Tracking topics on revision graphs of Wikipedia edit history

Bonan LI Jianmin WU Mizuho IWAIHARA
Graduate School of Information, Production and Systems, Waseda University,
Fukuoka 808-0135, Japan
libonan@ruri.waseda.jp jianmin.wu@moegi.waseda.jp iwaihara@waseda.jp

Abstract
Wikipedia is known as the largest online encyclopedia, in which articles are constantly contributed and edited by users. Past revisions of articles after edits are also accessible from the public for confirming the edit process. However, the degree of similarity between revisions is very high, so it is difficult to understand and summarize these small changes from revision graphs of Wikipedia edit history. In this paper, we propose an approach to give a concise summary to each change, by utilizing supergrams, which are consecutive unchanged token sequences, and topic detection methods.

Keywords
Wikipedia, delta summary, supergram

1. Introduction
User-generated contents (UGCs) and collaborative functionalities are becoming increasingly prevalent in Web applications. Wikipedia[8] is known as the largest online encyclopedia, in which articles are constantly contributed and edited by users. Past revisions of articles after edits are also accessible from the public for confirming the edit process.

The edit history of one article can be accessed by clicking the "history" tab at the top of the page. The page history contains a list of the page's previous revisions, including the date and time of each edit, the username or IP address of the user who made it, and their edit summary.

As shown in Figure 1, a revision graph is a DAG (directed acyclic graph) in which each node represents one revision with directed edges indicating their reference relationship. Users edit articles based on the current revision and occasionally on past revisions. Also, a completely new input may replace the current revision. Therefore in the revision graph, each node can have zero or more reference sources. Furthermore, a branch in the revision graph is created when the new revision is edited from a past revision that is not the current one.

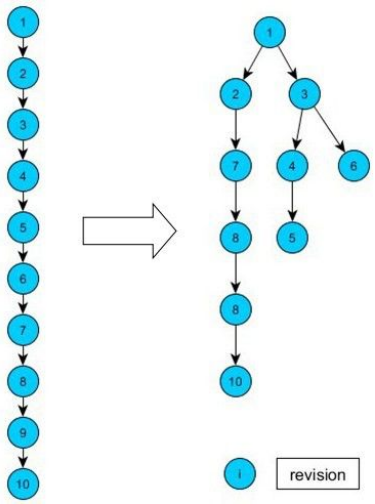

Figure 1. Example of revision graph

The current problem of Wikipedia articles, as the degree of similarity between revisions is very high. From Wikipedia edit history, each edge $<v_1, v_2>$ of the graph corresponds to the edit from revision r_1 to revision r_2. By taking diff between v_1 and v_2, what changes are made between them can be obtained. Figure 2 shows Jaccard similarity of two adjacent revisions of article “Natal Chart”, and the average of similarity is 98.2%.

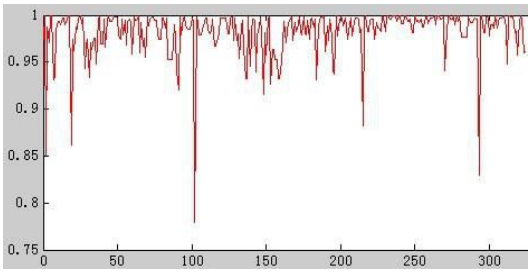

Figure 2 Jaccard similarity between two adjacent revisions of article" Natal Chart"

Table 1 shows statistics of randomly-chosen 10 articles from Wikipedia, in which a number of branches in the revision graph have been detected by our algorithm[6]. The emergence of branches has several reasons, such as malicious editors, minor updates, paragraph changed, and topic removal. Therefore it is necessary to make clear the causes of branches. Meanwhile, as a delta, namely the document diff between revisions, is often hard to understand for human, or too much detailed, containing simple or complex modifications. So how to summarize these deltas and understand by human is our objective. From the above, we can characterize the features of revision history of Wikipedia articles as "small changes, many branches, and unknown deltas."

Table 1 Statistics of Wikipedia articles

ID	Article Title	Total # of revisions	# of Branches
1	Racism	10,896	23
2	2006 Israel – Gaza conflict	2,456	12
3	PhpBB	1,312	37
4	Edith Wharton	1,114	16
5	Federal republic	717	33
6	Sarkar Raj	592	15
7	Grade inflation	456	24
8	Natal chart	346	11
9	Muhammad Naguib	283	8
10	Clarinet Concerto	256	12

Topic evolution in a series of scientific documents can reveal how research on one topic influenced research on another and helps us understand the lineage of topics.[3] We may adopt topic tracking to revision history of Wikipedia, to understand and objectively evaluate the contribution of an editor or an article. However, the characteristics of Wikipedia revisions, such as significant overlaps and minor changes, are quite different from scientific documents or news articles.

To address the challenges mentioned above, in this paper, we focus on describing our work from three aspects:

- ❖ Detect each change of revisions of the given articles with timestamps by unigram and produce summaries of these changes.
- ❖ Construct easily understandable summaries by utilizing supergrams[6], which are consecutive unchanged token sequences.
- ❖ Capture topic keywords from supergrams, but if supergrams are too many or too long, rank keywords by TF-IDF. Also mark up the revision graph of Wikipedia with generated summaries.

The rest of this paper is organized as follows. In Section 2 we describe the background of this research and survey related work. In Section 3 we describe basic concepts regarding our problem, and explain our method to construct supergrams and generate summaries. In Section 4 we show experimental evaluation of our method and the results. Finally, concluding remarks and future work are shown in Section 5.

2. Background

2.1 User-generated contents and Wikipedia

Contents created by users have become widely popular and well accepted. Wikipedia is a representative example of web sites delivering UGCs. In Wikipedia, users other than the contributor of an article may evaluate

the article, suggest changes, or even make changes. A warning system in Wikipedia is in operation such that a warning is given when an author is just espousing an opinion, certain statements are not verifiable, or has been called into question by other users.

2.2 Related Work

Latent Dirichlet Allocation (LDA)[1] is proposed is a flexible generative probabilistic model for collections of discrete data. The basic idea of LDA is that a document can be considered as a mixture of a limited number of topics and each meaningful word in the document can be associated with one of these topics. But this algorithm just focuses on the popular topic for the corpus consisting of different articles.

Zhu et al.[7] proposed an algorithm to accomplish topic detection and tracking task (TDT) in the threaded discussion community environments. They design several extensions to the basic TDT framework, focusing on discussion data. Different topics are overlapping and multiple topics may be discussed at the same time. In our problem, in changes in creating a new revision of a Wikipedia article occur in various scales and styles. When the size of a change is small, we cannot detect topics just from the delta. Also, we need to contrast topics with significant patterns of the revision graph, such as branches.

Yan Chen et al[2], proposed a real-time framework for detecting hot emerging topics for organizations in social media context. Developed semi-supervised learners to facilitate timely identification of hot emerging topics for organizations. But their styles of how new entries are created are quite different from revision history.

3. Tracking topics on revision graph

A revision graph G[5] is a directed graph where each edge represents the derivation relationship between two revisions. Revision graphs are important to capture features such as frequencies of reverts, represented by branches. But just derivation relationships of the revision graph do not present any illustration or explanation. So a plain revision graph is not enough for further understanding of edit history, such as when a particular topic was introduced to a Wikipedia article. In order to clarify topic evolution and what event causes a branch, we collect revision deltas from each of the two adjacent revisions at first.

DEFINITION 1 (Delta)

Given an edge $<v_i,v_j>$ in a revision graph G, the *delta* D of $<v_i,v_j>$ is the sequence $<t_1, f_1>$, $<t_2, f_2>$,..., $<t_n, f_n>$ such that t_k is an added token between v_i and v_j, and f_k is the frequency of token t_k in revision v_j.

3.1 Challenge of summarizing deltas

Before we capture topics from deltas, let us consider adding portions of deltas to the revision graph, so that trends of edits can be easily recognized. However, simply adding deltas to graph edges produces floods of text, or hard-to-read text fragments. In this case, we need to find appropriate summarization of deltas, such as:

- Extracting phrases that capture topics of the deltas. Sometimes, the delta is likely to be a complete sentence or a paragraph, we can extract text surrounding a delta to find important phrases based on term frequency.
- Minor updates, such as spell corrections, plural transformation, should be ignored.
- One delta may contain multiple text fragments, for interfering fragments, such as URL link, nonsense words, identifier, we need to filter out them by a stop word list and regular expressions. Only important fragments need to be detected.
- To avoid flooding, we need not to decorate every edge with summaries.

Only significant edges need to be decorated.

3.2 Token transition graph construction

According to the methods of summarizing deltas in the above, a delta is generated from each edge <v_i,v_j> of the revision graph. Let us consider deltas from a real article.

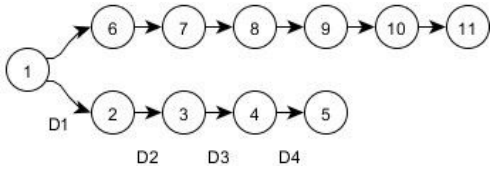

Figure 3 Deltas in a branch

Example 3.1 As shown in Figure 3, four deltas by manually from one branch from revision 1 are shown below.

D_1: Explosion on Boylston Street.

D_2: Two loud explosions on Boylston Street.

D_3: Friends said explosion occurred on Boylston Street.

D_4: A news report explosion ripped through Boylston Street.

Then we detect tokens and construct a token transition graph for all the deltas. As shown in Figure 4, each node is labeled with a token, and each edge represents at least one consecutive occurrence of two tokens (a bigram). We found “explosion” and “Boylston Street” appear in all of D_1,…, D_4.

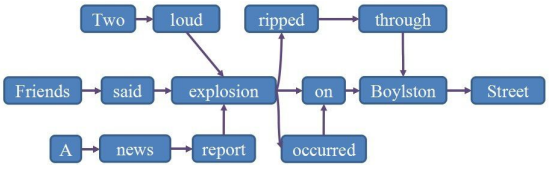

Figure 4 Token transition graph

When we collect deltas, we need to keep stop words in order to ensure readability of token sequences, that is why there are stop words like ‘a’, ‘on’, ‘through’ in the above graph. Otherwise, humans will have difficulties in understanding such token sequences.

Path contraction is to merge two adjacent nodes such that one node is the sole destination or origin of the other. As shown in Figure 5, tokens <two, loud>, <friends, said>, <a, news, report>, <ripped, through>, <Boylston, Street> can be merged into new token sequences. Through updating these new tokens in the original deltas $D_{1,\ldots,}D_n$, we obtain new deltas $D'_{1,\ldots,}D'_{n'}$.

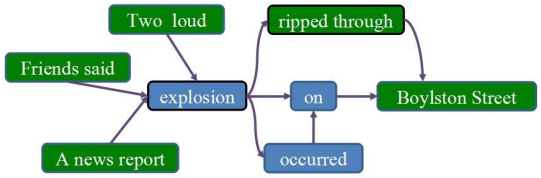

Figure 5 Path contraction

3.3 Supergram

There exist token sequences that keep appearing throughout all the deltas within a time scope. We can group such unchanged consecutive token sequences into *supergrams*[6].

DEFINITION 2 (Supergram)

A supergram $s=t_1t_2t_3...t_n$ in a delta subset DS, where DS is called a comparison scope, is an *n*-gram (*n*>=1) such that *s* occurs in all the deltas in DS, and no token sequence that properly contains *s* occurs in all of the deltas in DS.

In the result of Example 3.1, “explosion” and “Boylston Street” are supergrams. As we can see, supergrams are capturing meaningful phrases, so we should focus on utilizing supergrams in summarizing deltas. There are situations such that a large number of nodes are in one linear chain, which would come from unchanged large consecutive paragraphs, so its supergram becomes too long. In this case, we have to further summarize such a long supergram. Thus ranking tokens within the supergram is an appropriate method to resolve this problem.

TF-IDF is often used as a weighting factor in information retrieval and text mining. In a delta set DS, we can calculate the TF-IDF score as weighting in each supergram having frequency in the delta. The higher the score is, the more the supergram can represent the topic

of deltas. Meanwhile, we can present supergrams that emerges at a branch, as a potential cause of the branch. An emerging supergram is such that it is unique or in the first node after the branch node.

3.4 Three categories of topics

In a revision graph, one revision usually have multiple topics. We classify these topics based on the topic categories described as:

1) **Popular topic** (Category 1) is such that it appears in most of revisions. In general, we can discover such popular topics by LDA from the entire revision graph.

2) **Surviving topic** (Category 2). This is a topic that appears at a revision, and continue to appear until the latest (namely, newest) revision. Also, it can also be described as a surviving topic is in the mainstream. Here, the mainstream is the chain from the first vertex to the latest vertex in the revision graph. After a period of edits, certain topics become stable and survive to the latest.

3) **Extinct topic** (Category 3). This is a topic that is not surviving. In Example 3.1, "explosion, Boylston Street" belongs to this topic category. The definition of surviving topics is relative to the current revision, so if there are large amount of deletes after the current revision, several topics may be lost and surviving topics can be changed to extinct.

Figure 6 shows a part of topic visualization, and three categories of topics are marked in the revision graph. We can examine the evolution of the article over time.

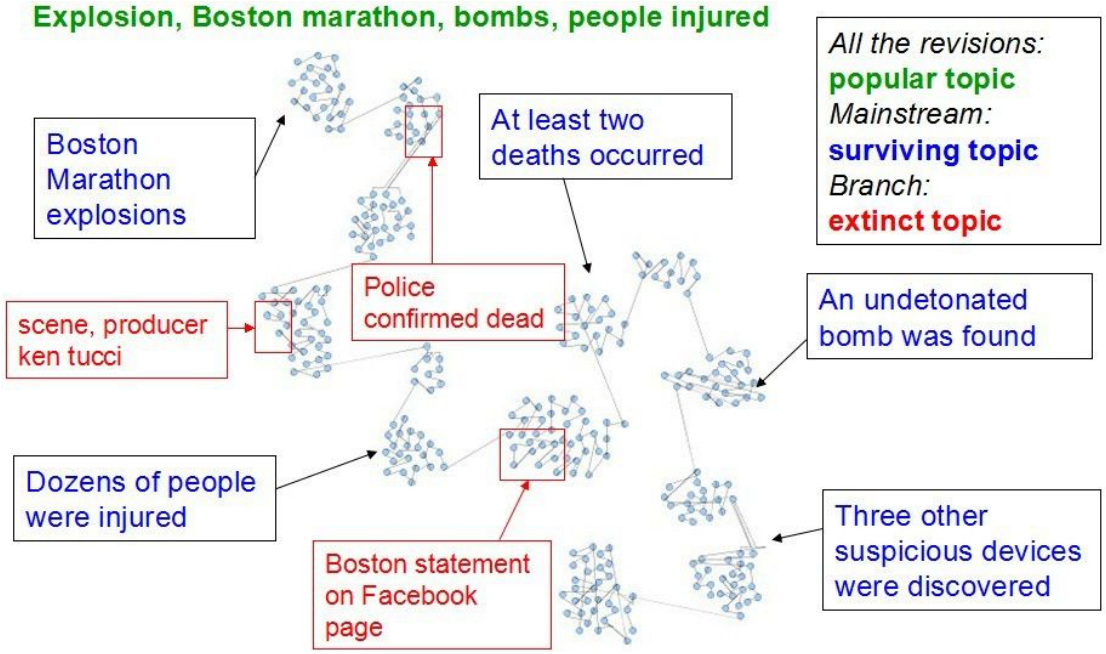

Figure 6 Part of topic visualization of "Boston Marathon bombings"

4. Experimental Evaluation

To evaluate the quality of topics generated by our method, we conduct human judgment evaluation and compare with two representative methods: 1) Baseline, which merges deltas simply and selects top five tokens based on term frequencies and 2) LDA on revisions merging. First we generate a revision graph of article "Boston Marathon bombings" based on n-gram cover[5]. Then we randomly selected 10 branches from the first 300 revisions. The resulting summaries at each branch by three methods are shown in Table 2. To evaluate qualities of the summaries, we asked ten volunteers to do ranking the results. We use alphabet 'A', 'B',

'C' as evaluation levels, where 'A' is the best, and 'C' is the worst.

Table 2 Generated summaries by three methods

Branch ID	Baseline (TF on deltas)	LDA on revisions	Proposed method (TF-IDF on supergrams)
1	Least, people, just, around, edt	category, explosion, boston, line, marathon	reported people lost limb
2	explosion, boston, least, police, killed	people, boston, explosion, line, injured	Marathon explosion, runner killed
3	hour, about, confirmed, announce, news	boston, explosion, marathon, category, injured	police confirmed dead
4	reference, scene, wbz, category, confirmed	boston, people, explosion, injured, line	scene, producer ken tucci
5	Marathon, statement, bomb, understand, exploded,	boston, marathon, explosion, line, finish	boston statement on Facebook page
6	people, hospital, local, reference, lost	boston, explosion, April, date, line	people lost limb in hospital
7	reported, finish, race, three, winners	boston, explosion, cite, line, marathon	reported people dead, injured
8	mandarin, evacuated, hotel, marathon, boston	boston, marathon, explosion, April, line	outside mandarin hotel
9	boston, police, spokeswoman, people, department	boston, marathon, explosion, line, April	no indication, how many people injured
10	boston, statement, facebook, hospital, working	boston, marathon, ref, explosion, line	People in local hospital.

As shown in Table 3, we can see that our proposed method of TF-IDF on supergrams has the best score, and LDA on deltas is the worst. In further analysis, Baseline has 11 votes of level 'A'(best), and these votes mainly concentrated in branches No.6 and No.8. The common point of these two branches is that the delta set DS is too small. The smaller the delta set DS is, the more the result of TF-IDF on supergrams tends to become identical to Baseline.

Table 3 Results of rankings by human judgment

Rank\ Methods	Baseline (TF on deltas)	LDA on revisions	TF-IDF on supergrams
A(Best)	11	0	89
B	88	2	10
C(Worst)	1	98	1

5. Conclusion

In this paper, we proposed a method for detecting topics on deltas of revision graphs, which represent revision history of Wikipedia articles. First, we detect deltas, namely the diff from each edge of revision graph through summarizing them. Secondly, we construct

supergrams, and then we capture keyword phrases from the supergrams by TF-IDF, and generate the results as topics of the deltas. Also we showed visualization by adorning edges of revision graphs with topics.

In future work, we try to improve our method from two aspects. We consider adding a new score on each edge, not just by N-gram diff score. The purpose here is to improve sensitivity of the score. Also we plan to improve delta summarization. Informativeness based keyword extraction[4] is a novel unsupervised keyword extraction approach, that uses clustering and three levels of word evaluation to address the challenges of short documents.

Reference

[1] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. the Journal of machine Learning research, 3, 993-1022.

[2] Chen, Yan, et al. "Emerging topic detection for organizations from microblogs." Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. ACM, 2013.

[3] He, Qi, et al. "Detecting topic evolution in scientific literature: how can citations help?." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009.

[4] Timonen, M., Toivanen, T., Teng, Y., Chen, C., & He, L. (2012). Informativeness-based Keyword Extraction from Short Documents. In KDIR (pp. 411-421).

[5] Wu, Jianmin, and Mizuho Iwaihara. "Wikipedia Revision Graph Extraction Based on N-Gram Cover." Web-Age Information Management. Springer Berlin Heidelberg, 2012. 29-38.

[6] WU, Jianmin; IWAIHARA, Mizuho. Revision graph extraction in Wikipedia based on supergram decomposition. 2013.

[7] Zhu, Mingliang, Weiming Hu, and Ou Wu. "Topic detection and tracking for threaded discussion communities." Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on. Vol. 1. IEEE, 2008.

[8] Wikipedia, http://en.wikipedia.org/wiki/Wikipedia