Copyright Model for Collaboration

-Background Paper-

Jing NING

Jing.ning12@ic.ac.uk

Supervisor: Dr. Chao WU

c.wu09@imperial.ac.uk

Individual Project, Imperial College London

June, 2013

Contents

1. Introduction

Copyright Model for Collaboration is a Msc Individual Project, which is aimed to determine the contribution of each participant for collaboration work done by wiki way.

Online collaboration is increasing in importance on the Web and the Wikipedia in one of the famous examples. It is a fact that the collaborative content does not depend on restricted authors and each article is revised frequently, so it is not easy to consider the reputation given to each user Thus, a model for measuring user contributions to versioned and collaborative information is needed.

This paper is composed of four sections. Following this first introductory part, the second section demonstrates a literature survey on several technical areas related to reputation for wiki. The third section describes the risk and main challenges for this project and the forth section illustrates a proposed schedule of this project.

2. Literature Survey

2.1 Content –Driven Reputation

User-driven reputation system is widely used, and it is based on users rating each other's behaviours or contributions. A best-known example is EBay, where customers and vendors rate each other after transactions. By contrast, the content-driven reputation requires no user input and it is based on their contribution to the content. For example, an author A edits the content of a specific page, and then B revises the same article and he can choose insertions, deletions, displacements and replacements that article. In this example, B chooses to retain what A has done. So a content-driven reputation system would increase the reputation of A depending on the amount of content preserved after B editing. Also, the reputation of B is likely to be increased.

User-driven reputation needs a certain amount of user intersections and feedbacks, while content-driven reputation can generate results faster. Another advantage of content-driven reputation is that it can play against their success. That is to say, content-driven reputation is more accurate and reliable comparing to user-driven reputation.

2.2 Notation

The following notation will be used throughout the paper.

v_i	We assume i>0 versions. v_0 is empty.
r_i: $v_{i-1} \rightarrow v_i$	A revision which might contains text insertions, deletions, displacements and replacements.
a_i	Author i.
$d(v_i,v_j)$	The distance between v_i and v_j measures that how much change that occurs during this revision.

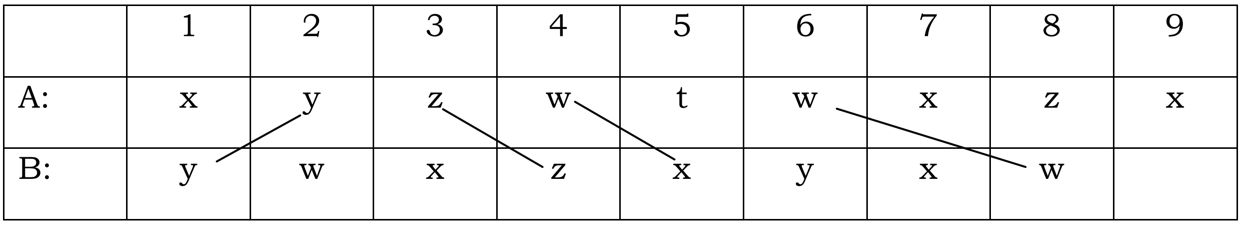
2.3 The string-to-string correction problem

The string-to-string correction problem is to determine the edit distance between two strings when considering the minimum cost that transform one string into the other.

The notions and formalizations that will be used in string-to-string correction problem are given by:

A, B	Two strings
a, b	Two single characters
Λ	The null string
A<i>	The ith character of string A
A<i: j>	The ith to jth characters (inclusive) of A
|A|	the length (number of characters) of string A
Edit operations	Change operation: a → b Delete operation: a → Λ Insert operation: Λ → b
Edit sequence	A sequence of edit operations: S = s1,s2, … sm
Cost	Cost of edit operation: γ(a → b) Cost of edit sequence: γ(S) = $\sum_{i=1}^{m} \gamma(s_i)$
Edit distance	The minimum cost of all edit sequences which transform A into B: δ(A,B)= min{ γ (S) | S is an edit sequence taking A to B }

Now, two sequences (A and B) are given, and through this example we can explore how to find the minimum cost when transform A into B.



	1	2	3	4	5	6	7	8	9
A:	x	y	z	w	t	w	x	z	x
B:	y	w	x	z	x	y	x	w	

In this diagram, each line represents that B<j> is derived from A<i> and this kind of derivation can either directly if A<i>=B<j> (In this example, it is B<1>

is derived from A<2> and B<1> = A<2>) or there is a change operation to A<i>(In this example, it is B<5> is derived from A<4> but A<4> ≠ B<5>).

A trace is when we ignore the transform order and consider of how an sequence transforms string A into B. If T represents a trace from A to B, the cost of T is given by:

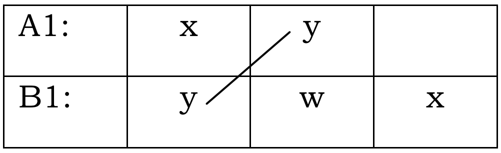
$$\mathrm{cost}(T) = \sum \gamma(\, A\langle i\rangle \rightarrow B\langle j\rangle) + \sum \gamma(\, A\langle i\rangle \rightarrow \Lambda) + \sum \gamma(\, \Lambda \rightarrow B\langle j\rangle)$$

The properties of traces are that:

Theorem 1: δ(A,B)=min{ cost(T) | T is a trace from A to B }

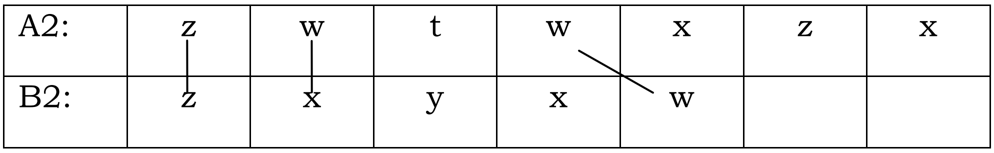
A trace can be spilt into two traces. In the example above, we could split them as:

T1:



A1:	x	y	
B1:	y	w	x

T2:



A2:	z	w	t	w	x	z	x
B2:	z	x	y	x	w		

Accordingly, cost(T) = cost(T1) + cost(T2).

When considered the computation of edit distance, D(i,j) is defined as :

$$D(i,j) \ = \delta(A\langle 1{:}i\rangle, B\langle 1{:}j\rangle)$$

$$(\ 0 \ \leq i \ \leq |A| \text{ and } 0 \ \leq j \ \leq |B|)$$

By using Theorem 1, D(i,j) is also the minimum cost trace from A<1:i> to B<1:j>. Then, we could get:

Theorem 2: $D(i,j) = \ \min \left\{ \begin{array}{l} D(i\text{-}1\ , j\text{-}1) + \gamma(A\langle i\rangle \rightarrow B\langle j\rangle), \\ D(i\text{-}1,j) + \gamma(A\langle i\rangle \rightarrow \Lambda), \\ D(i,j\text{-}1) + \gamma(\Lambda \rightarrow B\langle j\rangle) \end{array} \right\}$

For all i,j, $1 \ \leq i \ \leq |A|$ and $1 \ \leq j \ \leq |B|$

Theorem 3: $D(0,0) = 0;$

$D(i,0) = \sum_{r=1}^{i} \gamma(A<r> \rightarrow \Lambda);$

$D(0,j) = \sum_{r=1}^{j} \gamma(\Lambda \rightarrow B<r>)$

For all i,j, $1 \le i \le |A|$ and $1 \le j \le |B|$.

2.4 Edit Distance in content-driven reputation

There are several ways to compute the quantity measures and edit distance is usually be used which is based on insertions and deletions. Another two notations is defined as follows.

I(j,k)	It means that k words are inserted at position j.
D(j,k)	It means that k words are deleted at position j.
M(j,h,k)	It means that k words are moved from position j in version v to position h in version v'.

The edit distance between v_i and v_j is defined as

$$d(r,r') = I_{tot} + D_{tot} + M_{tot} - \frac{1}{2}\min(I_{tot}, D_{tot})$$

In this formula, I_{tot} is the total amount of words inserted in version v'. Similarly, D_{tot} is the total amount of text deleted and M_{tot} is the total amount of content moved. It can be noticed from this formula that every word that is inserted, deleted or moved contributes 1 to the edit distance in this formula, and every word that is replaced contributes 0.5.

2.5 The BASIC algorithm

Assessing how long that an edit lasts is a good way of measuring the content-driven reputation. That is to say, authors whose contributions last long gain reputation, while authors whose contributions lives short lose reputation. As it mentioned above, we use the concept of edit distance d(r,r') in the basic algorithm for measuring content-driven reputation.

$$\mathrm{Qual}(v_{i-1}, v_i, v_j) = \frac{d(v_{i-1}, v_j) - d(v_i, v_j)}{d(v_{i-1}, v_i)}$$

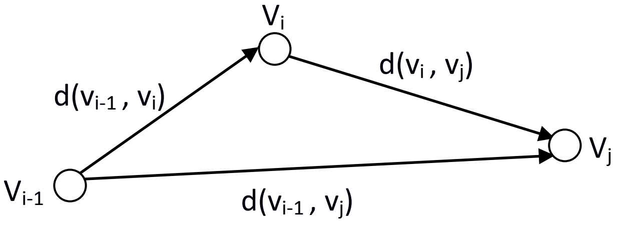


Figure 1 Distance involved in computation of Qual(vi-1 , vi , vj)

Given $d(v_{i-1},v_j)$, which represents the distance between versions i-1 and j and it is defined that r_i: $v_{i-1} \rightarrow vi$. If r_i is kept in the subsequent versions of the page, it means that these edits are useful and hence they are high quality to some extent. Otherwise, if r_i is completely undone in the following versions, it means that these edits are spam or incorporate material. If d satisfied the triangle inequality, $Qual(v_{i-1} , v_i , v_j) \in [-1,1]$. ($Qual(v_{i-1} , v_i , v_j)= -1$ means that r_i is entirely undone and $Qual(v_{i-1} , v_i , v_j) = +1$ means that r_i is completely preserved.) If the values outside this region, one of -1 and 1 is used depends on the value.

In the BASIC algorithm, all authors are initially given a reputation of zero and it considers versions i-1, i and j where $j-i \le m$ for constant $m \ge 1$. The constant m make a limitation of the number of previous versions that is considered.

Reputation is increased in BASIC algorithm where:

$$Inc(u,v,z) = \begin{cases} c_s \cdot d(u,v) \cdot Qual(u,v,z) \cdot w(\ r(a(z),t(z))\ ) & \text{if } a(z) \neq a(v); \\ 0 & \text{if } a(z) = a(v). \end{cases}$$

In this formula:

c_s is a constant and $c_s > 0$,

r(a(z) , t(z)) is the reputation of author a(z) at time t(z), that is to say, the reputation of author a at time t when version z is created.

w(x) is a monotonic increasing function, which ensures that $x \geq y \rightarrow w(x) \geq w(y)$. It can be given $w(x) = \log(1.1+x)$.

BASIC Algorithm

```
Input: A new version z.

Persistent variables r, $\bar{v}$

1.  $\bar{v} := \bar{v} * z$
    $k := |\bar{v}|$

2.  for versions(i-1, i, j) with $0<i-1<j$ and $k-i \le m$
    $r(a_i) := r(a_i) + Inc(v_{i-1},v_i,v_j)$
```

In this algorithm:

r(a) represents for current estimate of the reputation of author a,

$\bar{v}$ is a list of versions,

$|\bar{v}|$ is the number of elements in the list,

v_i is the ith element of $\bar{v}$,

$\bar{v} * z$ is the result of adding z at the end of $\bar{v}$.

2.6 Evaluation of BASIC algorithm and improved

The BASIC algorithm is easily attacked. One person can increase his reputation by using multiple users. For example, a person has two identity: the main ID is A and the sock-puppet ID B. By using B deleting all the content and then use A to recreate the page, he can gain reputation without performing any amount of productive work.

Thus, we need to improve the performance of BASIC algorithm to identity the useful work that is performed. The reputation-cap algorithm is given:

REPUTATION-CAP Algorithm

```
Input: A new version z.

Persistent variables r, $\bar{v}$

1.  $\bar{v} := \bar{v} * z$
    $k := |\bar{v}|$
```

2. for versions(i-1, i, j) with $0<i-1<j$ and $k-i \le m$

```
if inc($v_{i-1}$,$v_i$,$v_j$) ≥ 0
     then  r($a_i$) := max( r($a_i$), min(r($a_{i-1}$),r($a_i$), r($a_j$)+ Inc ($v_{i-1}$,$v_i$,$v_j$)));
     else  r($a_i$) + Inc ($v_{i-1}$,$v_i$,$v_j$)
```

3. Specifications

3.1 Risk

(High Risk) XX

(Moderate Risk)XX

(High Risk) XX

3.2 Challenge

1. XX
2. XX

4. Schedule

June	
July	
August	

5. References

[1]. Adler, B. Thomas, and Luca De Alfaro. "A content-driven reputation system for the Wikipedia." (2006).
[2]. Adler, B. Thomas, et al. "Measuring author contributions to the Wikipedia." Proceedings of the 4th International Symposium on Wikis. ACM, 2008.

[3]. Adler, B. Thomas, et al. "Assigning trust to Wikipedia content." Proceedings of the 4th International Symposium on Wikis. ACM, 2008.

[4]. De Alfaro, Luca, et al. "Reputation systems for open collaboration." Communications of the ACM 54.8 (2011): 81-87.

[5]. Chatterjee, Krishnendu, Luca de Alfaro, and Ian Pye. "Robust content-driven reputation." Proceedings of the 1st ACM workshop on Workshop on AISec. ACM, 2008.

[6]. Wagner, Robert A., and Michael J. Fischer. "The string-to-string correction problem." Journal of the ACM (JACM) 21.1 (1974): 168-173.

[7]. Wagner, Robert A., and Roy Lowrance. "An extension of the string-to-string correction problem." Journal of the ACM (JACM) 22.2 (1975): 177-183.

[8]. Golbeck, Jennifer Ann. "Computing and applying trust in web-based social networks." (2005).