $\tau^p(e)$ by Eq. (1). P_p is used instead of P_a, and is a set of texts which is permit to remain texts written by e and permitted by not e.

3.3.2 Quality Values of Editors

From text quality values, we calculate editor quality values. In this section, we calculate three types of editor quality values, such as the editor quality values based on the positive ratings using whole article $\tau^a(e)$, that using sections $\tau^s(e)$, and that using parts $\tau^p(e)$. We use the same equation for three types of editor quality values, such as $u^a(e)$ for using whole article, $u^s(e)$ for using sections, and $u^a(e)$ for using parts. For simplicity, we write $u(e)$ instead of $u^a(e)$, $u^s(e)$, and $u^p(e)$ using $\tau^a(e)$, $\tau^s(e)$, and $\tau^p(e)$.

$$u(e) = \frac{\sum_{d\in D(e)} \tau(e)}{|D(e)|} \tag{2}$$

where $D(e)$ is a set of all Wikipedia articles that e edits, and $|D(e)|$ is the number of articles in $D(e)$. If we calculate $u(e)$, we remove parts 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 parts to express their opinions and do not always delete them. Therefore, the quality values of these texts tend to be higher than those of general articles.

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

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

where $u'(e)$ is one of three editor quality values, such as $u^a(e)$ using whole article, $u^s(e)$ using sections, and $u^p(e)$ using parts.

3.3.3 Quality Values of Versions

Using $u'(e)$, we define the version quality value $T(v)$ of version v as follows:

$$T(v) = \frac{\sum_{e\in P(e)} u'(e)\cdot|p(e)|}{|v|} \tag{4}$$

where $T(v)$ is one of three version quality values, such as $T^a(v)$ using whole article, $T^s(v)$ using sections, and $T^p(v)$ using parts. $|v|$ is the number of letters in v, and $u'(e)$ is the editor quality value of e. This function means that the version quality value is the weighted averaging ratio of part quality values, and the weight is the number of letters in the parts.

4. Experiments

To determine the accuracy of the quality values calculated by our proposed system, we conducted an experimental evaluation. In this evaluation, we tried to confirm that when we use editor quality values to calculate text quality values, the article quality values are accurate.

In this experiment, we compared 4 systems. *page* is the system using article based ratings, *sec* is the system using section based

ratings, and *par* is the system using paragraph based positive ratings. *baseline* is the baseline system proposed by Adler et al. [2], and this method is based on edit distance.

In this experiment, we compared these systems using recall/precision graph [17]. We compared the answer set with the list of articles in ascending order of their quality values. If articles in the answer set are ranked higher, we will be able to confirm that the system calculates accurate quality values. The key in this evaluation is the appropriateness of answer sets. In current information system retrieval evaluation, observers create answer sets by judging relevance of articles. However, judging the quality of articles is difficult, so we cannot confirm the appropriateness of quality judgments of articles. Therefore, we put featured and good articles selected by Wikipedia users as the correct answer set.

4.1 Data Sets

In this experiment, we used the Japanese version of Wikipedia edit history dumped on Mar. 15, 2012, which can be downloaded at the Wikipedia dump download site [*8]. We selected 447,243 articles and 27,754,724 versions which are edited by more than 10 editors, and which are not empty at the last version. The number of editors is 2,647,419 including not registered editors who are identified by IP addresses, and bots which are listed [*9]. When we select articles, we referred to Wikipedia statistics [*10] to decide which articles we select. We do not select the articles that do not contain at least one link to Wikipedia articles. We also do not select the articles for specific purposes, such as redirect pages, notes, and rules of Wikipedia.

In this experiment, we set the answer set of "featured" and "good" articles as a correct answer set. Featured and good articles are selected by the votes of Wikipedia users (mainly readers). These articles are evaluated by "Featured article criteria" [*11] and peer reviewed by many active users. Therefore, we believe that these articles are high quality, so we could use featured and good articles as high quality articles for the test set. The number of featured and good articles are 72 and 297 respectively, so we use 369 articles as high quality articles.

4.2 Experimental Results and Discussions

Figure 5 shows the recall/precision graph. The meaning of each line is described at the top of Section 4. From this graph, we observe that *par*, the system which uses paragraph based positive ratings calculates article quality values more accurately than the other methods. When we compare the results of *par*, *sec* and *page*, *baseline*, the order of articles dramatically changes, different high-quality articles have high quality values. Therefore, we can calculate the most accurate quality values when we use positive ratings using paragraphs as a unit.

In the detail of Fig. 5, we found that *sec* and *par* attain high precision ratios at recall ratio 0.0 to 0.5, whereas these two systems are low precision ratios at recall ratio 0.6 to 1.0. This means that our method is effective for extracting about a half of featured and good articles. This is because, if a small number of editors edit an article, and the article is short, our methods and baseline system cannot capture the high quality articles.

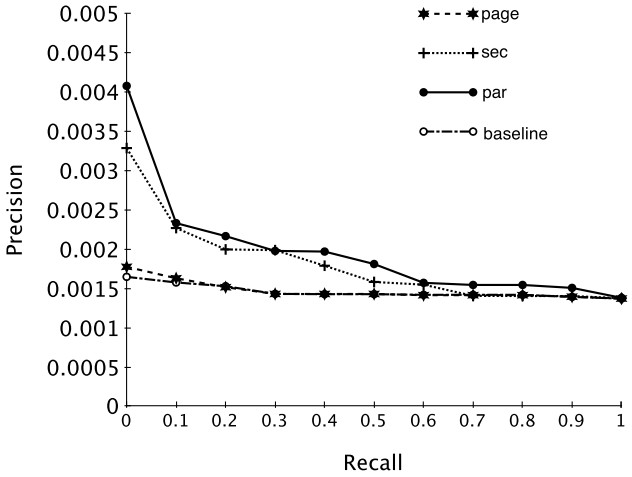

Fig. 5 11-pt interpolated recall/precision graph.

For discussing about this results, we pick up several articles from results as examples. An article "Department stores in Japan" [*12] is a long article, in which several parts are well written but several parts are poorly written, and thus this article is not selected as featured or good article. In this page, many editors did minor edits. Therefore, if we use *baseline* or *page*, many parts are given positive ratings by the editors who edit minor changes. As a result, this article is second-ranked when we use *baseline*, and third-ranked when we use *page*. If we use *par*, this article is 15-th ranked. Therefore, using *par*, we can calculate more appropriate quality values than using *page* and *baseline*.

An article "Foreign policy of the Barack Obama administration" [*13] is selected as a good article, and is very long article. This page is ranked 490th when we use *par*, but is ranked 442,754th when we use *page*. This is because this page is written by a low number of editors, and has a low number of versions. Therefore, if we use *page*, the texts in the article cannot gain positive ratings. However, if we use *par*, the texts in the article can gain positive ratings, because many descriptions of many sections are added but not deleted. In this case, the system *par* can calculate more appropriate quality values than *page* and *baseline*.

An article "Diazepam" [*14] is selected as an excellent article, and is about medicine. This page is a very technical article, a very small number (almost one) of editors edit this article. In this kind of page, almost all Wikipedia editors cannot decide which articles are high quality or not, because the editors rarely know about Diazepam. In this case, all systems output very low ranks to the article. In short, our proposed system cannot appropriately assess this kind of technical articles. Our approach requires improvement to resolve this issue.

[*8] http://dumps.wikimedia.org/jawiki/20120315/
[*9] http://ja.wikipedia.org/wiki/WP:BOTST
[*10] Wikipedia: What is an article?: http://en.wikipedia.org/wiki/Wikipedia:What_is_an_article
[*11] http://en.wikipedia.org/wiki/Wikipedia:Featured_article_criteria
[*12] http://ja.wikipedia.org/wiki/日本の百貨店
[*13] http://ja.wikipedia.org/wiki/バラク・オバマ政権の外交政策
[*14] http://ja.wikipedia.org/wiki/ジアゼパム

5. Conclusion

Wikipedia is the most popular and highest quality encyclopedia to be created by many editors. The information on Wikipedia keeps expanding, but its quality is not proportional to its quantity. In this paper, we assumed that not every editor reads whole articles, and this if there is low-quality text on a long article and that text had not seen by the other editors, the text is incorrectly treated as high-quality. To solve this problem, we used a section or a paragraph as a unit instead of using a whole article. This means that if an editor edits an article, the system treats that the editor gives positive ratings to the texts on the section or the paragraph which the editor edits. This is because, we believe that if editors edit articles, the editors may not read the whole article, but they should read a sections or paragraphs, and delete low-quality texts. From experimental evaluation, we confirmed that our proposed system could calculate accurate quality values if we used paragraph as a unit for positive ratings.

The study about information quality 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 high-quality or not. However, if the retrieved documents are low-quality, they should not be retrieved even if they are relevant. Therefore, as Toms et al. [18] already mentioned, when we combine an information retrieval system with our proposed high-quality article retrieval system, we will develop an information retrieval system more accurate than current information retrieval systems.

Finally, we describe several open problems:

Vagueness of quality value: In this paper, we calculated quality values of editors described at Section 3.3.2. However, this editor quality value is not always distinct because the frequency of editing is different for each editor. We suppose that if an editor rarely edits articles, the editor may just happen to obtain a high quality value, but vagueness of the editor quality value should be high. Therefore, we should develop a method to calculate vagueness of editor quality values that does not depend on editor quality value.

Use of natural language processing techniques: In our proposed method, we do not analyze linguistic structures; we only count the number of letters in texts. A strong point of our proposed system is that it can adapt to different language versions of Wikipedia articles. However, a weak point is that it cannot use important features that come from linguistic features. In our experiment, we found that high-quality articles are always written in formal language. Therefore, we should analyze texts using natural language analysis techniques for calculating the survival ratio of texts.

User interface and visualization: We developed a Web-based user interface. In this user interface, all users use the same Web pages as a result. However, we believe that undemanding and demanding users will want to browse different Web pages [19]. For example, if a text were mostly low quality, the system would determine it to be low quality for demanding users but high quality for undemanding users. Holloway et al. [20] and Otjacques et al. [21] have already discussed the user interface of Wikipedia before. Therefore, we should develop a user interface that is useful for every user.

Combination of explicit and implicit feature: At Section 2.1, we introduced about explicit feature, user ratings. In this study, we do not use explicit feature because mentoring user ratings is difficult. However, if we use implicit feature for mentoring user ratings, the accuracy of quality values should improve. There, we should develop how to combine implicit features with explicit features for improving the accuracy of quality values.

References

[1] Kittur, A. and Kraut, R.E.: Harnessing the wisdom of crowds in wikipedia: Quality through coordination, *Proc. 2008 ACM Conference on Computer Supported Cooperative Work*, *CSCW '08*, pp.37–46, ACM, New York, NY, USA (online), DOI: http://doi.acm.org/10.1145/1460563.1460572 (2008).

[2] Adler, B. and de Alfaro, L.: A content-driven reputation system for the Wikipedia, *Proc. 16th International Conference on World Wide Web* (*WWW '07*), pp. 261–270 (online), DOI: http://doi.acm.org/10.1145/1242572.1242608 (2007).

[3] Bendersky, M., Croft, W.B. and Diao, Y.: Quality-biased ranking of web documents, *Proc. International Conference on Web Search and Data Mining*, (*WSDM '11*), pp.95–104, ACM (online), DOI: http://doi.acm.org/10.1145/1935826.1935849 (2011).

[4] Castillo, C., Mendoza, M. and Poblete, B.: Information Credibility on Twitter, *Proc. International Conference on World Wide Web* (*WWW 2011*), pp.675–684, ACM (online), available from ⟨http://portal.acm.org/citation.cfm?id=1963500⟩ (2011).

[5] Henzinger, M.: Link Analysis in Web Information Retrieval, *IEEE Data Engineering Bulletin*, Vol.23, pp.3–8 (2000).

[6] Stvilia, B., Twidale, M., Smith, L. and Gasser, L.: Information Quality Work Organization in Wikipedia, *J. Am. Soc. Inf. Sci. Technol.*, Vol.59, No.6, pp.983–1001 (online), DOI: 10.1002/asi.v59:6 (2008).

[7] Stvilia, B., Gasser, L., Twidale, M.B. and Smith, L.C.: A Framework for Information Quality Assessment, *J. Am. Soc. Inf. Sci. Technol.*, Vol.58, No.12, pp.1720–1733 (2007).

[8] Kramer, M., Gregorowicz, A. and Iyer, B.: Wiki Trust Metrics based on Phrasal Analysis, *Proc. International Symposium on Wikis* (*WikiSym '08*) (2008).

[9] Wöhner, T. and Peters, R.: Assessing the quality of Wikipedia articles with lifecycle based metrics, *Proc. International Symposium on Wikis and Open Collaboration* (*WikiSym '09*), pp.1–10 (online), DOI: http://doi.acm.org/10.1145/1641309.1641333 (2009).

[10] Halfaker, A., Kittur, A., Kraut, R. and Riedl, J.: A Jury of Your peers: Quality, Experience and Ownership in Wikipedia, *Proc. International Symposium on Wikis and Open Collaboration* (*WikiSym '09*), pp.1–10 (online), DOI: http://doi.acm.org/10.1145/1641309.1641332 (2009).

[11] Lim, E.-P., Vuong, B.-Q., Lauw, H.W. and Sun, A.: Measuring Qualities of Articles Contributed by Online Communities, *Proc. IEEE/WIC/ACM International Conference on Web Intelligence* (*WI '06*), Washington, DC, USA, IEEE Computer Society, pp.81–87 (online), DOI: 10.1109/WI.2006.115 (2006).

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

[13] Suzuki, Y. and Yoshikawa, M.: Mutual Evaluation of Editors and Texts for Assessing Quality of Wikipedia Articles, *Proc. 8th International Symposium on Wikis and Open Collaboration* (*WikiSym 2012*), ACM Press (2012).

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

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

[16] Wilkinson, D.M. and Huberman, B.A.: Cooperation and quality in wikipedia, *Proc. 2007 International Symposium on Wikis* (*WikiSym '07*), pp.157–164, ACM (online), DOI: 10.1145/1296951.1296968 (2007).

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

[18] Toms, E.G., Mackenzie, T., Jordan, C. and Hall, S.: wikiSearch: Enabling interactivity in search, *Proc. International ACM SIGIR Confer-*

ence on Research and Development in Information Retrieval (*SIGIR 2009*), p.843 (2009).
[19] Hearst, M.A.: *Search User Interfaces*, Cambridge University Press (2009).
[20] Holloway, T., Bozicevic, M. and Börner, K.: Analyzing and visualizing the semantic coverage of Wikipedia and its authors, *Complexity*, Vol.12, No.3, pp.30–40 (2007).
[21] Otjacques, B., Cornil, M. and Feltz, F.: Visualizing Cooperative Activities with Ellimaps: The Case of Wikipedia, *Cooperative Design, Visualization, and Engineering* (*CDVE '09*), Lecture Notes in Computer Science, Vol.5738, pp.44–51, Springer (2009).

Yu Suzuki was born in 1977. He received his M.E. and Ph.D. from Nara Institute of Science and Technology in 2001 and 2004 respectively. He became an assistant professor at Ritsumeikan University in 2004, a researcher at Kyoto University in 2009, and an assistant professor at Nagoya University in 2010. His current research interests are Social Web analysis and data mining. He is a member of IPSJ, IEICE, DBSJ, JSAI, IEEE-CS, and ACM.