Wikipedia Reputation System Based on Edit History

Heng Lin[a], Liang Niu[b]

[a]*hl2521@nyu.edu*
[b]*ln932@nyu.edu*

Abstract

We will present a reputation system for the well know Wikipedia website based on the edit history of their articles. In our system, editors will gain reputation score from their edit(contribution) to some particular article, they may also lose their score if they did some bad editting behavior like vandalism[1]. Our model is based on the previous work of Adler and Alfaro [1] , in their model they didn't consider the absolute time for edit survival, we improved their model by adding factors generated from editing frequency calculated using absolute time. We have implemented a system to calculate the reputation score for all authors in an article for Wikipedia, and we also evaluate our model by visualing the reputation score for every word's author and compare it to the evaluation system in Wikipedia.

Keywords: Wikipedia, reputation system, Edit History

1. Introduction

1.1. Basic Concepts in Wikipedia

Wikipedia is a free online encyclopedia that aims to allow anyone to edit any article and create them. Wikipedia is the largest and most popular general reference work on the Internet and is ranked among the ten most popular websites. Wikipedia is owned by the nonprofit Wikimedia Foundation.[2] Wikipedia is basically a set of articles that can be editted by anyone. Anonymous user can also be able to edit any articles without registration, though Wikipedia do provide registration function. Thus it will cause some problems. For example, some articles will be editted by users maliciously. They may want to damage some articles or introduce bias or mistakes into some particular articles for their own interest. This kind of behavior is also called "Vandalism" [3]. Vandalism is very common in Wikipedia, thus there is a terminology "Edit War"[4] to describe the circumstance when serval people are trying to take control of an article. In our model, we tried to build a reputation system for Wikipedia so that we can generate reputation score for every author of an article, by doing that, we can decide to trust high reputation authors more because they have proved that they can do good editting.

1.2. Reputation System

As we said above, the system we want to build is a reputation system for authors. To achieve this goal, what we are doing is to generate a score for every edit. It is easy to check all edit history of a single article in Wikipedia. By fetching the edit history data, we actually get all revision from the very beginning of the article to some particular time point. In our experiment, we used a crawler to fetch such data so we can have latest version of an article. The data in stored in the manner that every edit is a full article. It is stored not the patches between two versions but full version for every edit.

For every edit, or for every revision, our model is to consider the edits around it. By saying some edits is "around" a edit, we are actually saying that all revisions are sorted in time sequence, and some edits are closed to a edit either they are n-neighbors (in particular, we take n equals to 3 or 10) of this edit or they have short time gap with this edit (in our model, we take this time gap as 1hr).

1.3. Applications

Reputation system can play a significant role in building a healthy wiki community, because it promotes the quality of articles in many ways. If editors can view previous editting history and their authors' reputation, they can decide whether to keep the edit or not more easily. If we compare the Wikipedia community, which is a knowledge contributing community, to Github, which is a code contributing community, then those who have high reputation are like programmers who have many stars and followers. High reputation will make an author looks more reliable and help other authors make decisions.
Besides, we can visulize the reputation distribution of an article. through that way, it will be obvious that what kinds of article will attract more high reputation authors and based on that, it will be intriguing to discuss the relationship between articles' topic and their authors. We tried to do that a little bit, even not fully discovered. And another way to use reputation is to reward those high reputation authors to promote their passion. Also reputation score itself is some kind of honor on the Internet.

2. Related Work

The work most related to ours is [1], where Wikipedia revisions are used to evaluate authors' reputation. Our work is mostly based on theirs and then do some change or improvement. Also, to discuss the relationship between reputation and article quality is inspired by Aniket Kittur's work done in 2008 [5], in which they use Amazon's crowdsourcing service called "Amazon Mechanical Turk" to evalute the quality of a Wikipedia article. They found that people tend to give higher score to those articles that have more high reputation authors. This leads us to use quality of an article as a evalutation metrics. To evaluate article's quality, we found that Joshua E. Blumenstock's work [6] is a quite intuitive way. Actually, at the early stage of our thinking, we have considered using word length as an important factor to evaluate edit quality. Zeng's work [7] is one of the most successful work done in 2000s, in which they used dynamic Bayesian Network to evaluate quality or trust for an edit. We took a quick look on it but the Bayesian Network method is not we are looking for, we hope to find a way that use edit history more directly to reflect how good a revision is. In many edit history based systems, like adlar's work [1] and Wohner's work [8], editting distance is used to evaluate difference between two versions. And we got the idea of coloring text as visulization from adler's work [9], in which they assigned text color to compare difference between two revisions. We use coloring in a different way, we assign words colors according to their authors' reputation to see an article's authors' distribution, which will be explained later.

3. Reputation Model

3.1. Notation

We use the similar notation in Adler [1]. The following is the notations we are using:

For some particular article, let's say the Wikipedia entry "Reverse Engineering", there are many versions of it. For all the versions, we sorted them in a time sequence, and thus there are versions from very beginning to latest. We call these versions v_0, v_1, v_2 and so on. And we assume v_0 is empty, which means we divide the creating of an article into two versions, one is v_0, the empty article, one is v_1, the first version with content. In most cases, the creator of an article will give some content to it, so we are actually adding an empty version before the first version. And thus we introduce the concept revision:

3.1.1. Adler's notation

we use r_i means the i_{th} edit of an article, which is used to refer to the change from $v_{i-1} to v_i$.
d edit distance, not like traditional editting distance calculation, we will measure edit distance based on words, somehow like the tool "diff".
a_i is the author of v_i. Because there may be continuous edits done by the same author, we will eliminate this situation by merging a few continuous edit done by same author to one edit, or keep only one with others deleted. One thing that is different from Adler's is that in our improvement, we considered anonymous user(even though they are anonymous, we can get their IP address through Wikipedia's API) as useful users. So this notation has different meaning in Adler's model and our model. Adler exclude anonymous authors' contribution, but we consider them as registered users.
$txt(i, j)$, in which $0 < i \leq j \leq n$. This is the notation to note how many words that are introduced in the edit of v_i are still left in v_j.

If $i = j$, then this means the amount of words that added to this article in r_j. All the amount is measured by number of words.
$d(v_i, v_j)$, in which $0 < i \leq j \leq n$. It is edit distance between v_i and v_j. When $i = j$, of course it equals to 0.

3.1.2. Our own notation

And the following notations is for our improvement of Adler's model. Cause we take absolute time into account, we define that:
$t(r_i)$, is the timestamp of edit $r_i : from r_{i-1} to r_i$.
And also $g(r_i)$, the average edit time gap for some edit r_i. It is computed by the following process:
For some edit r_i, find all the edit histories that happened in $[t(r_i) - 30min, t(r_i) + 30min]$, say there are n edit between this time gap. Then $g(r_i) = \frac{60min}{n}$. This notation is related with the edit frequency around some edit. Exeptions: for first a few edits and last a few edits, we use a different formula: $g(r_i) = \frac{t_l}{n}$ in which t_l is the longest time that we can trace to, for first a few edits, this time is $30min + t(r_i)$, for last a few edits, this time is $30min + (t(r_k) - r(r_i))$, in which r_k is the newest edit.

3.2. Concepts

We introduce some concepts that is useful for our model.
First one is authorship. Authorship is defined as the authorship of a word. Say an article "Reverse Engineering", there are about 800 authors. And we can imagine that different words are writed by different users. There is an algorithm that can show for a particular version of this article, who is the

author of which part. Say an article has 5 words: $w_0 to w_4$, and this algorithm can show the authors for all these 5 words: $a_0 to a_4$. Authorship is an important concept, we will use it in calculating $txt(i, j)$ and also in our improvement. We will also show authorship of an article.

3.3. Adler's Model

In this part we will simply introduce Adler's model of content-driven reputation system in a versioned document and our implementation.

3.3.1. Rule 1

Basically, there are two rules in Adler's model, the first one is based on what they called "Text Survial". It is the idea that if text introduced by r_i is still present in v_j, then this can indicate that the author of v_j agrees that r_i is a good or valuable edit. Thus, they proposed their first rule:

RULE 1 ((reputation update due to text survival)). *When the revision r_j occurs, for all $0 < j < i$ such that $j - i \leq 10$ and $a_j \neq a_i$, we add to the reputation of a_i the amount:*

$$c_{scale} * c_{text} * \frac{txt(i,j)}{txt(i,i)} * (txt(i,i))^{c_{len}} * log(1 + R(a_j, r_j))$$

where $c_{scale} > 0, c_{text} \in [0, 1]$, and $c_{len} \in [0, 1]$ are parameters, and where $R(a_j, r_j)$ is the reputation of a_j at the time r_j is preformed.

3.3.2. Rule 2

The second rule is based on the idea of "Edit Survival", the rule is as follows:

RULE 2 ((reputation update due to edit survival)). *When the revision r_j occurs, for all $0 < i < j$ such that $j - i \leq 3$, we add to the reputation of a_i an amount q determined as follows. If $a_j = a_i$ or $d(v_{i-1}, v_i) = 0$, then $q = 0$; otherwise, q is determined by the following algorithm.*

$$q := \frac{c_{slack} * d(v_{i-1}, v_j) - d(v_i, v_j)}{d(v_{i-1}, v_i)}$$

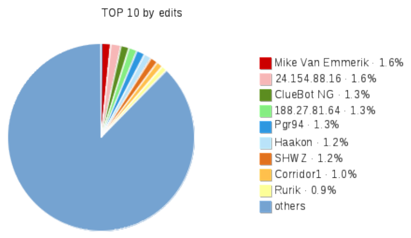
if $q < 0$ *then* $q := q * c_{punish}$ *endif*

$$q := q * c_{scale} * (1 - c_{text}) * (d(v_{i-1}, v_i))^{c_{len}} * log(1 + R(a_j, r_j))$$

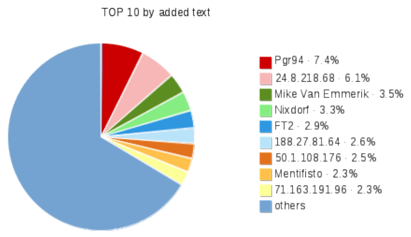
Basic idea behind this rule is that using edit distance to trace how much left for a edit in next 3 edits. The more left, the more reputation should be given to the author.

3.3.3. Implementation

When we are implementing this model, we are using Google's library called "diff_match_patch" to help us do the edit distance calculation, which is based on the algorithm described in Myers' work [10].
We also discovered some details in his model. What he didn't mentioned is that authorship



(a) TOP 10 by edits



(b) TOP 10 by added text

Figure 1: TOP 10 by two metrics of "Reverse Engineering"

3.4. Our Improvement

3.4.1. Take Timestamp into Account

3.4.2. Take Anonymous Users into Account

3.5. Edit War Optimization

4. Authorship

4.1. Authorship as Evaluation

4.2. Authorship and Article Quality

4.3. Conclusion

5. Self Assessment

6. References

[1] B. T. Adler and L. De Alfaro, "A content-driven reputation system for the wikipedia," in *Proceedings of the 16th international conference on*

World Wide Web, pp. 261–270, ACM, 2007.

[2] Wikipedia, "Wikipedia — Wikipedia, the free encyclopedia," 2016. [Online; accessed 16-Dec-2016].

[3] Wikipedia, "Vandalism — Wikipedia, the free encyclopedia," 2016. [Online; accessed 16-Dec-2016].

[4] Wikipedia, "Edit warring — Wikipedia, the free encyclopedia," 2016. [Online; accessed 16-Dec-2016].

[5] A. Kittur, B. Suh, and E. H. Chi, "Can you ever trust a wiki?: impacting perceived trustworthiness in wikipedia," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, pp. 477–480, ACM, 2008.

[6] J. E. Blumenstock, "Size matters: word count as a measure of quality on wikipedia," in *Proceedings of the 17th international conference on World Wide Web*, pp. 1095–1096, ACM, 2008.

[7] H. Zeng, M. A. Alhossaini, L. Ding, R. Fikes, and D. L. McGuinness, "Computing trust from revision history," tech. rep., DTIC Document, 2006.

[8] T. Wöhner and R. Peters, "Assessing the quality of wikipedia articles with lifecycle based metrics," in *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, p. 16, ACM, 2009.

[9] B. T. Adler, K. Chatterjee, L. De Alfaro, M. Faella, I. Pye, and V. Ra-

man, "Assigning trust to wikipedia content," in *Proceedings of the 4th International Symposium on Wikis*, p. 26, ACM, 2008.

[10] E. W. Myers, "Ano (nd) difference algorithm and its variations," *Algorithmica*, vol. 1, no. 1-4, pp. 251–266, 1986.