Edit summarization in Wikipedia based on supergram selection

呉　建民† 李　波南‡ 岩井原瑞穂※

†早稲田大学大学院情報生産システム研究科

〒808-0135 福岡県北九州市若松区ひびきの ２－７

E-mail: †jianmin.wu@moegi.waseda.jp, ‡libonan@ruri.waseda.jp ※iwaihara@waseda.jp

Abstract Document summarization has been well studies in recent years, but the basis of the existing methods fails in the scenario of Wikipedia edit history, in which revisions have significant mutual overlaps. In this paper, we propose a method to automatically summarize contributed contents during a specified edit period of a Wikipedia article, into a group of maximal-length phrases, which we call supergrams. Two supergram selection algorithms, TF-IDF and Extended LDA ranking are developed to pick up representative supergrams. We conduct a preliminary objective evaluation on these methods' capabilities of summarizing on short text fragments against conventional document summarization methods.

Keyword Wikipedia, text summarization, text mining

1. Introduction

User-generated contents (UGCs) [2] are contributed voluntarily by ordinary people and distributed through interactive medias such as blogs, discussion form posts, wikis, and digital images. The use of UGCs has seen rapid growth in recent years. Wikipedia [18] is among the most successful UGC platforms, known as the largest online encyclopedia, in which articles are constantly contributed and edited by users. The collaborative online encyclopedia currently ranks 6th on the Alexa list of top web destinations [16].

Past revisions of Wikipedia articles after edits are 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 [19]. 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 authored it, and their edit summary. A *revision graph* is a DAG (directed acyclic graph) where each node represents one revision and each directed edge represents a derivation relationship from the origin node to the destination node. Users create a new revision by editing either the current revision, or one of past revisions. Also, a completely new input may replace the current revision. In general, confluence of edges occurs when one revision is created by merging multiple parents. But in real edit histories, such merges are seldom observed. Therefore, we focus on edit history that is modeled as a revision tree, as shown in Figure 1. In this paper, we call by a *branch* a subtree that is generated by removing the unique directed path from the initial revision to the current revision. When a new revision is edited not from the current revision but from a past revision, all the revisions that do not have a directed path to the new revision will belong to one of the branches.

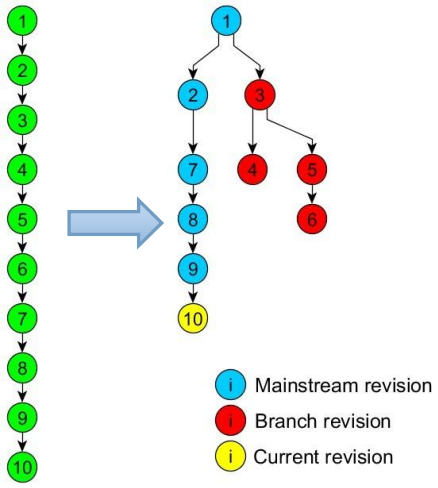

Fig. 1. Example of a reconstructed revision graph (tree)

One of major issues in utilizing Wikipedia edit history is that the degree of similarity between consecutive revisions is very high. In the revision graph, each edge $<v_1, v_2>$ corresponds to the edit from revision r_1 to revision r_2. By taking document difference (or *diff*) between v_1 and v_2, what changes were made between them can be obtained. Figure 2 shows Jaccard similarity of two adjacent revisions of article "Natal Chart," where the average of similarity is 98.2%, while we can observe significant

changes through occasional drops of similarity.

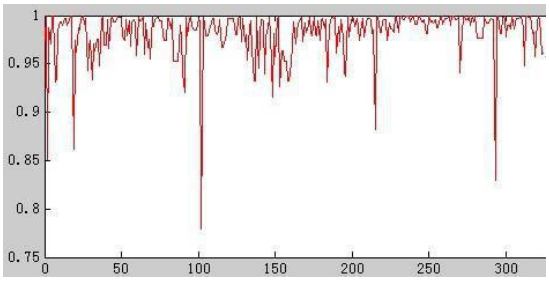

Fig. 2. Jaccard similarity between two adjacent revisions of article "Natal Chart"

Table 1 shows branch statistics of randomly-chosen 10 articles from Wikipedia, in which a number of branches in the revision graph is detected by our algorithm[14]. The causes of branches can be classified into a number of reasons, such as: revert of a minor update, malicious editing, paragraph reorganization, and topic removal. For validating edit process, it is necessary to summarize changes occurred in these branches. Meanwhile, a delta, or diff between revisions, is often inappropriate for summarization, because it is too large, too much detailed, too little, or too much fragmented. So our objective is to find summarization of deltas that are reflecting edit history well and easily understood by human.

Table 1. Wikipedia articles statistics.

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 [5]. We may adopt topic tracking to revision history of Wikipedia by edit summarization, 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 the following three issues:

1. Detect each change of revisions of a given article by unigrams.

2. Construct easily understandable summaries by utilizing supergrams [14], which are consecutive unchanged token sequences. We consider generating summaries for each meaningful part of the revision graph, such as a scope in the mainstream and extinct branches.

3. 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, we conclude the paper with future work in Section 5.

2. Related work

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 [12]. 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.

Regarding topic detection, Latent Dirichlet Allocation (LDA) [1] is proposed as 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 is suitable for detecting prevalent topics for a large corpus consisting of different articles. For selecting significant phrases from overlapping deltas, we need a different mechanism.

Zhu et al. [15] proposed an algorithm to accomplish topic detection and tracking task (TDT) in collaborative environments of threaded discussions. In our problem, 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 one delta.

TIARA-generated visual text summary [7][9] is to build a topic-based, interactive visual analytic tool that aids user in analyzing large collections of text. But it is based on parallel topics and each topic must exist in the entire lifespan.

Yan Chen et al. [3] 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. Kittur et al. [6] studied the distribution of topics over two years of edit history in Wikipedia. However, their work focuses on the global topic trend of all articles' edit history rather than a single one. Also the topic of each article is calculated from its annotated Wikipedia categories, while our work is focused on revision-wise topic changes. Georgescu et al[4]. studied temporal summarization of Wikipedia updates through detecting edit bursts responding to external events. However, their approach does not reflect edit contexts. We utilize revision graphs to detect edit contexts and reflect into summarizations, such as whether revision deltas are in an extinct branch or surviving.

3. Edit summarization on revision graph

3.1 Revision graph and delta

A revision graph *G* is a directed graph where each edge represents the derivation relationship between two revisions. From a revision graph we can detect edit reverts with scale in terms of branch sizes. However, without textual description of deltas, it is difficult to analyze the causes of branches, such as addition of new topics, rephrasing of existing topics, or vandalisms. It is also important to know the time when a particular topic is introduced to an Wikipedia article. In order to detect topic evolution along the revision graph, we collect revision deltas from each of the two adjacent revisions 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.

Note that the above deltas are based on addition to preceding revisions. Deltas based on deletion from preceding revisions, and topic detection from them can be constructed similarly. But since articles are generally growing in size over time, we focus on addition deltas.

3.2 Scoping mainstream

We extract revisions of the target article from the Wikipedia edit history file, where the revisions are numbered in chronological order[20]. Then we perform preprocessing, such as removing punctuations, wiki markup language operators, and vandalism [8]. A simple version of vandalism is detected as deletion of the whole article.

We extract the revision graph consisting of vertices and edges, then obtain mainstream and branches. The *mainstream* is the path from the initial revision to the current revision. Each *branch* is regarded as a subtree rooted at a revision vertex of the mainstream. The mainstream can have an extremely larger tree height than branches. Assigning a single topic to such a long path is insufficient for capturing topic changes within the path. So we need to find significant points that indicate a topic change, and divide the mainstream into *scopes*. We also assume that revisions in a branch form a scope. We check the changes of the text length (token counts) of each revision along the timeline. When the size of the diff of two adjacent revisions is greater than a certain threshold, we regard it as the starting point of a new topic. Here we set the threshold as 50 tokens.

3.3 Challenge of summarizing deltas

Let us consider a simple idea of adding portions of deltas to the revision graph, to recognize trends of edits. However, simply labelling graph edges by deltas produces floods of text, or hard-to-read text fragments. In this case, we need to find appropriate summarization of deltas, such as:

1. Phrases that capture topics of deltas should be extracted. Deltas are diverse in size; larger deltas can contain a complete sentence or a paragraph, but smaller deltas are insufficient to find phrases. In this case, we need to extract text surrounding the small delta.

2. Minor updates, such as capitalizing/uncapitalizing letters, spell correction, plural transformation, should be ignored.

3. One delta may consist of multiple text fragments, interrupted by wiki markups, URL links, nonsense words, and identifiers. We need to filter out them by a stopword list and regular expressions. Only meaningful fragments need to be detected.

4. To avoid flooding, we should not to adorn every edge with summaries. Only significant edges need to be

adorned.

3.4 Token transition graph construction

Now we discuss extracting phrases from deltas, where deltas are taken from a branch or a scope in the mainstream, to reflect the structure of the revision graph. Let us consider a real example of deltas from article "Boston Marathon bombings". Four deltas from one branch are shown below.

Example 3.1

D1: Explosion on Boylston Street.

D2: Two loud explosions on Boylston Street.

D3: Friends said explosion occurred on Boylston Street.

D4: A news report explosion ripped through Boylston Street.

We detect tokens and construct a token transition graph from the union of all the deltas. As shown in Figure 3, each vertex is labeled with a token, and each edge represents at least one consecutive occurrence of two tokens (a bigram). We find that "explosion" and "Boylston Street" appear in all of D_1,..., D_4. When we collect deltas, we need to retain stopwords in order to ensure readability of token sequences, that is why there are stopwords like 'a', 'on', 'through' in Figure 3. *Path contraction* is to merge two adjacent nodes such that one vertex is the sole destination or origin of another vertex. As shown in Figure 4, tokens <two, loud>, <friends, said>, <a, news, report>, <ripped, through>, <Boylston, Street> are merged into new token sequences. Through updating these new tokens in the original deltas D_1,...,D_n, we obtain new deltas D'_1,..., $D'_{n'}$.

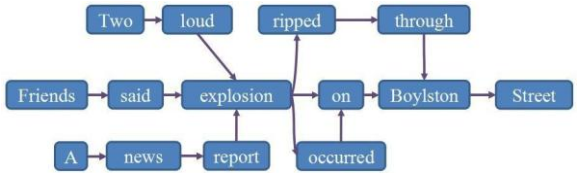

Fig. 3. Token transition graph.

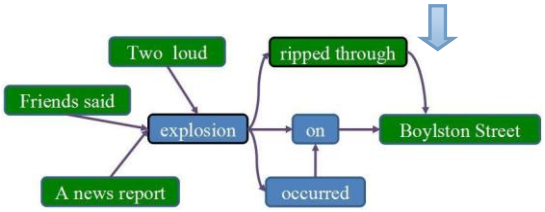

Fig. 4. Path contraction.

3.5 Supergram

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

DEFINITION 2 (Supergram) A supergram $s=t_1t_2t_3...t_n$ in a delta set DS 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 deltas of Example 3.1, consecutive tokens such as "A news report," "ripped through", and "Boylston Street" satisfy the condition of supergram and grouped into supergrams in Figure 4. However, token "explosion" is not properly contained by any other token sequences, so the token itself is a supergram. As we can see, supergrams are capturing meaningful phrases and reflecting unchanged sequences within the scope under consideration.

A large number of tokens could be contained in one linear chain, which come from a large unchanged consecutive text. In this case, its supergram becomes too long. Since such long supergrams do not represent changes occurred in the scope, we should exclude them from summaries. On the other hand, many short supergrams like phrases and specific nouns occupy a certain proportion in the delta set. In this case, we apply ranking of tokens occurring in supergrams to find significant supergrams for summaries.

3.6 TF-IDF on supergram

TF-IDF is often used as a weighting scheme in information retrieval and text mining. In a supergram set SS, we can calculate the TF-IDF score as weighting of each supergram. Details of the method are shown in Figure 5. According to TF-IDF scores in Figure 5, if length(supergrams) >10, remove stopwords and do TF-IDF to return SSnew, else return the initial SS. Supergrams contain stopwords for readability, but stopwords need to be removed when computing TF/IDF. The higher the TF-IDF score is, the more the supergram can represent the topic of deltas. Meanwhile, we can present supergrams that occur at a branch as a potential cause of the branch.

3.7 Three edit-contextual categories of topics

In a revision graph, one revision usually contains multiple topics, and the amount of topics will increase over time. We classify these topics based on which context in the revision graph the topics represent:

1. **Popular topic** is such that it is most prominent among all the revisions in a view. We can discover such topics by LDA[1][10] from the revision graph.
2. **Surviving topic** is a topic that appears at a revision, and continues to appear until the latest (current)

revision. It can also be described as a surviving topic in the mainstream. After a period of edits, certain topics become stable and survive to the latest.

3. **Extinct topic** is a topic that is not surviving to the current revision. In Example 3.1, "Boylston Street" belongs to this topic property. 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.

Algorithm

Algorithm ExtractPopularTopics
Input: all the revisions on the graph
Output: (token : probability)
Method: Apply LDA (Gibbs Sampling) [17]

Algorithm ExtractingSurvivingAndExtinctTopics
Input: A set of revisions of an article
Output: supergram set SS_{new}
Method: Divide revisions into mainstream(surviving) and branches(extinct).
For each branch *b*, **call** DetectTopics(*b*)
Divide the revisions of the mainstream into scopes and for each revision set *s* of the scope, **call** DetectTopics(*s*)
Function *DetectTopics(scope s)*
for each scope set **do**
1) Compute deltas by taking the diff for each of two adjacent revisions (Section 3.3)
and obtain the delta set DS (including freq).
2) Construct the token transition graph from DS, where each edge is assigned
a token. Then perform path contraction (Section 3.4).
3) Construct Supergrams
Select tokens of length greater than 2 as supergrams and insert into set SS_{new}.
4) TF-IDF score
for each delta *d* in DS and **for each** supergram *s* appearing in *d* **do**
if the length of *s* > 10, remove stopwords from *s*. Compute TF-IDF weight of *s*,
where TF is the frequency of *s* in *d* and
DF = (size of DS) / (# of deltas containing *s*).
return SS_{new}

Fig. 5 The proposed algorithm for revision topic detection

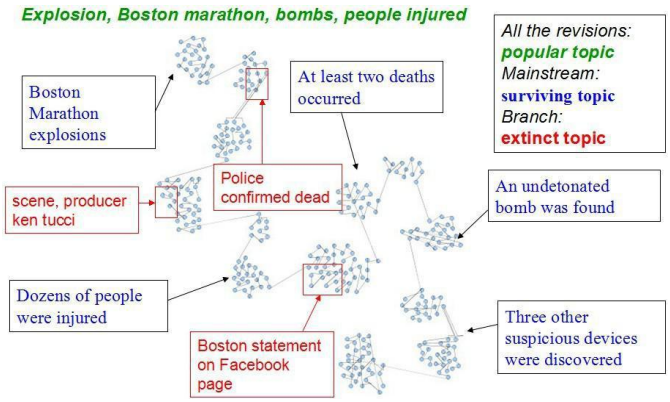

Fig. 6. Conceptual visualization of topics in "Boston Marathon Bombings"

Figure 5 shows the proposed algorithm for detecting topics by three categories. Figure 6 shows a conceptual depiction of three topic categories, where three topic categories are marked on the revision graph. From the figure, we can examine the evolution of the article over time.

4. Experimental evaluation

4.1 Data set

To evaluate quality of extracted topics, we used Wikipedia article "Nazi Germany" as benchmark, which has 1,093 revisions. As described in Section 3.3, we first execute cleaning the revision set and then compute deltas for each pair of adjacent revisions. Then we apply the algorithm of Figure 5, to divide the revision graph into branches and scopes, and then obtain supergrams.

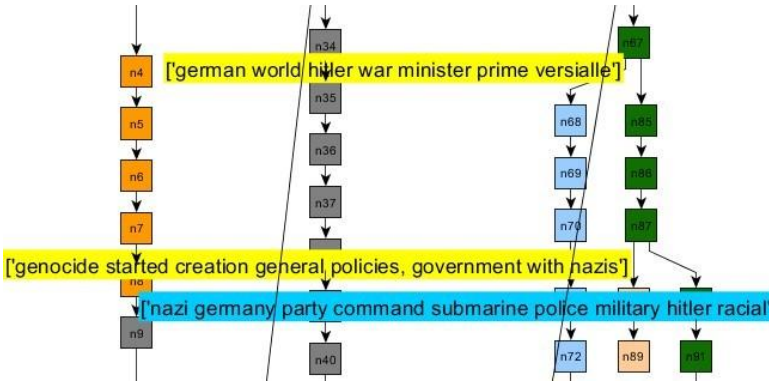

Fig. 7. Zoom-in of a part of the system output.

4.2 Result analysis and evaluation

Figure 7 shows screenshot of our system implementing the proposed method, where each color of vertices corresponds to a scope, and the first edge in each scope is labeled with its topic. The supergrams in yellow background color are extinct topics and those in blue

background color are surviving topics.

An interesting relationship is found between popular topics and surviving topics. Most of top (30+) tokens of popular topics are not evenly distributed in the scopes and branches; they tend to appear in earlier revisions of the history. The main reason is that if a topic appeared early and remained in the mainstream, the topic will have more frequencies and tend to be selected as a popular topic in LDA calculation. This matches our purpose of popular topics, since central topics of the article will be added at the beginning and will not be deleted later. Surviving topics are representing a scope of the mainstream, which can illustrate what topics are introduced in the scope.

To evaluate the quality of topics generated by our system, we conduct human judgment evaluation and compare with two representative methods. Our problem of summarizing edit deltas involve three levels of hierarchical objects: revisions, deltas and supergrams. Therefore we contrast our proposing algorithm with topic summarization over different objects.

1. **TF-IDF on deltas (Baseline)**, which simply merges deltas within a scope and selects top six tokens based on TF-IDF. When LDA is applied on deltas or supergrams, the size of the training set is insufficient, which causes that the result of LDA becomes close to TF. Thus LDA on deltas is not chosen.
2. **LDA on merged revisions**. In this case, we use the merged text of the revisions within a scope for LDA. Since LDA can find topic distribution from a corpus larger than the other methods, LDA is expected to perform better than the cases applied on deltas and supergrams.
3. **TF-IDF on supergrams (Proposed method)**, as described in Figure 5.

Table 2 shows the expected performance of three methods on the three levels of objects.

Table 2. Expected performance of three methods on the three levels of objects.

Object level/scoring	**TF-IDF**	**TF**	**LDA**
Revisions	Neutral	Neutral	**Good**
Deltas	**Good**	Neutral	Bad
Supergrams	**Good**	Neutral	Bad

Table 3. Comparing detected topics by three algorithms.

ID	**Baseline (TF-IDF on deltas)**	**LDA on merged revisions**	**Proposed method (TF-IDF on supergrams)**
1	humanity, crime, trial, allied, put, ever	nazi, germany, german, war, police, party	spring 1945 of fall 1944 east
2	hitler, german, including, many, world, prime	german, nazi, force, hitler, italian, invasion	german, world, hitler, war, minister, prime, versialles
3	government, general, started, creation, genodic, policy	nazi, german, war, force, italian, invasion	genocide started creation general policies, government with nazis
4	operation, barbarossa, north, campaign, theatre, eastern	nazi, german, von, han, karl, war	this known african, south north theatre campaign
5	nazi, german, td, germany, nsdap, party	war, party, nazi, hitler, regime, army	nazi, germany, party, command, submarine, police, military, hitler, racial
6	karl, von, german, han, wilhelm, force	nazi, german, force, hitler, invasion, italian	force, italian, britain, war, invasion, defeat, wilhelm, campaign, soviet, france
7	reich, nazi, third, war, tr, germany	war, nazi, karl, wilhelm, force, franz	reich, europe, adolf, empire, glorious, regime, power, republic failure
8	nazi, war, von, germany, german, wiltelm	nazi, war, force, germany, von, party	war, wilhelm, italian, walther, friedrich, invasion, campaign, battle
9	impure, refered, also, see, cultural, history	germany, war, nazi, invaded, regime, united	also see history germany
10	nazi, reich, german, von, hitler, minister	war, reich, germany, minister, nazi	hilter, state, goverment, karl, party, minister

We randomly selected 10 scopes from article “Nazi Germany.” The resulting summaries by the three methods

are shown in Table 3. To evaluate qualities of the summaries, we asked ten volunteers to rank the results. We use symbols ‘A’, ‘B’, ‘C’ as evaluation levels, where ‘A’ is the best, and ‘C’ is the worst.

Table 4. Results of rankings by human judgment.

Rank\ Methods	Baseline (TF-IDF on deltas)	LDA on revisions	TF-IDF on supergrams
A(Best)	17	2	81
B	79	5	16
C(Worst)	4	93	3

As shown in Table 4, we can see that our proposing method of TF-IDF on supergrams has the best score, while LDA on revisions is the worst. In further analysis, two interesting findings have aroused our attention. First, Baseline has 17 votes of level ‘A’ (best), and these votes mainly concentrated in No.2 and No.9. The common point of these two data sets is that the delta set is too small. The smaller the delta set is, the more the result of TF-IDF on supergrams tends to become identical to Baseline. Second, as observed in No.3 and No.4, TF-IDF on supergrams retain topic phrases better than the other methods. The other two methods are generating topics that contain token fragments, which makes difficult to capture the meaning of deltas.

5. Conclusion

In this paper, we proposed a method for edit summarization on deltas of revision graphs, which can explain changes in the revision history of Wikipedia articles. Our approach is based on supergrams, which consist of consecutive tokens within a revision subset such as a branch or scope, so that edit contexts are reflected into summarizations. We found that TF/IDF scoring on supergrams has the best performance in finding useful phrases.

In future work, we try to improve our method through refinement of scoring by reflecting more detailed edit contexts. We also plan to improve delta summarization, by incorporating keyword extraction that performs well on short documents[11].

6. Reference:

[1] Blei, D. M., Ng, A. Y., & Jordan, M. I.. Latent dirichlet allocation. J. Machine Learning Research, 3, 993-1022, 2003.

[2] Carmel D, Roitman H, Yom-Tov E. On the relationship between novelty and popularity of user-generated content. ACM Tran. Intelligent Syst. &Technology (TIST), 2012, 3(4): 69.

[3] Chen, Yan, et al. "Emerging topic detection for organizations from microblogs." Proc. ACM SIGIR Conf. Research and Development in Information Retrieval. ACM, 2013.

[4] Georgescu, M., D. D. Pham, et al. Temporal summarization of event-related updates in Wikipedia. Proc. 22nd Int. Conf. World Wide Web companion (WWW '13 Companion), pp. 281-284, 2013.

[5] He, Qi, et al. "Detecting topic evolution in scientific literature: how can citations help?" Proc. ACM Conference on Information and Knowledge Management (CIKM), 2009.

[6] Kittur, A., Ed H. Chi and B. Suh. What's in Wikipedia?: mapping topics and conflict using socially annotated category structure. Proc. ACM SIGCHI Conf. Human Factors in Computing Systems (CHI '09), 2009.

[7] Liu, S, Zhou M X, Pan S, et al. Interactive, topic-based visual text summarization and analysis. Proc. ACM Conf. Info. and Knowledge Management (CIKM), pp. 543-552, 2009.

[8] Mola-Velasco, S M. Wikipedia vandalism detection. Proc. 20th Int. Conf. World Wide Web, pp. 391-396, 2011.

[9] Pan, S, Zhou M X, Song Y, et al. Optimizing temporal topic segmentation for intelligent text visualization. Proc. Int. Conf. Intelligent User Interfaces, pp. 339-350, 2013.

[10] Song, Y., Pan, S., Liu, S., Zhou, M. X., & Qian, W. Topic and keyword re-ranking for LDA-based topic modeling. Proc. 18th ACM CIKM, pp. 1757-1760, 2009.

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

[12] Wierzbicki A, Turek P, Nielek R. Learning about team collaboration from Wikipedia edit history. Proc. Int. Symp. Wikis and Open Collaboration (WikiSym’10), No.27, 2010.

[13] Wu, Jianmin, and Mizuho Iwaihara. "Wikipedia Revision Graph Extraction Based on N-Gram Cover." WAIM2012 Workshops, LNCS 7419, pp. 29–38, 2012.

[14] WU, Jianmin and Mizuho IWAIHARA, “Revision graph extraction in Wikipedia based on supergram decomposition,” Proc. WikiSym 2013(OpenSym 2013), Hongkong, Aug. 2013.

[15] Zhu, Mingliang, Weiming Hu, and Ou Wu. “Topic detection and tracking for threaded discussion communities.” Proc. Web Intelligence and Intelligent Agent Technology(WI-IAT'08), 2008.
[16] Alexa, http://www.alexa.com/siteinfo/wikipedia.org
[17] Gibbs LDA(C++), http://gibbslda.sourceforge.net/
[18] Wikipedia, http://en.wikipedia.org/wiki/Wikipedia
[19] Wikipedia Editing, http://en.wikipedia.org/wiki/Wikipedia:How_to_edit_a_page
[20] Wikipedia edit history export pages, http://en.wikipedia.org/w/index.php?title=Special:Export&action=submit&pages=