BELEGARBEIT

Change Tracking in Wikipedia

Yihan Deng
dengyihan@aim.com
Matrikelnummer: 3376831
Bearbeitungszeitraum: 15.02. – 30.09.2011

Technische Universität Dresden, Fakultät Informatik
Institut für Systemarchitektur, Lehrstuhl Rechnernetze
Hochschullehrer: Prof. Dr. rer. nat. habil. Dr. h. c. Alexander Schill
Betreuer: Dipl.-Medien-Inf. Philipp Katz

Selbstständigkeitserklärung

Hiermit erkläre ich, Yihan Deng, dass die vorliegende Arbeit zum Thema „Change Tracking in Wikipedia“ selbständig von mir ohne die Hilfe Dritter und ausschließlich unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt wurde.

Dresden, 30. September 2011

Acknowledgements

- I offer my most sincere gratitude to Dipl.-Medien-Inf. Philipp Katz, the supervisor of my thesis, for his supervision, consideration, patience, support and dedication. I profoundly appreciate his valuable pieces of advices during discussions and commenting on the writing.
- I am heartily grateful to my parents and all the friends, who offer great favor in the thesis.

Contents

Chapter 1

Introduction

Wikipedia is an international project that uses wiki software to collaboratively create an encyclopaedia 1.1. At present it is the fifth most popular website[1] and contains more than 13 million articles in 271 languages[2]. Over the past ten years, Wikipedia has experienced tremendous success. Its pages are read by billions worldwide every month, with 395 million unique visitors in December 2010 alone representing 31.8% of the Internet users[2], however, less than 0.05% of these readers are actively contributing to its content[2]. The

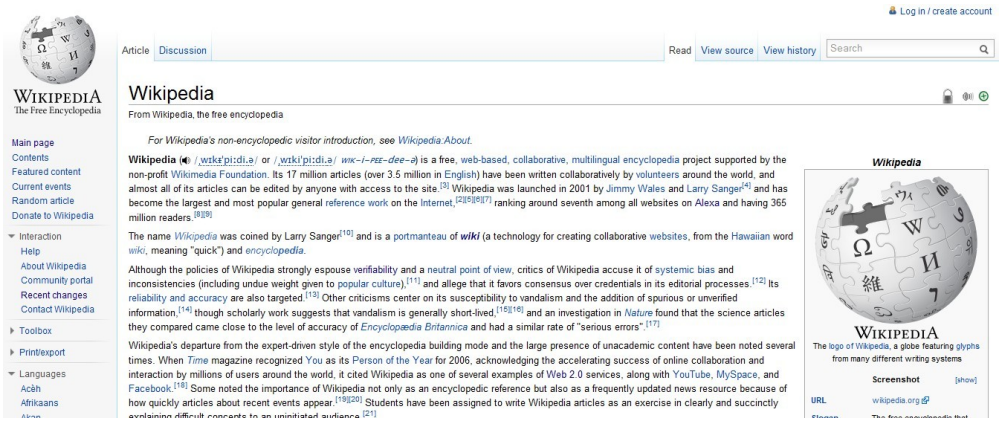


Figure 1.1: English Version of Wikipedia

concept of wiki was conceived of by Ward Cunningham, who implemented the first wiki engine and started the first wiki web called WikiWikiWeb in 1995.

[1] `http://alexa.com/siteinfo/wikipedia.org`, accessed on July 11, 2011

[2] `http://stats.wikimedia.org`, accessed on July 11, 2011

The name "Wiki" was, in fact, borrowed from Hawaiian term "wiki", which means "quick", "fast", or "to hasten". The antecedent of Wikipedia was the Nupedia project founded by Jimmy Wales, whose initial idea was to build an online encyclopedia licensed under the GNU Free Documentation License. In January 2001, Wikipedia was started as a side project to allow collaboration on articles' composition prior to the lengthy peer review process. Later it grew faster and attracted more authors than Nupedia which was closed in 2002. By now, Wikipedia has already 279 language versions, among which English, German and French versions are the top three for the number of articles. 204 language versions contain over 1,000 articles each, and 98 ones over 10,000 each. In June 2003 the Wikimedia Foundation was founded as an independent non-profit institution, which was also responsible for other wiki projects involving dictionary and thesaurus, collection of quotations, textbook, media, etc [Vos05].

The most important characteristics of Wikipedia are authorship and version control. Authorship means that any user of Wikipedia can become an editor, who has the freedom to edit every page in Wikipedia. Consequently every page of Wikipedia is collaboratively edited by a group of users, which is so-called "peer creativity". In other words, the users of Wikipedia can be both readers and editors. Version control means the user editing history is fully stored and the Wikipedia pages can be recovered to any previous revision. With version control mechanism, the users of Wikipedia are encouraged to verify the pages rather than be prevented from making mistakes.

For freedom and openness of editing, the quality of Wikipedia articles has always been in debate at the early stage. The negative opinions said that the trust system embedded in Wikipedia was primarily social and the quality of articles cannot be guaranteed, while the supporters proposed that the characteristics of Wikipedia make it easier to correct errors than to add malicious content or to delete content [VWD04]. The researchers of [KK08] proved that the collaborative updates by large numbers of contributors from different cultural backgrounds and knowledge structures resulted in higher quality and less biased articles, and the wide use and rapid development of the Internet has enabled aggregated judgments of Wikipedia knowledge much easier. Nowadays, most models of collective intelligence are also premised on aggregating the independent contributions of many people, which facilitates the collection of "the wisdom of crowds". According to the research result from *Nature* [Gil05] the scientific entries in Wikipedia are of equal quality compared with those in the more established *Encyclopædia Britannica*, which is widely regarded as highly reliable and trustworthy.

Besides being used as a web-based free-editing knowledgebase, generic wiki systems can also be used as a model for new ways of working in which people contact with each other bypassing hierarchies and collaborate over the organizational boundaries. This way of working is called swarm creativity which has been described in [Glo06] as a part of a Collaborative Innovation Network (COIN). In other words, it can be seen as a cyberteam of self-motivated people with collective vision, who collaborate in achieving a common goal by sharing ideas, information and work. In addition, a wiki system can also be employed as a management tool in enterprises, projects, or organisations. Futhermore it may also be applied to the field of education since it makes the communication and interaction between the teacher and students much easier.

1.1 Motivation

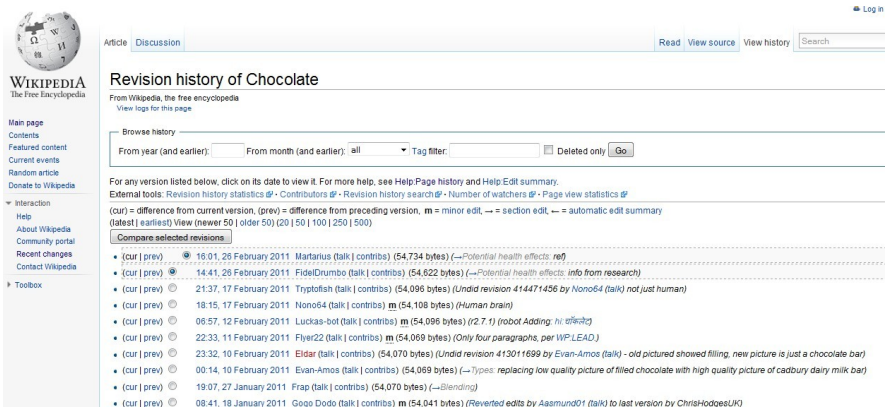


Figure 1.2: History View of Mediawiki

The history view of Mediawiki[3] is shown in Figure 1.2, it supports some basic interactions to explore the revision history of an article. There are navigation links to change the number of entries per page and travel backwards and forwards in an article's edit history, which supports a comparison of any two revisions of a Wikipedia page. The desired revision can be chosen via the radio button before each revision entry.

[3]MediaWiki is a free software open source wiki package written in PHP, originally for use on Wikipedia. `http://en.wikipedia.org/wiki/Wikimedia_Foundation`, accessed on July 7, 2011

Because of the unprecedented large scale of the Wikipedia system, it is impossible to get a clear insight into the change of wikis based on the revision history alone, especially when the changes are numerous and constant. In Wikipedia, all previous versions of each page are saved by the wiki software, which are named "revisions".

Normally the users of Wikipedia have access to all the previous versions in the "edit-history", which means all editors can trace the edit history and progress of each article. The history view typically visualize the differences (addition, deletion, or modification) between two distinct versions of an article by a "diff" function, so editors can see the difference and change of the text directly. Furthermore, a summary of each editing may also be provided, including the difference in the size of characters from the previous version, a short line of comments on the editing by the editor, etc. These comments give a brief explanation of the change and help other editors to notice the change in the new version without having to look into the text. However, the difference in the form of the number of characters does not reveal the details of editing, and comments on the changed text are not always furnished. Even where comments are provided, they may not be able to fully reflect the nature of editing.

The "edit-history" of the article in Wikipedia is normally displayed as a linear sequence of revisions in chronological order. The edit history points to the relationships between editors and patterns of user-behaviour under the wiki principles, the so called "five pillars" [Fpw], including which kind and how significant a change made is in the editing, which part and which type of the content has been changed, how many words or internal links have been added to which section, whether there are any format or template changes in comparison with the previous revision, and above all whether there are any semantical changes to the article and what the reasons are.

Furthermore, the real-world events have a direct influence on the editing of Wikipedia page. According to the topic detection and tracking (TDT) theory described in [All02], each page in Wikipedia can be seen as a topic and the events occurred in the entire "editing history" of this page can be regarded as the stories of this topic. Hence the "editing history" contains also the clues of editing events, which reflect the features and the patterns of the editing events.

However, with the swift growth in the volume of the "edit history", it becomes increasingly difficult to grasp the trend of the editor's actions with direct observation of the revision list and to manually check diff-views between revisions. Therefore an automatic change tracking system for Wikipedia to

extract the edit history and reflect semantic and event-related change between versions will be of great help to get a better insight into Wikipedia and to understand the significant change related to the revision content and the context.

1.2 Research Questions

The research questions that are addressed in this thesis are as follows:

- **What types of change and user actions exist in Wikipedia editing and how can these change be categorized and tracked?** Wikipedia articles undergo constant change from cooperative editing, yet these changes are mostly scattered in unordered user editing actions or cluttered with long-term editing history, which renders it difficult or nearly impossible for the readers to get an overview of these changes from numerous revisions without further processing, though a basic difference function has been provided by Wikimedia. So the first subject of this research is to analyze the change of Wikipedia , in other words, the change will be quantified, located and presented in corresponding categories. The goal is to help the users to get an easier understanding of the change between revisions along the edit history.

- **What features can be used to detect social events and how can the events be detected?** Each Wikipedia page presents a concept and there is some inner connection between these concepts such as semantical correlation, internal link, or categorical relationship. There are always some events, which lead to the editing of a group of related concepts. Editing activities in Wikipedia can thus be seen as the reflection of public interest and attention. The highly frequent editing and change in Wikipedia content indicate the change in public attention which is influenced by various social events. With the aim of detecting the editing event as well as social event, so as to understand the change, we should first answer several questions: What is the editing event? Which features can be used to detect events? Are there any relations between the change in features shared by related articles during the progression of editing event? What methods should be applied to detect the event based on the features and which one is the best?

- **How is the editing behaviour of users influenced by social events?** After the aforementioned questions are addressed, the social events reflected from editing events can conversely be detected by analyzing the event related features we extracted. Based on the results of event detection, the following two questions will be further discussed: what types of relations exist between user editing and social events? Are there any patterns for events-editing behaviour relations?

1.3 Structure of the Thesis

The structure of the thesis is listed as follows:
Chapter 2 describes the background knowledge and relevant theories which will be employed in our wiki change tracking approach. The state of the art of wiki research is also presented.
Chapter 3 introduces the concept and design of our approach in detail, including crawling of Wikipedia, revision modeling in Wikipedia, change visualization, event detection.
Chapter 4 describes implementation details according to the concept, such as the development toolkits, libraries and important APIs.
Chapter 5 evaluates the implemented prototype by showing the evaluation methodology and the comparison results of detection approaches with different detection factors and configurations.
Chapter 6 concludes the thesis by presenting the main contributions and further work that needs to be done.

Chapter 2

Background

This chapter provides a brief introduction to the important theories and technologies that are relevant to the realization of our change tracking and event detection approach. First of all, the definition of information extraction and the information extraction methods and data structure in Wikipedia are introduced. Then in Section 2.3 the difference functions will be described and compared, which are followed by the explanation and comparison of the methods for statistical comparison. Finally a state-of-the-art analysis of the current studies on Wikipedia change tracking is given.

2.1 General Information Extraction

In common information extraction senarios, the information extraction (IE) systems are used to extract domain-specific information from data in different structures and formats. The domain and types of the extracting target must be defined in advance. The main tasks of the IE systems are identifying targeted object and combining the targeted objects in a coherent framework, while ignoring the irrelevant objects in the domain [CL96]. In comparison with information retrieval, where the task mainly focuses on returning relevant information and ranked results to the keywords of users, information extraction pays more attention to the collection of information according to a predefined structure such as template or ontology. The input of an information extraction procedure is a data source like HTML pages, while the output is mostly populated database.

2.2 Information Extraction from Wikipedia

In order to extract the data from Wikipedia, the inner structure of Wikipedia and access methods to the data in Wikipedia should be analyzed. The following sections describe the Wiki Markup and the common structure of articles in Wikipedia as well as the methods of data extraction from Wikipedia.

2.2.1 Wiki Markup

Wiki Markup also called Wikitext language, is a lightweight markup language, which can be used to create pages in wiki websites like Wikipedia. This format makes the editing of content in wiki easier in comparison with other formats such as HTML. For example, the Wiki Markup in Wikipedia has a simple way of hyperlinking to other pages within the site by using "free links" annotation [[..]]. However, the ultimate purpose of Wiki Markup is to be transformed by the wiki software into HTML, which is interpretable by web browsers, Listing 2.1 shows an example of Wiki Markup.

```
== heading ==
this is a paragraph
* this is a list item
* this is another list item
[[Wikipedia|link to article]]
```

Listing 2.1: Wiki Markup Example

Some of the most commonly used Wiki Markups with their meanings and places of occurrence are listed in Table 2.1.

Wiki Markup is a kind of semi-structured data. There are not only various objects and structures in Wiki Markup but also large numbers of unstructured raw texts as content between Wiki Markups or templates. For this reason, the concrete structure of four important components in a Wikipedia page are shown as follows, each subcomponent is described with its possible usage according to the task of change tracking:

Article:

- First paragraph — Definitions

Markup	Meaning	Occurrence
[URI Text]	External link	Anywhere
[[Text]]	Internal link	Anywhere
[[File:name.jpg]]	Image	Anywhere
{{Text}}	Template	Anywhere
”’Text”’	Bold	Anywhere
”Text”	Italic	Anywhere
<ref>Text</ref>	Reference	Anywhere
* Text	Unordered list item	Beginning of line
# Text	Ordered list item	Beginning of line
– – – –	Horizontal Line	Beginning of line
== Text ==	Title (second level)	Beginning of line
=== Text ===	Title (third level)	Beginning of line

Table 2.1: Typical Wiki Markups [FBA10]

- Full text — Description of meaning; related terms; translations
- Redirects — Synonymy; spelling variations, misspellings; abbreviations
- Title — Named entities; domain specific terms or senses
- Section heading — Category suggestions;faceted support of the definition

Article links:

- Context — Related terms; co-occurrences
- Label — Synonyms; spelling variations; related terms
- Target — Link graph; related terms
- LinksTo — Category suggestion
- LinkedBy — Category suggestion

Categories:

- Category — Category suggestion
- Contained articles — Semantically related terms (siblings)

- Hierarchy — Hyponymic and meronymic relations between terms

Infobox Templates:

- Name
- Category suggestion; entity suggestion

Furthermore, there is a special markup in Wikipedia pages which is called template, it is usually used to describe the common features and repetitive content in a standard format 2.2. It is mainly used for warnings, notices and so called infoboxes as well as navigational boxes. For example, a commonly used template *infobox* which is roughly found in every third Wikipedia article [LBN10], summarizes the facts of a page concept in attribute-value form. A template offers a structured and consistent container for common information of Wikipedia pages, which facilitates the task of extracting structured data and transforming them into other formats. The general format of templates is represented in Listing 2.2:

```
{{ TemplateName
  | field1 = value1
  | field2 = value2
  | field3 = value3
}}
```

Listing 2.2: Template Example

2.2.2 Access to Wikipedia Data

There are several methods for extracting data from Wikimedia [Wma].

- **Wikimedia-API:** The first method is to use the Wikimedia-API, which offers the possibility of obtaining specific properties of an article such as different revisions and metadata in REST[1] style. The requests and

[1]Representational State Transfer (REST) is a software architecture, that controls all the API functions per HTTP request of an URL and the response will be returned in XML or JSON format. It is still based on the client/server architecture with a stateless

responses are built around the transfer of representations of resources and the resources in Wikipedia can be retrieved with a fixed identifier like address or URI of the pages. The Wikimedia-APIs thus provide a complete access mechanism to the data of important components in Wikipedia such as revisions, page titles, links, etc.

- **Special:Export:** The second method is to use "Special:Export" [Spe] interface, which allows the exporting of entire categories and bulk of content of Wikipedia pages by inputting full page name, and the required pages and revisions are returned as wiki text in a XML container. However, the exports of editing history are limited to 1,000 revisions.

- **Web Feeds (Atom, RSS):** The third way is to use the web feed, a web feed is a special data format used to make web content available for subscription. Various aspects of Wikipedia can be monitored with RSS or Atom feeds. For example, you can get a subscription of RSS feed to the recent changes of any Wikipedia article. It returns differences between the most recent revision of an article and its previous revision as well as the author name and the timestamp. The new pages and watch list also provide RSS feeds to be subscribed.

- **IRC Channel:** The fourth way to obtain the change information from Wikipedia is the IRC channel. IRC is a form of real-time Internet text messaging or synchronous conferencing. It is mainly designed for group communication in discussion forums. In Wikipedia IRC is employed to offer a "chat room" for Wikipedia users, in which the users can have a live chat with each other. IRC in Wikipedia also offers a channel for returning recent change so that the live change report via IRC in format of differences between revisions and information about the users as well as the timestamp can be received.

Because the entire edit history of each article is concerned in our task, using Web feeds, IRC or "Special:Export" is obviously not a good choice, since not only recent change report for revisions, authors and timestamp but also the original text of each revision are required for further natural language processing and analysis, so Wikimedia API is employed to access

communication protocol and every message contains all necessary context information and thus neither the server nor the client has to store the context data. In contrast to Remote Procedure Call (RPC), requests in a REST system are not directed to procedures but to resources (documents) using a generic interface with standard semantics.

the Wikimedia database, by which comprehensive information of Wikipedia pages and corresponding revisions can be obtained.

2.3 Difference Function

"Diff" is so-called difference function, which provides the function of calculating the differences between two revisions, so that we can know what has been changed, and where exactly the changes take place. There are mainly two types of diff algorithms: longest common subsequence (LCS) and greedy matching.

- **Longest Common Subsequence:**
  The unix diff utility was developed in the early 1970s, which was implemented based on a text differencing method called "longest common subsequence". It compares the differences between the old and new versions line by line with an output report in terms of insertion, deletion and replacement. Mediawiki uses this to show the differences of wiki texts. Google-diff-match-patch libraries [Gom] provide a powerful algorithm to perform the operation for plain text comparison. It implements the Myer's diff algorithm, which is also a longest common subsequence algorithm, with a layer of pre-diff speedups and post-diff cleanups to improve both the performance and the output quality. Google-diff-match-patch libraries offer diff methods for different levels of comparison. For example, the diff is normally operated under character mode, but it can also be shifted to word mode or sentence mode or semantic diff. It is able to compare two blocks of plain text and return a list of differences.

- **Greedy Matching:**
  There are also diff tools which are based on other algorithms such as greedy matching algorithm or method which analyzes corresponding control flow graphs of the original and modified versions of a program [AOH04]. Greedy matching method does comparison between the old and new versions in chunks of sentences or tokens, and unmatched parts in the old version can be tagged as deletion and unmatched parts in the new version can be seen as addition. In the control flow graphs program changes can be mapped into clusters, which are single entry, single exit parts of code. Then clusters can be reduced to single nodes in the two

graphs. Afterwards these nodes are recursively expanded and matched to calculate the differences.

The change tracking and the event detection are main focuses of the thesis, so after parsing each part of page out of the revised raw text, the diff process should be conducted so that the difference and change in paragraphs and sections from the page can be determined. To this end, the performance and quality of the output is the most desirable features of diff function. The details of diff function will be explained in Chapter 3.

2.4 Statistical Comparison: Goodness-of-Fit Tests

Statistical comparison plays an important role in all kinds of experiments [PMD+05], which facilitates the distribution comparison and analysis. The typical use cases of the statistical comparison are: the comparison of data from different sources with respect to theoretical distributions; the comparison of excepted distributions with reconstructed distributions; the calculation of the distance of two distributions with the aim of change determination and regression test. Either binned or unbinned data can be used as test sample in the statistical comparison.

- χ^2 **(Chi-squared) Test :** It is an important method to quantify the measure of the deviation between two distributions. χ^2 test is mainly used to describe discrete distributions. It can also be used to measure the deviation between unbinned data. As can be seen in Formulation (1), if B represents the number of bins, the statistics X^2 is

$$X^2 = \sum_{i=1}^{B} \frac{(O_i - E_i)^2}{E_i} \tag{1}$$

  where O_i is the observed frequency and E_i is the expected frequency.

- **Kolmogorov-Smirnov Test :** Another important test method which is based on the Kolmogorov Empirical Distribution Function (EDF) definition is the Kolmogorov-Smirnov test. Similar tests include: Goodman test and Kuiper test. The test returns the statistics of linear

function of the maximum vertical distance between the EDFs of two distributions. These tests can only be applied to continuous distributions. With the order statistics $X_{(1)} < X_{(2)} < \cdots < X_{(n)}$ and EDF $F_n(x) = (1/n)\sum_{i=1}^{n} I(X_i \leq x)$, where $I(\cdot)$ is the indicator function, so the cumulative distribution function of Kolmogrov-Smirnov statistic is represented in Formulation (2).

$$D_n = \sup_x \left| F_n(x) - F(x) \right| \tag{2}$$

- **Anderson-Darling Test:** The Anderson-Darling test measures the integrated quadratic deviation of two EDFs, as shown in Formulation (3), which is suitably weighted by a weighting function $\psi_{AD}(F(x))$. If the weighteing function is defined as Formulation (4), then it is called the Cramer–von Mises test, which can be used on unbinned data. The Formulation (5) defines the weighting function of Anderson-Darling test, which can be performed on both binned and unbinned data.

$$Q_n = n \int_{-\infty}^{\infty} [F(x) - F_n(x)]^2 \psi(F(x)) \mathrm{d}F(x) \tag{3}$$

$$\psi_{CvM}(F(x)) = 1 \tag{4}$$

$$\psi_{AD}(F(x)) = [F(x)(1 - F(x))]^{-1} \tag{5}$$

In general, the χ^2 test, for its simplicity, is the least powerful one because of the information loss in the data grouping (binning). On the other hand, the tests which are based on the supremum statistics are more powerful than the χ^2 one, focusing only on the maximum deviation between two EDFs [PMD+05]. The most powerful tests are the ones containing a weighting function like Anderson-Darling tests, which places higher weight on observations in the tails of the distribution rather than only measures the general deviation of two distributions.

2.5 State of the Art

At present there are several different research tracks related to Wikipedia and wiki projects[Sot09], the four hot spots of research which are related to the thesis are described as follows:

1. As far as the research focus is concerned, quantitative analysis is obviously one of the most desirable choices. It is suitable to model the Wikipedia system behavior, the activity patterns of authors and the evolution of contents over the time. Most of the research create quantitative models based on the system log files by using statistical methods and data mining techniques.

2. The second research topic which is also attracting the attention of many researchers is measuring the quality of contents of the wiki system. The interactive nature and highly dynamic evolution of the wiki content cause many problems with respect to quality and trustiness, and therefore automatic system which could assess the quality of contents in the wiki systems should be built.

3. The third theme, which should also be emphasized is the social networks, the web graphs and the links models. As collaborative systems, wikis are naturally studied for social networking. The problems like behavioral patterns and the structure of the linked data repository provide opportunities for researchers.

4. The last research track is how to reconstruct the wiki system contents in a more efficient and machine understandable manner. It should allow users to retrieve information in semantic ways. This research line also consists of semantic tagging and annotation of Wikipedia contents, organization of contents according to users' ratings. A further introduction to the DBpedia as an example for information extraction and semantic usage of Wikipedia will be presented in the following sections.

In order to understand the change in Wikipedia, the quantitative analysis is naturally the first choice, so several typical examples of change visualization in Wikipedia are summarized in Section 2.5.1, the methods of change quantization are described and compared. As next, the methods and implementations against vandalism in the editing of Wikipedia are introduced in Section 2.5.2. In the following section, a special method to detect personalized event based on link graph and Wikipedia page view statistics is presented in Section 2.5.3, which offers the approach to find the event related pages according to the link graph of the input concept page. Meanwhile, the view statistics of event related pages are used to confirm the detection results. At last, DBpedia is introduced as typical paradigm of Wikipedia crawling and reconstruction in

Section 2.5.4, the crawling methods and the live extraction extension offer a good example of information extraction based on structure of Wikipedia page and semantic usage of linked data.

2.5.1 Visualizing the Change of Wikipedia

Due to the large scale of Wikipedia it is difficult to directly find out the inherent laws of this information as well as the relations between the datasets. For this reason, many researchers have offered their visualization mechanisms to facilitate the analysis of the changes in Wikipedia, especially for exposing the hidden patterns in large and complex datasets.

History Flow

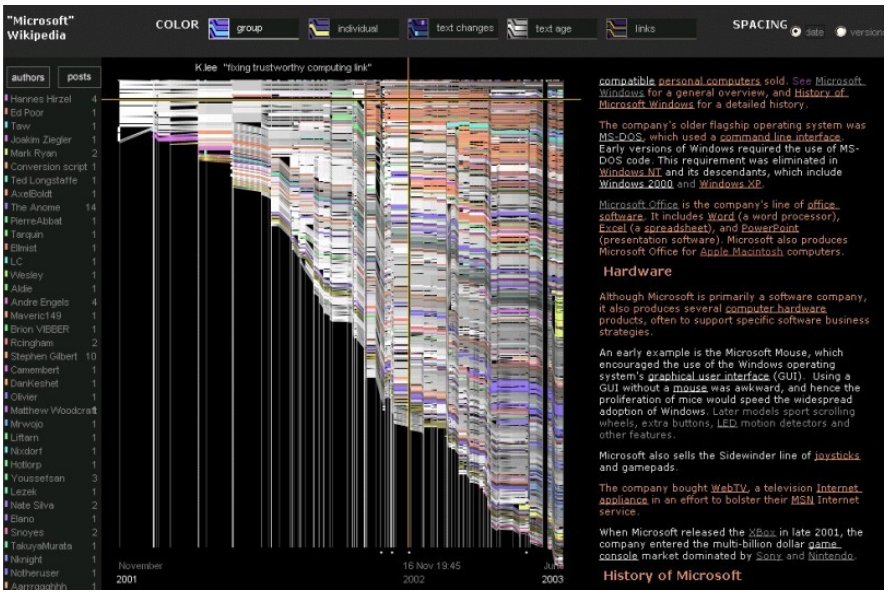


Figure 2.1: History Flow for Wikipedia Revisions [VWD04]

The authors of [VWD04] developed a classic visualization tool for Wikipedia which is named History Flow shown in Figure 2.1. It visualizes the revision history of an article that shows addition, deletion and replacement of the contents, updating time, authors and lifetime of the contents.
The vertical lines in different colors represent each revision of the document which are edited by different authors. The length of the line indicates

the amount of the text created by the corresponding editors. If an editor has modified the content, the change would be presented by elongating or shortening the vertical lines. Every saved line stands for a new revision, the history flow is the line which connects the same unchanged text fragments between consecutive revisions. The text fragments, which are inserted or deleted, will not get connected. So the gaps between revisions represent deletions and insertions of certain text fragments. To find the difference between two revisions, History Flow has employed a simple algorithm based on longest common subsequence [Hec78], which provides a sentence level granularity. Furthermore, the history flow provides four modes of views with different focuses, namely, *Community View*, *Individual Author View*, *Recent Changes View*, and *Age View* [Hgp]. On the basis of these data, the researchers have identified multiple updating patterns of Wikipedia articles. For instance, an URL of a Wikipedia page is inputted to the History Flow to visualize the evolution of this page's contents, and a wiki page will be presented for the analysis of a Wikipedia page's evolution. History Flow has revealed some of the metrics and patterns of Wikipedia, e.g., the variety of negotiation processes in reaching consensus, by which the surprisingly effective "self-healing" capability based on collaborative page editing is clearly detected. Furthermore, the diversity of authorship and the busty rhythms of page editing are also recognized.

JwikiVis

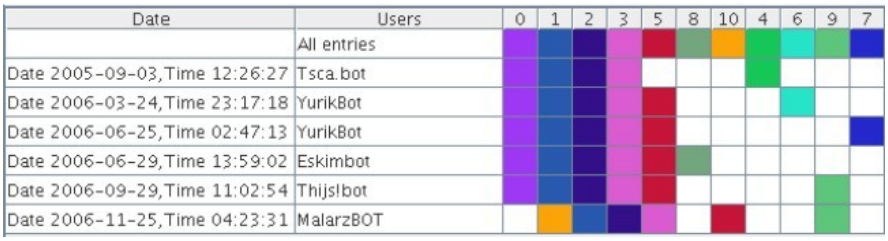


Figure 2.2: JWikiVis in 2D [GG07]

In [GG07] presents a desktop visualization software called JwikiVis. It helps to understand how collaborative documents are created and how they evolve over time. This software was inspired by History Flow and has similar visualization functions. Due to the text structure of Wikipedia articles, a text is described as a string of characters modified at some intervals, which

renders the visualization in two dimensions shown in Figure 2.2. So JwikiVis 2D illustrates the text structure on one axis and the flow of time on the second axis. Every rectangle in the 2D visualization stands for a part of a text, for instance, a paragraph or a single line. The first two columns on the left handside give the timestamps and the editors of the text and the rest columns show the parts of the text at a certain point of time. So the rows of the table represents the revisions and the columns of the table stands for the positions of each part in the text. If a new part is added to the revision, it occupies the corresponding position and the following parts will keep the correct order of the whole text. Based on the visualization, the problems of part durablity and frequency patterns have been presented and discussed: parts durability means the duration of existence for certain part of a article, the authors indicate that the part with high durability shows more significance to the whole page whereas the high updating frequency of the page reveals the strong likelihood of occurrence of important facts concerning people's lives.

WikiDashboard

The above-mentioned approaches are for external visualizations, which are developed in an independent program separated from the Wikipedia. It seems that visualizations implemented as extensions of the existing interfaces will be better applied in live settings and are more valuable to the wiki communities; The authors of [SCKP08] have offered a powerful tool called WikiDashboard which "aims to improve social transparency and accountability on Wikipedia articles". It extends the Wikipedia interface by adding information about the authors who have edited the page and the recent editing frequency of the page. WikiDashboard presented in Figure 2.3 is a transparent layer, which is placed on every page of Wikipedia. It presents visual statistical results about the editors as well as the edit frequency of each Wikipedia page. WikiDashboard offers the possibility to grasp the interactive activities and the collaborative patterns of Wikipedia pages. Especially noteworthy is its aggregate edit activity graph which represents the weekly edit trend of the analyzed article. A list of the top active editors and their editing records for that page can also be obtained, by which the page can be edited with more awareness, so that the page quality can be improved. Meanwhile, the editor activities are traced, which provides the possibilities for further research on these records of user actions.

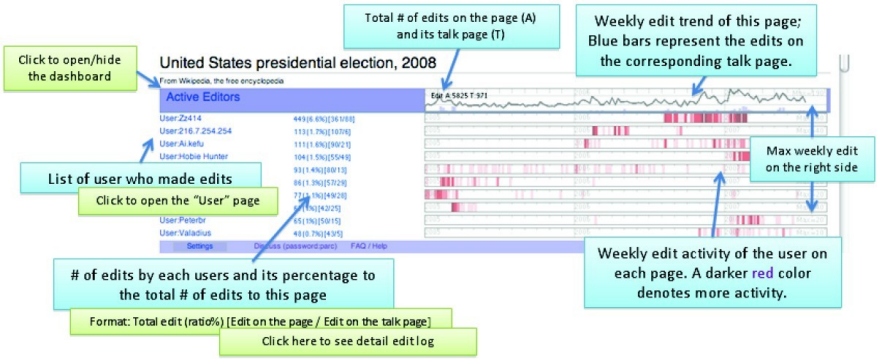


Figure 2.3: Wiki Dashboard [SCKP08]

Wikichange

Figure 2.4: Embedded WikiChanges [NRD08]

In [NRD08], the authors provide another tool named WikiChanges shown in Figure 2.4, which illustrates the complete revision history for each article. This tool helps to plot the revision activity in a timeline. The y-axis of the graph shows the update amount whereas The x-axis represents the chronological line, which exposes the complete edit history of articles in an easily understandable format. It is especially easier to spot recurring trends of edit in articles. The web browser based Wikichanges and is embedded in the real Wikipedia pages, has also been provided . It is noteworthy that, the author has built the revision summaries for a given period based on the calculation of new inserted terms between revisions. Since only the oldest and newest revision of a given period are considered, the algorithm can therefore keep a complexity under $O(N \cdot \log(N))$, however it will lead to loss of important details, if the extraction period is not correctly defined and the term related to developing process of the event can not be detected by this type of tagging.

Wikirage

WikiRage[2] is a website which presents statistics on wiki editings for entries tracking in Wikipedia, which shows the ranking list of most popular pages and editors with most updatings. The change is captured from recent change page provided by Wikimedia through IRC channel, and robots or minor edits are ignored in the capturing process. The sample rate fluctuates between 5 and 10 minutes. Every obtained entry will be accessed to retrieve more meta information. All the actions are stored in a database. It can be seen as a tool for showing current hot topics and the most passionate editors in Wikipedia, since it lists out the most edits per editor over various time periods after statistical analyses have been performed on the basis of the collection of the edit timestamps.

2.5.2 Combating Vandalism

From the above mentioned tools and mechanisms, it can be seen that plenty of researches have been done on the collaborations and conflicts in Wikipedia. Collaboration is an iterative, creative process where two or more people or organizations work together to realize shared goals by sharing knowledge, learning and building consensuses. Due to the full freedom and openness of Wikipedia, the Wikipedia editors will not always make perfect companions for each other. It is inevitable that there exist many vandalism situations in edits in Wikipedia. The authors of [CSSE10] said, "Vandalism is defined as malicious editing intended to compromise the integrity of the content of articles". Examples of vandalism are, insertion of unrelated text to the topic pages, mess deletion, insertion of slurs and vulgarities, offensive copy, etc. Combating vandalism by manual efforts is labor intensive, therefore, researchers have been looking for automated approaches.

Vandalism Detection with Active Learning and Statistical Language Models

The prevalent anti-vandalism methods are mostly based on rule-based metrics and machine learning techniques [WIPR10]. The anti-vandalism system, which uses rule-based metrics, keeps a list of static rules, for example, in the

[2] `http://www.wikirage.com`, retrieved on July 9, 2011.

Figure 2.5: Vandalism [Van]

form of regular expressions. A databases of blocked blacklist of saboteurs can also be established for vandalistic phrase examination. Because the static rules lack flexibility, the efficiency of the rule-based systems is relatively low. According to [SGV08], rule-based system can only detect 30 % of all vandalisms. Consequently different kinds of machine learning mechanism are applied to improve the anti-vandalism effect.
In [CSSE10] an active learning approach using the features of statistics language model to classify and rank the potential instances of vandalism is proposed. A comprehensive taxonomy of Wikipedia actions and the corresponding categorization of vandalism are presented based on features obtained from the statistical language model of revisions. A supervised active learning model presented in Figure 2.6 is empolyed to facilitate the annotation of training set. The annotation is started with annotated Weimar data provided by Potthast et al.[PSG08], the revision history are divided into five partitions chronologicallly, which are used for five iterations of annotation process. At first partition a classifier is built to rank the result, the top 50 results of each iteration are added to the training pool incrementally. After annotation, two classifiers (logistic regression and SVMs) are deployed to evaluate the detection effectiveness. As can be seen from the experiment results, the average precision of the both classifiers has increased after learning iteration and the two classifiers show different effectiveness on different types of the vandalism.

This paper has provided a good example of building statistical language model for the revision history of Wikipedia articles and the methodology of classification of editing action and vandalism provides also inspiration for further quantitative analysis based on machine learning approach.

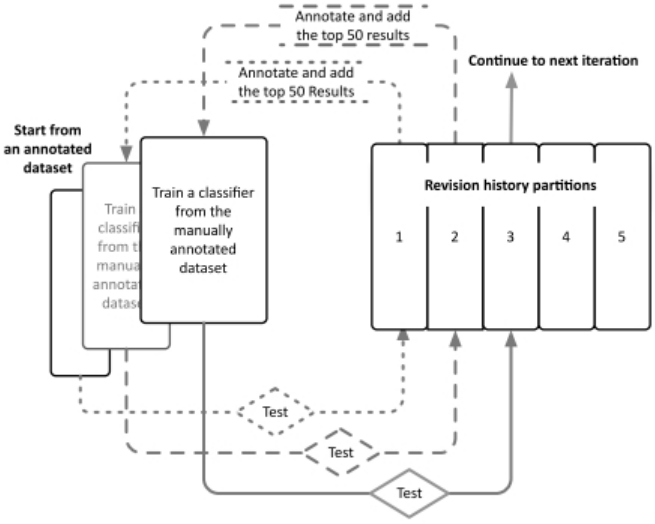


Figure 2.6: Active Learning Model [CSSE10]

Vandalism Detection with Spatio-Temporal Analysis

[WKL10] developed STiki as an anti-vandalism tool for Wikipedia. The unique feature of this tool is that it does not rely on NLP over the article or the diff text to locate vandalism, but leverages spatio-temporal properties of the revision metadata. The so-called "temporal properties are the function of time, at which an event occurs", and the spatial properties are the geographical distance or membership function of the related events. Compared to the previous NLP efforts, STiki is more efficient, robust to evasion and language independent. It performs a real-time anti-vandalism procedure and consists of a server-side processing engine which checks each of the probable revisions by subscribing to the IRC channel for the information of Wikipedia edits. Aided by meta data, it makes a decision regarding whether an edit is vandalism. Then the client-side GUI presents likely vandalism to end-users in the corresponding classification. The users can also provide feedback through client to aid the decision making in server side .

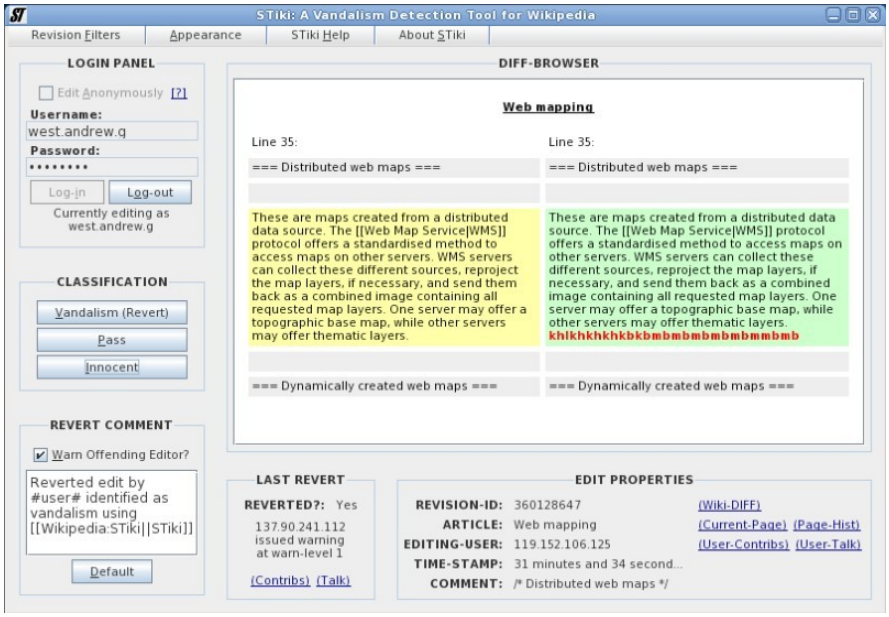


Figure 2.7: STiki Anti-Vandalism Tool [WKL10]

The anti-vandalism methods can be seen as a deep quantitative analysis on revision history, so the significance of analyzing on anti-vandalism methods lies in the enlightenment of features choosing and modeling of revision for extreme change detection, because both editing event and vandalism can lead to significant change in page content, which is also reflected by similar features in wikipedia pages. Therefore the methodology of vandalism detection provides an appropriate starting point for our event detection.

2.5.3 Personalized Event Detection with Wikipedia Link Graph and View Statistics

Wikipop [CN10] uses the Wikipedia link graph and page view statistics (number of visits per day) to detect trends in public interest. The application detects the most popular concepts which are related to the given keyword, the detection is conducted in two steps. The keyword related concept (Wikipedia page) is first inputted, then the internal links of this concept page are extracted. In second step, the returned concept are ranked based on the weighting methods based on link graph and view statistics. Each Wikipedia page describes a concept and internal links define the relations between concepts,

these pages are organized in the category system as well. Based on this structure a weighted link graph for pages in Wikipedia can be established, and activation vector spreading algorithm is employed to discover the link graph. Because Wikipedia pages have a very small link graph distance with each other, some hub pages with general topics will connect to a large number of pages, which negatively impacts on the accuracy of the link spreading. So two weighting methods are introduced to solve this problem: The first one is ISR (Indegree Square Ratio) which uses indegree ratio of two pages to describe the semantic relatedness between them, which overcomes the problem of the wide spread of the activation through the hub nodes: let $indegree(i)$ be the indegree of the node i, and ISR can be defined as follows:
if

$$indegree(i) > indegree(j)$$

then

$$ISR(e(i,j)) = 1$$

otherwise

$$ISR(e(i,j)) = \frac{indegree(i)^2}{indegree(j)^2}$$

Another method is called PDC (Popularity Development Correlation) which observes the semantic relatedness between articles with the help of view statistics in the period of peeks popularity. For a link e(i, j) between concepts i and j, the Pearson correlation coefficient for the values of page views of i and j for 5 days around the peak popularity is used to compute PDC, which means if Pearson correlation < 0, PDC is zero. So for link $e(i, j)$ between concepts i and j the weight of the edge can be represented as

$$W(e(i,j)) = ISR(e(i,j)) \cdot PDC(e(i,j))$$

The most related concepts with high ranking score are returned, which facilitates the grasping of recent hot spots of topics according to the inputted keyword.

2.5.4 Information Extraction of DBpedia

DBpedia is a project developed by the researchers at the Free University of Berlin and the University of Leipzig, in cooperation with Open Link Software. It aims to extract structural content from Wikipedia. It offers the possibility of querying relationships and properties associated with Wikipedia and other

knowledge resources with data presented in the form of RDF. DBpedia has been described by Tim Berners-Lee as one of the most famous projects of the Linked Data project [Dbp].
Wikipedia articles generated by Wikimedia software, are mostly free texts, with some types of structured information in the form of Wiki Markup. Such structured information includes Infobox templates, categorization information, images, geo-coordinates, links to external Web pages, disambiguation pages, redirects between pages, and links across different language editions of Wikipedia. The DBpedia uses a flexible and extensible framework to extract all these different kinds of structural information by Wikipedia, focusing extractors, such as Label Extractor, Mapping Extractor, Infobox Extractor, Wiki Page Extractor, etc.
The data in Wikipedia is stored in the template of wiki format, for instance, the Infobox, which presents a summary of some shared facets among articles and makes navigation to other interrelated articles easier. Generally speaking, a template is similar to a function: it receives parameters that can be viewed as values of attributes, and it has a well-defined return value, namely the Wikipedia source text. The template attributes present information about instances of a specific concept, and the template's return value contains the source text which is necessary to display the box and its content in a table form. However, the heavy-weight extraction process of DBpedia has been a drawback. It requires manual effort to produce new release and the extracted information is not up-to-date.
The authors of [HSLA09] extended DBpedia with a live extraction framework shown in 2.8, which is capable of processing a huge amount of changes per day in order to consume the constant stream of Wikipedia updates in the form of Wikipedia OAI-PMH[3] live feed, so that DBpedia can be kept highly topical without manual updates. This also allows direct modification of the knowledge base and closer interaction between the users and DBpedia. The researchers also introduced that the Wikipedia community itself is now able to take part in the DBpedia ontology engineering process. In general, this framework offers an interactive circulation between Wikipedia and DBpedia and makes DBpedia more maintainable and robust.

[3]The Open Archives Initiative (OAI) is an attempt to build a "low-barrier interoperability framework" for archives (institutional repositories) containing digital content (digital libraries).

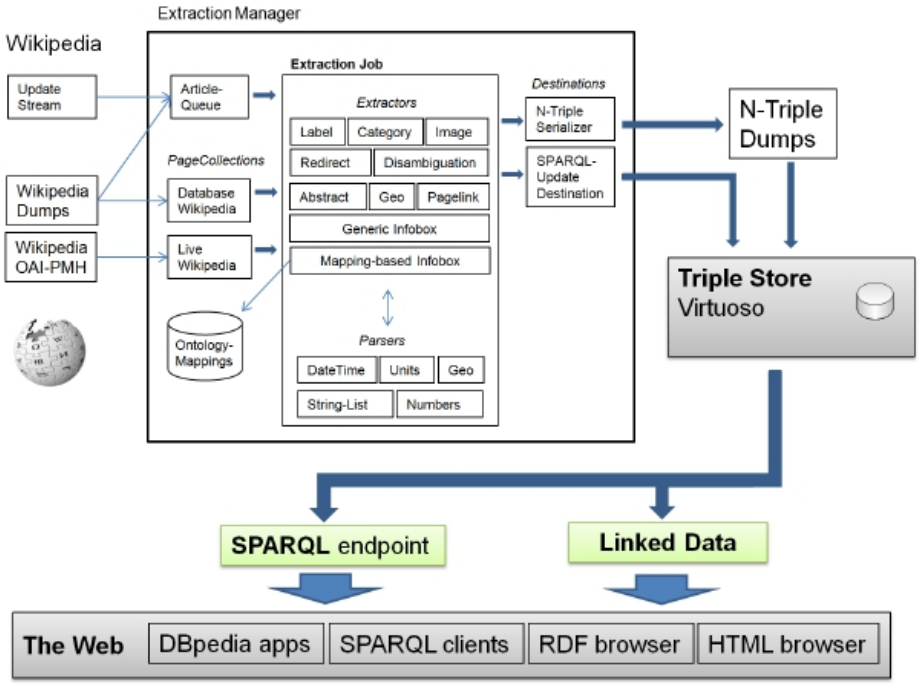


Figure 2.8: Live Extraction Framework Extension Based on DBpedia Extraction Structure [HSLA09]

2.5.5 Summary

The visualization of change in Wikipedia history is the starting point of Wikipedia research, the researchers have used different methods to illustrate the change at different levels. The change of paragraph, updating frequency and the corresponding editing statistics for user have been quantized and illustrated in detail. However, the visualization is mainly based on the editing history provided by Wikimedia and simple diff result between revisions. In recent years, the research of Wikipedia has focused more deeply on the text content and inner structure of the revision, NLP and machine learning approach as well as link based approach are used to explore the change and activities. The detection of vandalism in Wikipedia is one of the hotspots on current research of Wikipedia. Vandalism is a form of destructive activities in Wikipedia, which can be all types of user actions, such as deletion, addition, reversion, or replacement, etc. The anti-vandalism method based on active learning approach and rule based location and time features have been presented. The approach implemented in Wikipop facilitates the detection of

personalized topics, which shows another perspective of usage of internal link features as well as evaluation method for event detection in Wikipedia.
What we can learn from the above-mentioned methods is a common way to detect and trace the extreme change in Wikipedia, which starts with revision history accessing, followed by choosing and extracting features for the target activites and speical type of change, and ends with a comparison of features between the revisions. The most important thing for change tracking and event detection here is the choosing of features. We may choose some text based features or some spatio-temporal based features to observe the changes of Wikipedia from different angles of view. Thereafter the changes and the corresponding activities as well as the events and patterns can be determined. The features choosing and the revision modeling method of in our concept will be explained in the next chapter.

Chapter 3

Concept

In this chapter the basic concept and ideas of Wikipedia change-tracking system will be introduced. First the basic architecture of the system is presented, and then a brief explanation to each important component of the system will be given. After that, the flowchart of the change-tracking processing is presented. For an insight into the change details, which involve not only text content modification but also the alteration in the text structure and layout, some features are chosen as the indexes for change detection between revisions. The features are classified into two types, namely implicit and explicit features. At last the change tracking and events detection based on these chosen features will be explained. More specifically speaking, first, the revisions are compared to identify and measure the change in predefined metrics, such as in the the numbers of words and paragraphs, so that the degree of change between the revisions can be determined, and the type of change will also be categorized; second, the event-related features of the revisions are further extracted, the content distance is measured through the comparison of term frequency distribution, while the change of time interval and link intensity between revisions are also calculated; third, the approach of event detection will be presented.

3.1 Basic Architecture

The basic architecture of the system is presented in Figure 3.1. The *wikiparser* is the basic component of our system, which transforms the Wiki Markup text

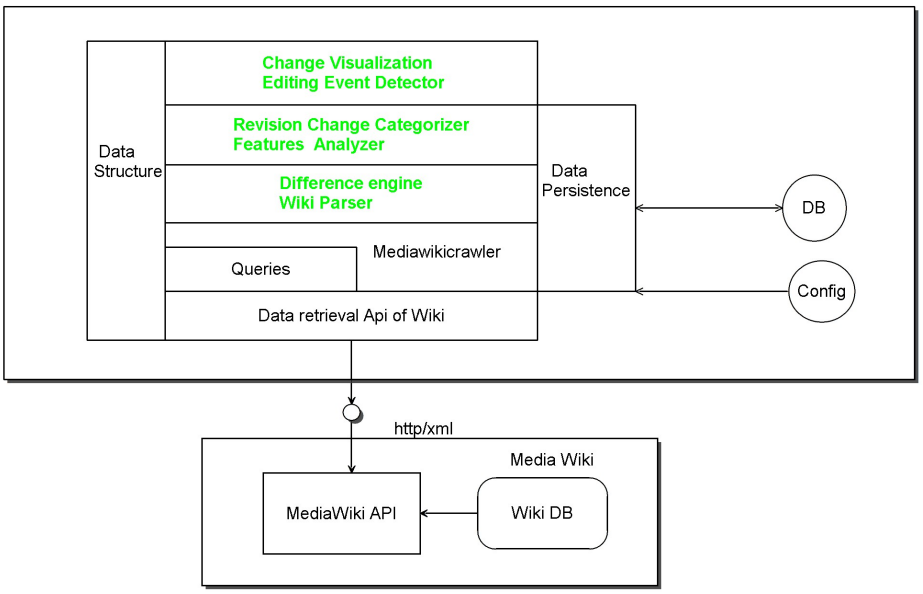


Figure 3.1: General Structure of Wiki Event Analyzer

into the desired structure and returns the text content in different size levels such as the whole page, section, paragraph. The diff function is designed to return the details of the changes in different actions including insertion, reversion and deletion based on the parsed results, and then the editing actions of each revision can be categorized. Meanwhile the feature analyzer will extract the predefined explicit and implicit features to identify the change between revisions. For the implicit features, namely the key phrase change in plain text, the corresponding processing of NLP will be conducted, and a distribution of frequency statistics for key phrases in each revision can then be established so that the newly added key phrases will be detected and the corresponding editing event can consequently be identified. On the uppermost layer of our system is a change visualization and event detector, which is responsible for change illustration and event detection based on our detection algorithm. Figure 3.2 shows the procedure of our prototype. First of all, the Wiki Markup text is parsed to obtain the desired text snippet and meta information from revisions. Then the diff function compares the given snippets to return an intermediate result of plain text differences. Next, the defined explicit and implicit features are analyzed to represent the changes between revisions. And, at last the analytical results of the features and timestamps are employed to perform the event detection process.

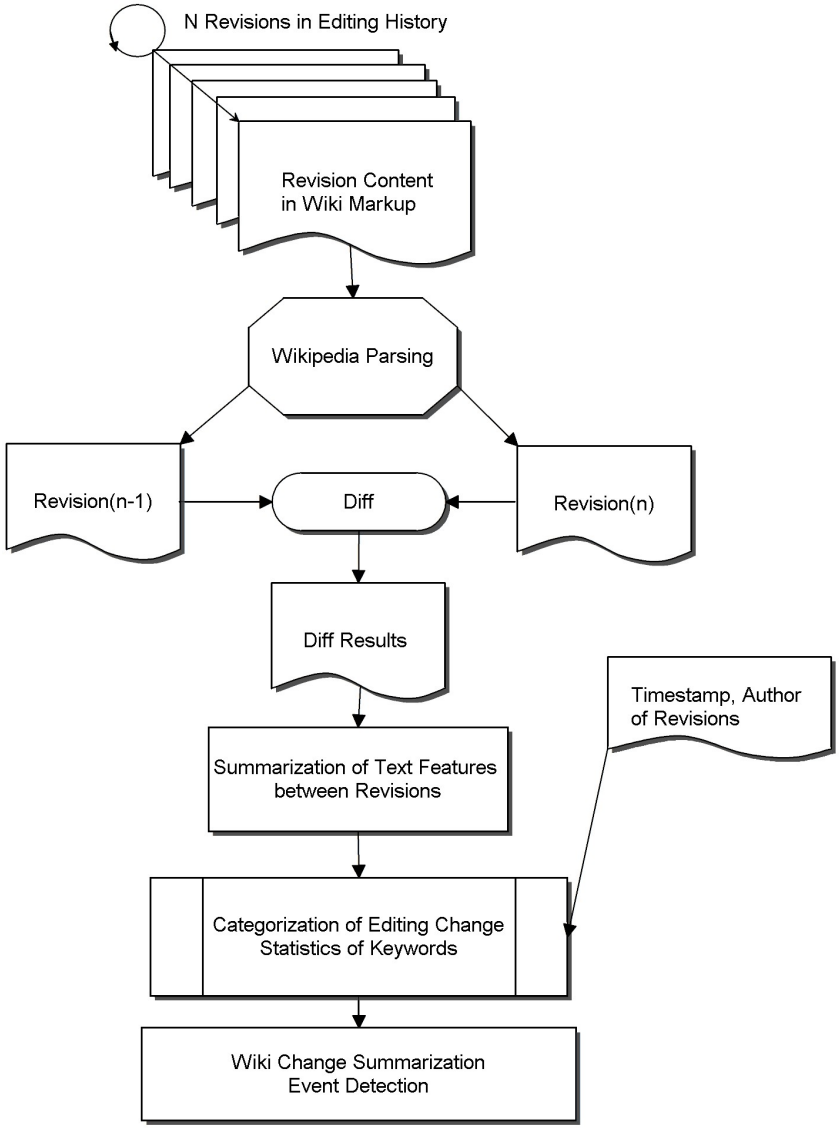


Figure 3.2: Flow Chart of Change Tracking and Event Detection

3.2 Parsing of Wikipedia Revision Data

Parsing namely syntactic analysis, is the process to analyze a text, which is composed by a sequence of tokens (for example, words), to determine its grammatical structure with respect to a given (more or less) formal grammar. It then returns the wanted part of the text in a defined format. With the purpose of making the change in each revision easier to understand, i.e getting an insight into the change in each part of the wiki text at different levels, a suitable parser for wiki text should be employed.
As shown in Figure 3.3, the basic structure of a Wikipedia article was marked with different Wiki Markups. As already explained in Section 2.2.1, a common page in Wikipedia is normally constituted of a hierachy of basic elements such as sections, paragraphs, internal links and categories. The key parts of the Wikipedia page for our analysis include the entire text content for implicit features extraction as well as the internal links of one revision for explicit features extraction. So our parsing process is to obtain the arbitrary size of wiki text, the implicit and explicit features. The former is for the difference process, whereas the latter two features will be used for editing event detection.

3.3 Features in Wikipedia Revision

The features, which are extracted from Wikipedia's page revisions, can mainly be divided into two types: explicit and implicit features. Explicit features stand for the internal links or connections between Wikipedia articles, that is the references and content relations between articles, while implicit features represent the statistic change in the terms and natural language related metrics in traced text content. Because implicit features cannot be observed directly, therefore the features of this type should be extracted from the text in each revision to reveal the possible concealed relations.

3.3.1 Explicit Features from Wiki Text

- **Content**
  For an article, the content features mainly involve words and sentences or text level metrics. They can clearly manifest the content change, and

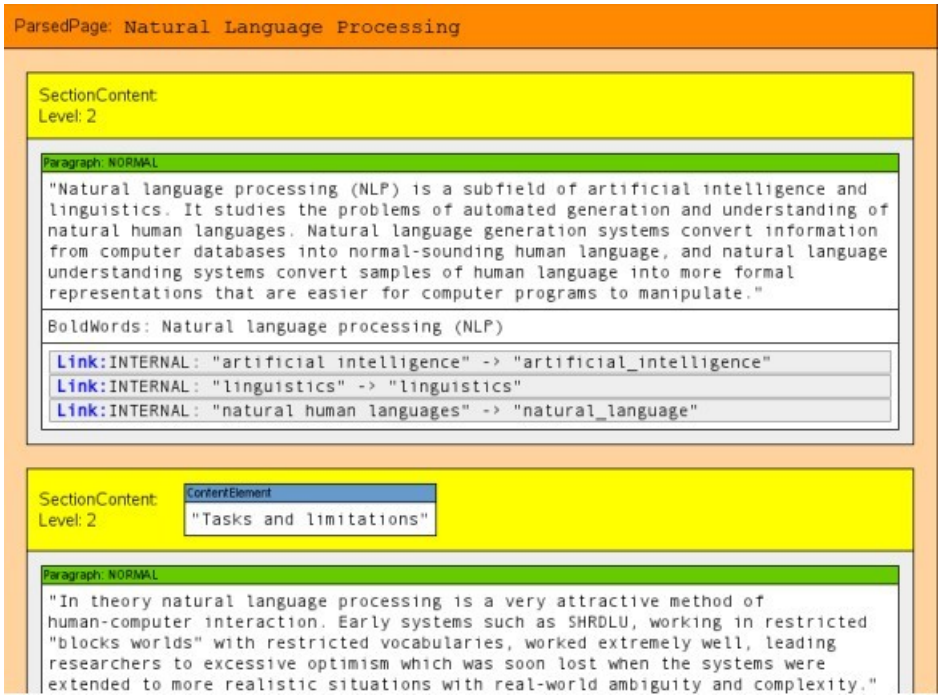


Figure 3.3: Parsed Wikipedia Structure [ZMG08]

are the most direct indicators of change.

- **Word Number(word_num)**
  Under this feature, there are following sub-features for different levels of blocks in one article.
  Total word number of an article;
  Word number of paragraphs;
  Word number of sections;
  These parameters can describe the changes well at different levels of components. For example, a corresponding report can be generated not only for sections or paragraphs but also for the whole Wikipedia article. Word number can be seen as a basic index of edit change, because all the change can be directly reflected by the change of word number and the basic user actions such as addition and deletion can be also identified by change of word number.
- **Text Content**
  After parsing the wiki article into sections and paragraphs, each part of the text can be obtained on demand. The change at the section level between two revisions can be firstly detected.

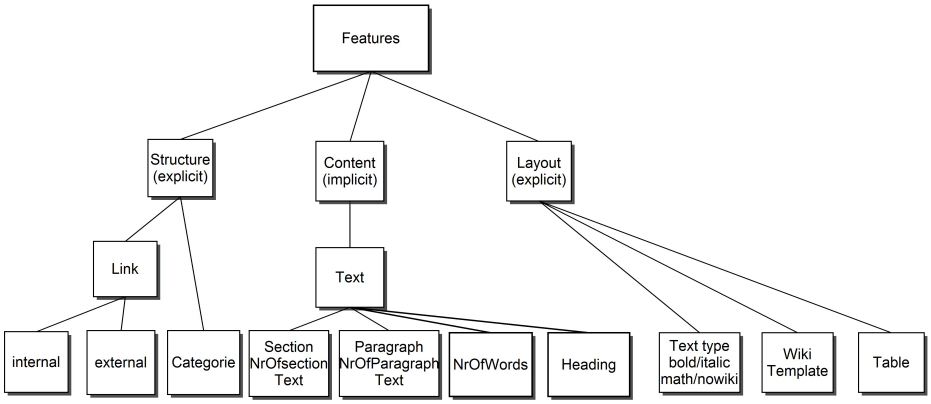


Figure 3.4: Features in Wikipedia Revision

The sections that cannot be found in the previous revision will be marked as addition, while the sections that appeared in the previous revision but cannot be found in the current one will be marked as deletion. If the change at the section level are detected, the text content in the section could be returned. A further exploration might also be done in the text content for the analysis of implicit features, which will be explained in the following section.

- **Layout**
  - **Templates**
    The wiki templates have already been explained in detail in the previous chapter. It is a special structure in the wiki texts, representing the common point for a group of wikipages in a well formed structure. It shows the change of a Wikipedia article from a distinctive perspective. As we know all the pages in Wikipedia have undergone an evolution process from simpler structure and semantic meaning to more complex ones, the appearance of templates is one of the sign of this evolution process.

- **Structure**
  If the linkage analysis of Wikipedia articles needs to be conducted, the structural features should first be extracted to figure out the physical connections between articles and to identify their hierarchy of categories.
  - **Link Number(link_num) Link Target Name(page title)**
    There are two basic features can be used to indicate the link change

in Wikipedia article. The first one is internal link, which links a page to another page within English Wikipedia. Wikipedia articles may also include links to webpages outside Wikipedia (external links), In the last several sections of a Wikipedia article such as notes, reference, the external links are widely used to cite the external descriptions as additional remarks.
Through the parsing of Wikipedia internal link numbers and external link numbers can be obtained, then the numbers of links between two revisions can be compared, so that the change in links intensity of each article can be monitored. The relationship between articles can also be studied from the perspective of link structure.

– **Category of Article**
This refers to the classification of the entries in Wikipedia, an article can be classified into one or more categories or set up new categories to cover the new common theme of articles. It can be applied to calculate the semantical distance between definitions and semantic meaning of an article can also be determined by comparing the categories.

3.3.2 Implicit Features in Wiki Text

Although three types of explicit features have been analyzed, it is still necessary to do natural language processing on each text snippet to explore the implicit features of plain text. To this end, the preprocess should be applied so that the term frequency statistics and term distribution can be obtained subsequently. The implicit features involve statistics about the term distribution and term frequency for the revisions of a Wikipedia article, which calls for the topic or thematic analysis of the article. After the section level change analysis by word number comparison and difference engine processing which are mentioned above, the statistics of term should then be collected for each revision with the aim of further exploring the connections between articles as well as detecting the trend of content changing. The term frequency or term distribution is, in fact the best index for further exploration of the content.

3.3.3 Difference Function

According to the analysis which has been made before. A common LCS difference function, which has already been introduced in Section 2.3, can satisfy our requirement. After the parsing process, the first level comparison is made based on the word number change, in which addition and deletion can be detected. Then the text is compared in the changed section to identify the sentences level or word level changes. Google diff algorithm is employed for the development of our difference function.

3.4 Component Diagramm

As can be seen in Figure 3.5, the important components of the event detection prototype are illustrated in the diagram, namely *crawler*, *analyzer* and *summarizer*. The concept of the *crawler*, the *analyzer* and the *summarizer* will be described in the following sections.

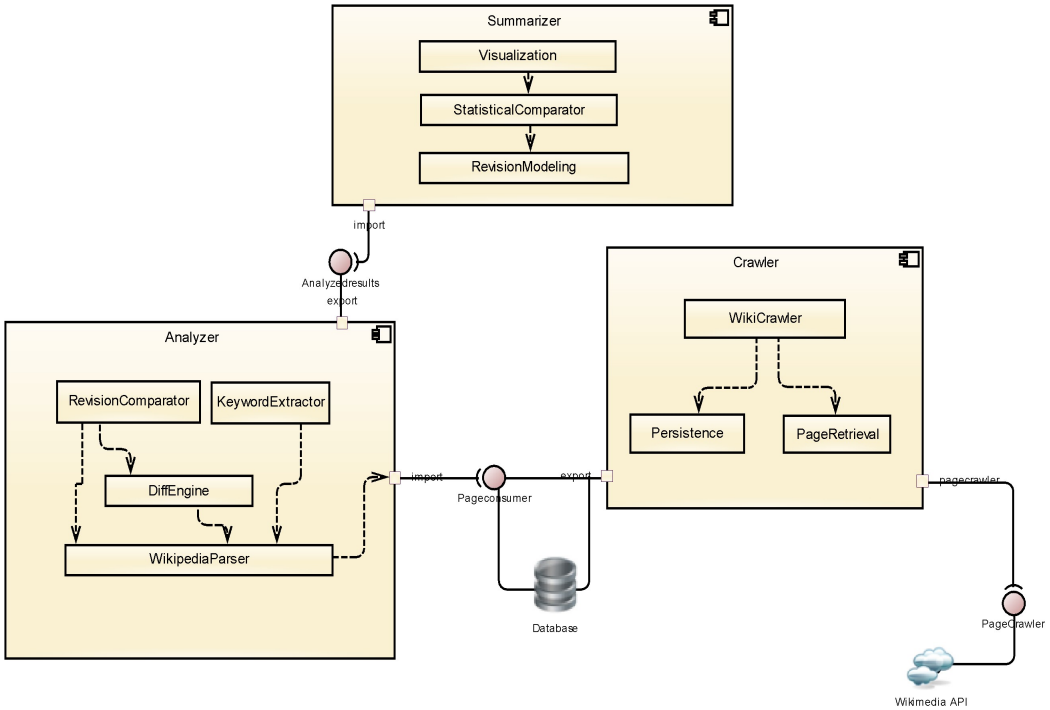


Figure 3.5: Component Diagramm

3.5 Event Detection based on Extracted Features

In this section, the method of revision modeling and the extraction of content feature are introduced in the first place, which is followed by the explanation of the other two features, namely time interval and link intensity. As next, an overview of processing steps is presented. At last, the algorithm of event detection is presented.

- **Revision Content Modeling:** Due to user editing, the content of Wikipedia articles is changing. Each Wikipedia article represents an independent concept, and the concepts are linked through the Wikipedia internal link mechanism to constitute a topic related link graph. It is easy for the editors to refer to the other Wikipedia pages via internal links, which enrich the content of the article and promise more background knowledge. This makes different revisions of the articles connected with contextual relatedness according to a specific topic. Conversely, the change in the content of revisions and linkage between them can be tracked in order to find out event related change as well as the change in the editor's attention. As explained in Section 2.5.3, the research has already been conducted on event detection through weighted link graph. Moreover, after parsing process, all the internal links have already transformed to plain text, the newly added internal links can therefore be processed as a normal term in the parsed text. As a result, our event detection concept mainly focuses on detecting revision content change using term frequency statistics. For example the earthquake and tsunami disaster in Japan in *March, 2011* leds to plenty of edits on a series of pages related to the territories where the disaster occurred and the corresponding articles for the background knowledge, therefore, the analysis of the content change in these articles can naturally reveal the clue of event-related information such as event beginning time and description of the event. So the keywords of each revision are chosen to represent the revision content. With the frequency statistics of these keywords a bag of words model can be built for a revision, so that the content change can be traced by comparing the distribution of keywords. If the frequency of a keyword remains the same for a long period of time, the keyword can be marked as fixed text glossary. Those keywords that appear for the first time or words with drastic change in the frequency

can be marked as newly appeared or event related vocabulary.

- **Time Interval of Revisions:** The edit interval between revisions is another important index for event detection, since during or after a certain event the article will be edited very frequently in a short time interval. So the subtraction of timestamps of two adjacent revisions is used to represent the average edit interval, which indicates the editing rhythm of Wikipedia pages in the entire course of the editing history. Under normal circumstances editing event begins with the increase in editing frequency, namely user attentions. As a result the conversion from long to short editing interval can be seen as the first occurrence of an editing event. Inversely the increasing in editing interval can be seen as the end of an event. Naturally, the editing interval will be short and constant during the entire editing event.

- **External Link Intensity:** The link intensity is the third index for our event detection. Besides using internal link to cite the Wikipedia page, the quotation from external web pages is another important way to enrich and approve Wikipedia page. Furthermore, the existing time and title of the external link reflect also timeliness and outline specification of the event, which can be used as supplementary evidence to the two aforementioned features by event detection process.

3.5.1 Overview of the Processing Steps

The detection process consists of the following parts. Firstly, some preprocessing shown in Figure 3.6 will be made. The text content of each revision is extracted and parsed into plain text. Yet due to the wiki parser which has been employed, the template and file Wiki Markups still exists after parsing process, so a further elimination of the template and file markups will be conducted. Next, tokenizing is employed, the tokenizer is to break character streams into tokens in certain length by using predefined grammatical rules. In the prototype, tokenizer is used to obtain the tokens in the form of a single token or a sequence of tokens in one sentence. After the parsing process all the Wiki Markups have already been removed. Preprocessing is performed with the aim of improving the general quality of the input data and enhancing the accuracy and efficiency of the keyword extraction. In second step, The RAKE algorithm 3.5.2.1 is applied to extract the keywords in the revision content. The text content will first be separated with word delimiters and stop words.

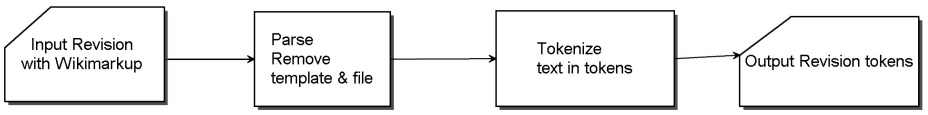


Figure 3.6: Preprocssing Steps

The candidate keywords are the token sequence between these delimiters and stop words. After that, co-occurrence matrix will be established and scores of each tokens will be calculated, which will be explained in detail in the following section. Then top 30% keywords with high scores will be chosen as the keywords for the revision. Subsequently the bag of words model is established to illustrate the frequency change of the corresponding keywords in each revision.

3.5.2 Rapid Automatic Keyword Extraction

3.5.2.1 RAKE versus *TF-IDF* and χ^2 (Chi-squared)

Keywords extraction is one of the most important techniques in information retrieval, which plays an important role in text mining and document clustering. *TF-IDF* is a popular algorithm for keywords extraction, where the words with high frequency in a document and low frequency in the remainder document corpus will be highly ranked in the keyword list. So this method extracts the keywords from a group of documents and the entire corpus of documents will be observed during the extraction process. But for our prototype, a group of keywords will be used as indexes to observe the content change in each revision. For instance, the newly added content could have more or less influence on keywords of the revision. So using *TF-IDF* to extract keywords based on the entire revision corpus is not possible to detect the change with only one revision text. A keyword extraction algorithm for a single page should be employed instead of a corpus of pages.
To achieve this goal, rapid automatic keyword extraction (RAKE) and the χ^2 based keyword extraction are two of the possible candidate methods. Matsuo and Ishizuka [MI04] apply χ^2 square method to measure the bias degree of the co-occurrence distribution of words which indicates the importance of the words. The co-occurrence of a term represents the relationship between

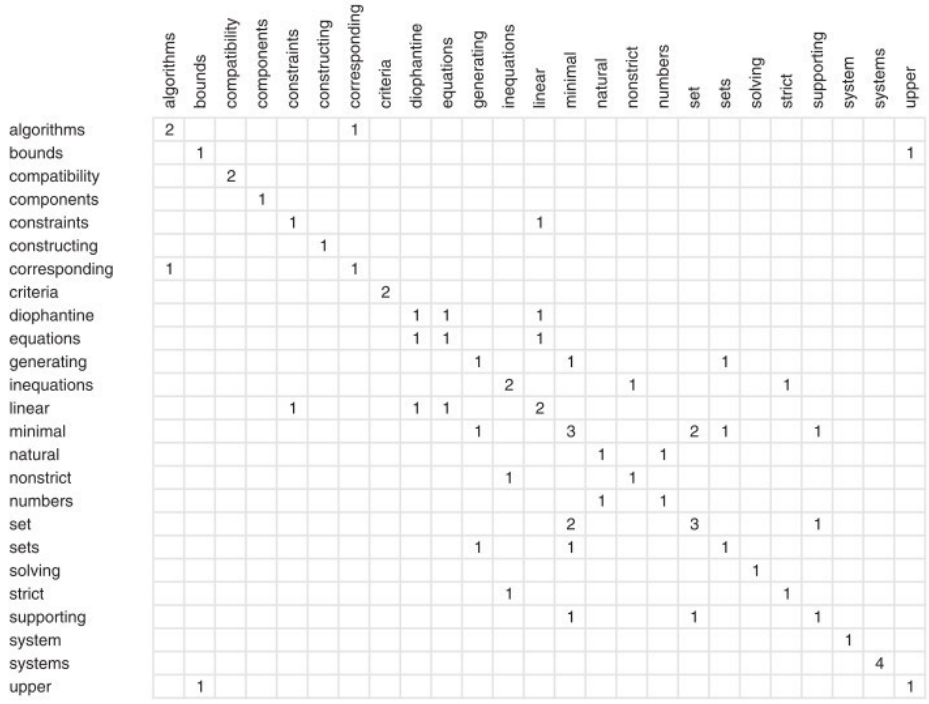
this term and the other terms in the page. If the term refers to one or more terms with high occurrences, then it might have an important meaning in the the whole page. The bias degree of the co-occurrence distribution will be described by the χ^2 measure. Matsuo and Ishizuka indicated that the degree of biases is not reliable when the term frequency is small. Since in some short texts the important words have very low occurrence, in short texts some important words will be ignored by the first step, and only the words with high frequency will be used as keyword candidates. Meanwhile based on the full text experiment performed on a 27-page document by Matsuo and Ishizuka, the χ^2 based methods work well on the large document.
So for our prototype, the RAKE will be employed which is an unsupervised, domain-independent method for extracting keywords from individual documents. It is more computationally efficient than the other methods and higher precision and comparable recall scores. It divides the sentence with stop words and delimiters which offers the possibility of adapting to different domains and topics of documents by predefining corresponding stop word lists [MWB10]. In the following section the RAKE algorithm will be explained in detail.

3.5.2.2 Candidate Keywords and their Scores

The first step of RAKE is extracting the candidate keywords from the target document represented in Figure 3.7. The text of the document is then spilt into lists of words by specified delimiters, and the candidate words will be extracted from these sequences of words between each delimiter and stop word, which means the candidate keywords are those words and phrases, located between delimiters and stop words in a sequences of a document.

Compatibility – systems – linear constraints – set – natural numbers – Criteria – compatibility – system – linear Diophantine equations – strict inequations – nonstrict inequations – Upper bounds – components – minimal set – solutions – algorithms – minimal generating sets – solutions – systems – criteria – corresponding algorithms – constructing – minimal supporting set – solving – systems – systems

Figure 3.7: Candidate Keywords Parsed from Document [MWB10]



	algorithms	bounds	compatibility	components	constraints	constructing	corresponding	criteria	diophantine	equations	generating	inequations	linear	minimal	natural	nonstrict	numbers	set	sets	solving	strict	supporting	system	systems	upper
algorithms	2						1																		
bounds		1																							1
compatibility			2																						
components				1																					
constraints					1								1												
constructing						1																			
corresponding	1						1																		
criteria								2																	
diophantine									1	1			1												
equations									1	1			1												
generating											1			1					1						
inequations												2				1					1				
linear					1				1	1			2												
minimal											1			3				2	1			1			
natural															1		1								
nonstrict												1				1									
numbers															1		1								
set														2				3				1			
sets											1			1					1						
solving																				1					
strict												1									1				
supporting														1				1				1			
system																							1		
systems																								4	
upper		1																							1

Figure 3.8: Co-occurence Matrix [MWB10]

3.5.2.3 Calculation of the Scores of Candidate Keywords

After all the candidate keywords have been obtained, a co-occurrence matrix in Figure 3.8 will be established for each token of the key phrase, and then the score for each candidate keyword will be calculated, which represents the sum of the scores of its member words. In order to calculate the word scores, the following values of each token in the candidate keyword should first be calculated:

1. **Word frequency (*freq(w)*)** counts only the occurrence times of the word w itself, which means the value represents only the frequency of the word occurrence regardless of the co-occurrence times. The *freq(w)* is represented in the diagonal line of the matrix. In the above example, *freq(systems)* scores higher than *freq(linear)*.

2. **Word degree (*deg(w)*)** has a high score if a word occurs with a high frequency and in longer candidate keywords, which means this value sums up the occurrence and co-occurrence. The more accumulated occurrence times, the higher the degree score will be. The *deg(w)* is illus-

	algorithms	bounds	compatibility	components	constraints	constructing	corresponding	criteria	diophantine	equations	generating	inequations	linear	minimal	natural	nonstrict	numbers	set	sets	solving	strict	supporting	system	systems	upper
deg(w)	3	2	2	1	2	1	2	2	3	3	3	4	5	8	2	2	2	6	3	1	2	3	1	4	2
freq(w)	2	1	2	1	1	1	1	2	1	1	1	2	2	3	1	1	1	3	1	1	1	1	1	4	1
deg(w) / freq(w)	1.5	2	1	1	2	1	2	1	3	3	3	2	2.5	2.7	2	2	2	2	3	1	2	3	1	1	2

Figure 3.9: Word Scores Calculated from the Word Co-occurrence Graph [MWB10]

trated by the sum of column of the matrix. For example, *deg(minimal)* scores higher than *deg(systems)*.

3. **Ratio of degree to frequency (*deg(w)/freq(w)*)**. High ratio of degree to frequency means that those words have more occurrences in longer candidate keywords. In this example *deg(equations)/freq(equations)* scores higher than *deg(linear)/freq(linear)*

As can be seen in Figure 3.9, the candidate keywords with high ratio of degree to frequency are accepted as the key words of the target document.

3.5.2.4 Keywords Frequency Statistics

In this step, the keywords statistics (bag of words model) for each revision are established. With the extracted keywords from the last step 3.5.2, the content of each revision can be labeled with its keyword tokens. Firstly, the duplicate keyword token are deleted, because RAKE algorithm returns a group of keywords for the revision and every keywords are made up of tokens. Some tokens might also exist in other keywords, so for the frequency statistics only a sequence of unique tokens of keywords is used as statistical vocabulary of the revision content. Next, the frequency distribution of these keywords are calculated according to the bag of words model. As a result, each revision can be represented by the distribution of the keywords frequency. With this distribution, the change in the content between revisions can subsequently be measured by various statistical comparison methods such as Anderson-Darling, χ^2, Kolmogorov-Smirnov, which have been introduced in Section 2.4. The exact significant level will be determined based on corresponding comparison methods and use case. According to the comparison method offered by JAIDA 4.4.2, the value of the comparison result represents the distance between two

distributions, which means large value indicates significant difference whereas small value suggests no significant changes. The degree of change is quantized by the comparison value. For instance, if the content is changed, i.e. some keywords changes occur in the revision, the distribution of keywords may turn out to have a new shape; consequently, the number of keywords and their frequencies will change respectively, which leads to the fluctuation of the result value. Similarly, the revisions with drastic changes in keywords and their frequencies of occurrences can be seen as the occurrence of an editing event, whereas the revisions with constant frequency indicate an already established fact or consensus of the Wikipedia community.

3.5.3 Event Detection:Event Boundary Determination and Ranking

After statistical comparison, the differences in keywords distribution between two adjacent revisions are identified, based on which the change in content can be determined. In our concept, an event consists of three important elements: the beginning time, the development process, and the content change 3.10. The beginning time is the timestamp of the first event related revision; the development process is constituted by the event related revision sequence in the chronological order; and the content change shows the newly added keywords since the last revision, which gives a brief description of the event. In order to detect all these three elements from the entire revision set, the beginning time of the event should be determined first. As we know, the editing after the event occurrence leads to significant change in the content and rapid shortening of the editing interval, so the derivative of the content change value and the editing interval as well as the link intensity are used to track the ratio of change and to determine the event beginnings. Given two Revision points:
$Pt_0 = \{Timeinterval_0, Contentdistance_0, Linkintensity_0, Timestamp_0\}$
$Pt_1 = \{Timeinterval_1, Contentdistance_1, Linkintensity_1, Timestamp_1\}$

$$tDerivative_0 = \frac{Timeinterval_1 - Timeinterval_0}{Timestamp_1 - Timestamp_0} = \frac{\Delta Timeinterval}{\Delta Timestamp}$$

$$cDerivative_0 = \frac{Contentdistance_1 - Contentdistance_0}{Timestamp_1 - Timestamp_0} = \frac{\Delta Contentdistance}{\Delta Timestamp}$$

$$lDerivative_0 = \frac{Linkintensity_1 - Linkintensity_0}{Timestamp_1 - Timestamp_0} = \frac{\Delta Linkintensity}{\Delta Timestamp}$$

However, at the beginning of the event the content change is not always clearly visible. In other words, the event related editing starts sometimes with minor edits, followed by increasing editing activities that will reach a peak shortly after. Therefore, three distance metrics (see Table 3.1) between revisions are chosen to locate the first event related revision in the revision history. The first one is the nonzero statistical content distance value with increasing derivative, which covers all the revisions with content changes including the minor edits, and the second one is the small time interval with decreasing derivative, by which the earliest event related revision can be found out precisely. The third metric is the positive value of link intensity with increasing derivative, which indicates increasing number of citations of external resources. Hence, the detection factors are represented in Table 3.1, where V represents the value of the factors:

Detection Factors	**Metrics**
$F_{TimeInterval}$	$V_{TimeInterval} < 1day \wedge tDerivative < 0$
$F_{ContentDistance}$	$V_{ContentDistance} > 0 \wedge cDerivative > 0$
$F_{LinksIntensity}$	$V_{LinksIntensity} > 0 \wedge lDerivative > 0$

Table 3.1: Detection Factors

However, the configuration of these three factors, namely which factor or combination of factors show highest effectiveness of detection and the different properties of the these three factors, should be further tested and discussed in Chapter 5. Moreover, the decompostion of these three factors will be conducted when neccessary so that the properties and uses of the subfactor can be tested completely.

The entire event development process is expanded under the predefined weekly time span, i.e. the following editing revisions which happened within one week are clustered into the same event as the previous editing group, which can be seen in Figure 3.10. But the determination of the event end cannot be decided by the end of this development process, because events always happen at one specific time point. Besides the editing which directly follows the timestamp of the event, there is still some subsequent editing which will be made randomly and periodically, so there is no definite end of editing events or social events in Wikipedia context. Subsequently, the newly emerging keywords are used to describe the content of event 3.10. Since new keywords and meaningful groups will be added in each editing, these emerging keywords reflect directly the new event. In the last step, after the event related revisions

are clustered into corresponding event groups, the event groups will be ranked according to the editing intensity (number of revisions between boundaries), which means the event group with maximum number of revisions is considered as the biggest event in the page. The entire detection approach is presented in Listing 3.1. There are also other factors, which can be used to do the ranking, i.e. the sum of content distance value or sum of link intensity. In order to find a most suitable ranking method, the corresponding ranking experiments will be performed in Chapter 4.

```
Begin Detection
While Editing History Set not Null
IF (Detection Factors and their Combinations
  satisfy the detection conditions>
    Revision Timestamp is stored as Candidate
      Revision)
End IF
End While
While Candidate Revision Set not Null
IF (Timeinterval between Candidate Revision n and n
  +1 < 7 days)
    Candidates Revisions are Clustered to the same
      Event
End IF
End While
    Return Ranked Event according to the Editing
      Intensity
End Detection
```

Listing 3.1: Detection Approach

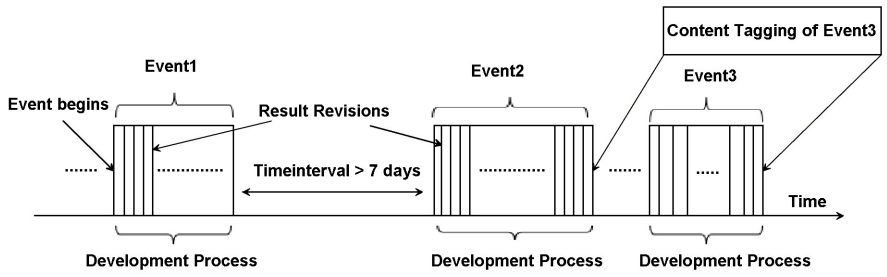


Figure 3.10: Events in Revisions

Chapter 4

Implementation

This chapter describes the implementation details of the prototype. For each part, the important classes, methods and used libraries as well as APIs are presented.

4.1 Crawling of Wikipedia

In this section, the implementation and extension of the Wikipedia crawler and persistence approach of crawled page and revisions are presented.

4.1.1 Wikimedia Crawler in Palladian

Palladian is an internet information retrieval toolkit, developed for facilitating the use of algorithms for text processing. It provides the functions of classification, extraction and retrieval.
Our prototype starts with the `WikiMediaCrawler` in the retrieval package of Palladian, The `MediaWikiCrawler` provides the function to retrieve content and metadata from Wikipedia. It uses the MediaWiki API [Wma] to obtain the information. To communicate with the API, the crawler bases itself on the Java Wiki Bot Framework (JWBF). It facilitates access to the Mediawiki database, which provides methods to connect, modify and read corpus from wikis. It also offers the possibility to create a wiki bot and provides the corresponding methods to cope with MediaWiki API which makes access to

the Wikimedia database much easier. There are plenty of basic features of `WikiMediaCrawler` and our extensions are listed as follows:

- Download the content of a complete wiki.
- Store content in relational database.
- Select namespaces to crawl and/or **select the articles to crawl according to our data set**.
- Access login protected Wikis.
- For a single page, the following information will be extracted:
  - Page title;
  - HTML content of the head version, rendered by wiki;
  - Complete revision history including revisionID, timestamp of modification, author and **the content of the revision**;
  - All links to other pages within the same wiki.

Figure 4.1 presents the architecture of the Wikimedia crawler. In the entire system, `MediaWikiCrawler` and `PageConsumer` have played major roles. `MediaWikiCrawler` fetches pages from the wiki database, and `PageConsumer` processes the extracted data. Package `data` provides classes to model the basic data structure of wiki pages. These classes store fetched pages and their metadata temporarily, while package persistence is used as a persistence layer to store the extracted data in a database. The configuration of `MediaWikiCrawler` will be loaded from the configuration file. Package `queries` extend some queries of the Java Wiki Bot Framework (JWBF). All the interaction between `MediaWikiCrawler` and MediaWiki API are conducted by JWBF. `PageConsumer` always runs with the `MediaWikiCrawler`, i.e. the crawler fetches wiki pages to the processing queue and `PageConsumer` processes the pages at the other end, which causes too high coupling between these two components. In order to solve this problem, another class named `Consumfromdatabase` is implemented, by which the pages can directly be processed from the database at any time instead of during crawling, which provides more flexibility and lower coupling between those two components.

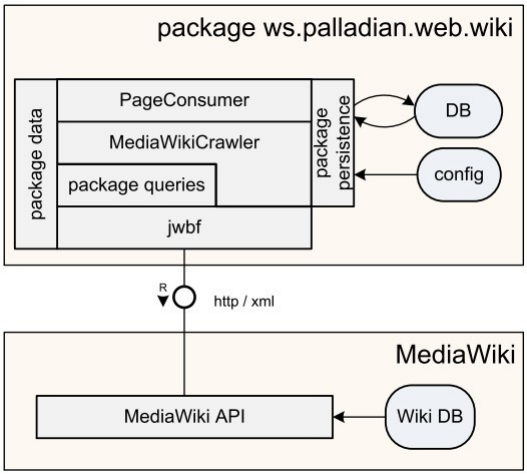


Figure 4.1: MediaWikiCrawler Architecture.

4.1.2 Modeling Wikipedia

In this section, some important classes for modeling Wikipedia's structure and content will be introduced, which are illustrated in Figure 4.2.

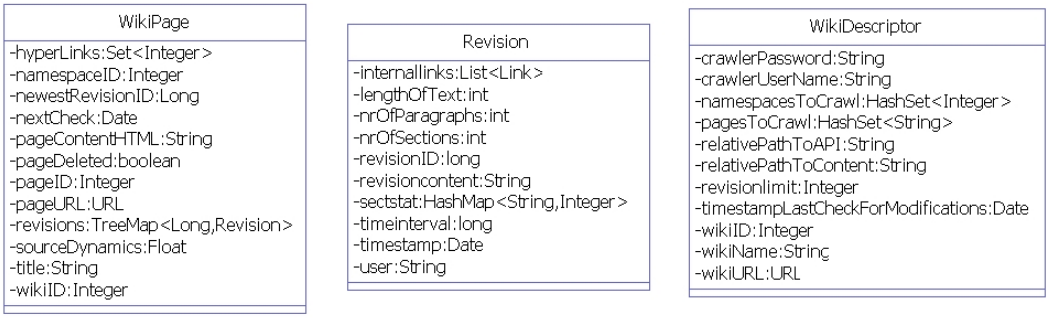


Figure 4.2: Wiki Model

- **WikiDescriptor:** This class is in charge of loading local settings for wikicrawler from the YAML file. YAML is a human-readable data serialization format, which is more data oriented than the other markup languages. It is designed to be easily mapped to data types that are common to most high-level languages such as list and array, so it is suitable and chosen for the configuration of our `WikiMediaCrawler`. The wiki to be crawled, the target namespace as well as the target article titles will

be defined in attributes `wikiName`, `namespacesToCrawl`, `pagesToCrawl`, while `crawlerPassword and crawlerUserName` are used for the registration for Wikimedia query. `revisionlimit` regulates the number of revisions which should be crawled for each page. `WikiDescriptor` is nearly the same as `WikiDescriptorYAML`, and the only difference between them is the attribute `timestamplastCheckForModifications` which identifies the status of the `WikiMediaCrawler`. In other words, after the data in the current database is crawled completely, the `WikiMediaCrawler` will alternatively be switched between the crawl mode and the sleep mode, which will be used as a time record for the last checking of new updates.

- **PageTitle:** After `WikiMediaCrawler` configuration is loaded, the corresponding namespaces and page titles in the namespaces can then be crawled and stored in the `page` table.

- **WikiPage:** In the Wikipedia, content is described and stored in pages. In our prototype, the model in Wikimedia crawler of Palladian is extended and represent each page with a unique `namespace ID`, a unique `page ID`, `wiki ID` and `pageTitle`. The page content will be expressed in HTML format, and some further attributes such as `newestRevisionID`, `hyperlinks`, `sourceDynamics` will be set up for further manipulations on pages. With the page titles which have been already stored in the `page` table, the further query over the above mentioned information will be made. Once a wikipage is crawled, the page-related meta information can then be stored in the `page` table. All the revisions of this page will be crawled subsequently. After the crawling of all the revisions for the page is finished, the crawling for the next page will start.

- **Revision**: For each revision, four attributes have been defined to identify the features of a revision: `revisioncontent` stands for the revision content with Wiki Markup; `revisionID` is the unique identification for the revision; `timestamp` is the update time of the revision; and `user` indicates who has made the revision.

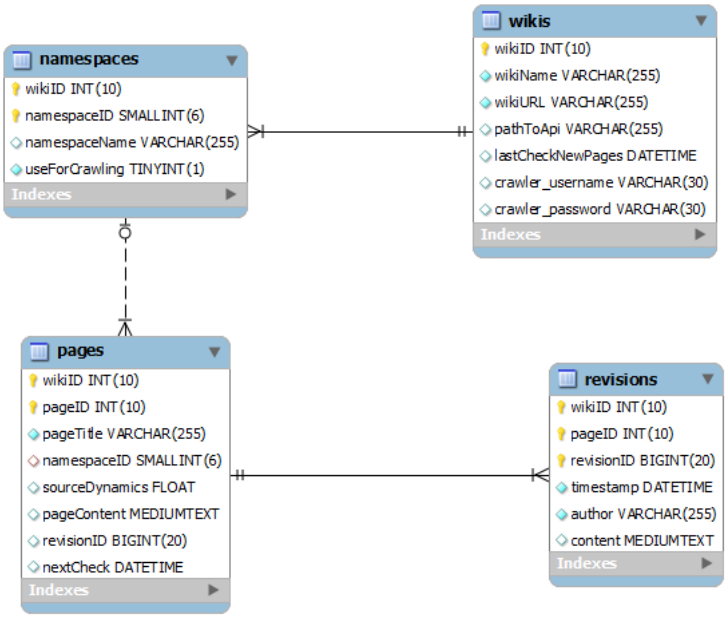


Figure 4.3: Database Schema

4.2 Parsing of Wikipedia

After the crawled revisions and corresponding meta information for each page have been stored in the database, they can then be used to reconstruct the past status of the chosen Wikipedia page and then, the task is to access and analyze revisions in the database using the Wikipedia parser which returns the desired size and structure of the revision content without Wiki Markups for further analysis.
Here two typical parsers for Wiki Markup are introduced: Wikipedia java parser[1] and jwpl Wikipedia parser[2].

Wikipedia java parser is an event-based java parser for wiki-markup text, which parses Wiki Markup and converts it to HTML or other output formats. It supports, however, only a subset of the Wiki Markup notation, including tables text decoration like italics and bold, ordered and unordered list headings, regular links and smart links and <nowiki> tags, etc. The smart links here refer to a method to tackle the links transformation problem in the parsing process from Wiki Markup to html file. The parser can be extended by adding

[1] `http://code.google.com/p/java-wikipedia-parser/`, accessed on July 09, 2011
[2] `http://code.google.com/p/jwpl/`, accessed on July 09, 2011

new markup from user which could tackle with these individual cases, and any links can be resolved correctly by extending the resolving method. But this parser is only a startpoint of wikiparsing and some of the functions still needs to be extended to cover all the requirements in our parsing process.

In contrast to Wikipedia Java parser, Jwpl parser is more powerful for Wikimarkup text. Developed by Ubiquitous Knowledge Processing Lab of Darmstadt University of Technology. It analyzes the structure of a text with Wiki Markup and represents it as a Java object which allows for structured access to the contents of Wikipedia. It is not a standalone release of the parser, since it is part of the JWPL Wikipedia API release. However, it can be used without accessing Wikipedia with JWPL. The wiki text will be divided into sections and paragraphs where the internal links and external links can also be accessed. Meanwhile the meta information such as word number and word format in wiki text as well as the category of each article is also available for further text analysis. The procedure of the parsing and comparing process in our prototype are presented in Figure 4.4.

4.3 Revision Change Categorization

In this section, the implementation of revision comparison and change categorization will be described in detail.

4.3.1 Difference in Revision Content

In order to compare revisions of different text sizes and levels, the revision class is extended with several attributes which can be used as features to identify the changes in the revision text:

- word number
- internal links
- number of sections
- sectstat, which represents the section title with the corresponding word number in the section, and is used to identify the change in each section.

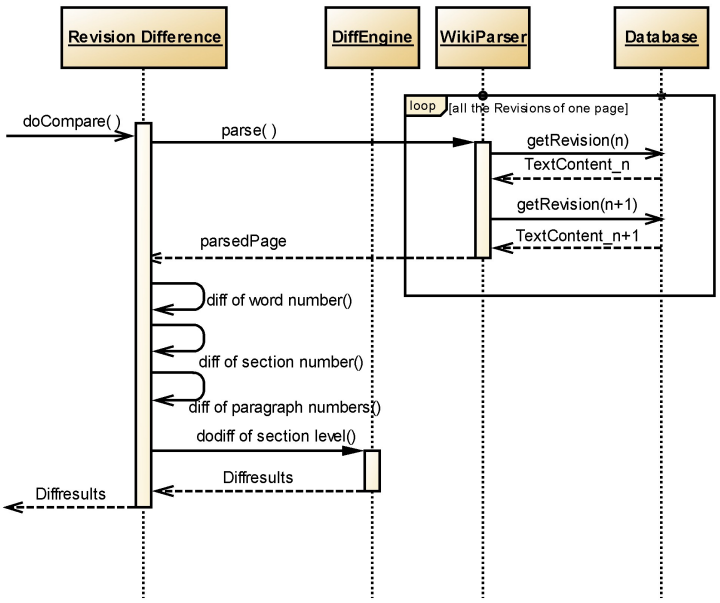


Figure 4.4: Sequence Diagram of Revision Comparison

The number of words and sections as well as the word number of each section are compared to establish a section level differences between two revisions. In other words, the word number of the whole text is compared firstly to find out the overall change situation between revisions, and then the number of sections and the word number in each section are compared, by which the differences at the section level can be determined. The newly appearing sections in the new revision can be labeled as insertion whereas the sections that can be found in the old revision but not in the new one can be labeled as deletion. With regard to the sections found in both revisions, the difference function is employed to make a finer comparison, so that the exact changes can be located and the editing actions such as deletion, insertion, are determined.

4.3.2 Keyword Extraction of Revision Content

In order to establish a histogram of keywords for each revision, keyword extraction component is developed according to the RAKE algorithm. Because in RAKE the stop words are used to split the text into sequences of token, so the `StopWordManager` is initialized first to get the predefined stop word list, and then the input tokens are split into word fragments by

eliminating stopwords with `extractCandidateKeywords`, which are called keyword candidates. Next the co-occurrence matrix for keyword candidates will be created by calling `createCoOccurrenceMatrix`, which is followed by `computeDegreeFrequenceRatio` that calculates the degree of frequency ratio for each candidate. After the matrix is filled with the score of each candidate, the scores of candidate keywords (word fragments) obtained from the keyword extraction step are summed up. At last the candidate keywords with high scores are returned as extraction results.

4.4 Detecting Events in Revision History

In this section, the implementation of the editing event detection using statistical methods based on the extracted features will be explained.

4.4.1 Building-up Histogram

In order to characterize the bag of words model for each revision, JAIDA is depolyed to establish the 1D histogram for each revision[3]. First of all, the analysis factory `IAnalysisFactory` is initialized, by which various data types and comparison methods can be generated. Next, the binned histogram in `IHistogramFactory` is employed to describe word bag of revisions. Each extracted keyword from last step is assigned with a fixed serial number which corresponds to the binned position on the x-axis, while the y-axis indicates the frequency of the word. The keywords are stored in the static HashMap. After the keyword map is initialized for the first revision, the new keywords in the following revision will be inserted into the map in an incremental way. The new keyword list will be compared with the old one and only the new keywords are added into the keyword map whereas the x-axis is always extended to the right with the newly assigned serial number. In this way, the keywords in one page are placed in a fixed order, so that the change and newly added keywords can be clearly treated and stressed for the comparison in the next step.

[3]JAIDA is a toolkit for data anaylsis, `http://java.freehep.org/jaida/`, accessed on July 09, 2011

4.4.2 Statistical Comparsion

After the generation of histogram for each revision, various comparison methods offered by JAIDA[4] such as χ^2 (Chi-squared), Kolmogorov-Smirnov, Anderson-Darling, etc. can be applied to measure the distance between the keywords distributions (see Figure 4.5). *StatisticalComparison.compare* returns the distance of two distributions, which indicates the degree of differences between two distributions.

```
StatisticalComparison.compare(hist1,hist2,"chi2","");
 //Chi squared algorithm
StatisticalComparison.compare(hist1,hist2,"AD","");
 //AndersonDarling algorithm
```

Listing 4.1: Statistical methods

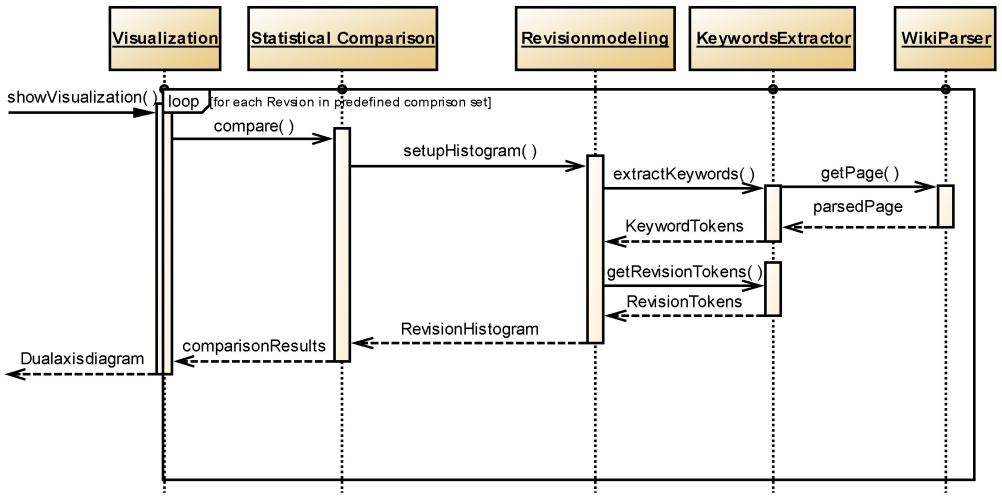


Figure 4.5: Statistical Comparison

[4] `http://java.freehep.org/jaida/statistical-comparison.html`, accessed on July 11, 2011

4.4.3 Event Detection

The comparison results of two adjacent revisions are temporarily stored in class `result`, which represents the comparison result, time interval, timestamp and their event boundary flags. According to the event detection condition defined in Section 3.5.3, the revisions are clustered into different event groups with event beginning and event development process. Each result object is defined with two flag bits `start` and `end` to present the beginning of event and the end of editing event under the predefined time span.

4.4.4 Event Tagging with Keywords and Title of External Links

Only with the event time boundary, the situation of event during a certain time period is still unclear. As we know, the increased keywords are strongly related to the current event, and the newly appearing external links quote also the event related news and instance from external medium. Therefore the diff of keywords as well as the title of increased external links of two adjacent events are extracted through RAKE keyword extraction, which has been explained in Section3.5.2 and Palladian link title extraction toolkits (see Listing 4.2), which give the event not only the content description from the Wikipedia page itself but also the approvement of external websites. More concretely, the new emerging external links are generally from the news agencies or web archives, the title of these links provides therefore a brief and clear description to the corresponding event, while the keywords diff based on RAKE represents the more detailed aspects of events.

```
HttpResult httpResult = documentRetriever.httpGet(
  pageUrl);
Document document = parser.parse(httpResult);
String title = PageAnalyzer.extractTitle(document);
```

Listing 4.2: Title Extraction from External Links

4.4.5 Visualization with JFreechart

JFreeChart[5] is a Java chart library which facilitates the visualization of data set in different applications. It provides a wide range of chart types and easy extendibility. According to our results data, the three factors should be displayed on the same time axis, so a tri-axis timeseries chart is developed based on the *Dual Axis Demo 2*. The x-axis represents the time series of the data set whereas the three y-axes depict the content distance, the time interval and the link intensity respectively.

4.5 Summary

In this chapter, the implementation of our prototype has been explained in details. The important components and classes are described in details, together with the used libraries and toolkits. In the next chapter the test results are presented and the core functions of the prototype will be evaluated.

[5] `http://www.jfree.org/jfreechart/`, accessed on July 09, 2011

Chapter 5

Evaluation

In this chapter, the concepts and their implementation are evaluated, which is divided into two parts: analysis of the Wikipedia data, which presents the visualization result of Wikipedia change, together with the meaning of the extracted factors and their usages in event detection and the result analysis of event detection. The steps of validation and the methods of result comparison as well as different prototype configurations are presented and compared.

The validation of event detection is based on the time line comparison. The detection results of our prototype are compared with the benchmarks extracted from the "event by month" page on the Wikipedia portal.

5.1 Analysis of the Wikipedia Data

In this section, the visualization results of revision modeling and change comparison based on the extracted features are presented:

5.1.1 Revision Modeling

Our prototype provides the histogram of each revision. As can be seen in Figure 5.1, the revision of page Tsunami with revision is represented in 1D histogram. For this revision, 1031 keywords (word on x-axis) tokens have been found in the revision content and there are totally 829 non-repetitive keyword tokens from the first revision till this one, which has been accumulatively

placed on the x-axis. In order to clearly observe the change of each token by the comparison in the next step, each token is assigned to one bin (discrete interval in histogram). The frequency of each token is shown on the y-axis and the distribution of the token frequency constitutes the basis for statistical comparison in the following steps. Figure 5.2 shows four consecutive revisions of page *Tsunami*, which indicates the change in word tokens as well as count and their distributions in these four adjacent revisions.

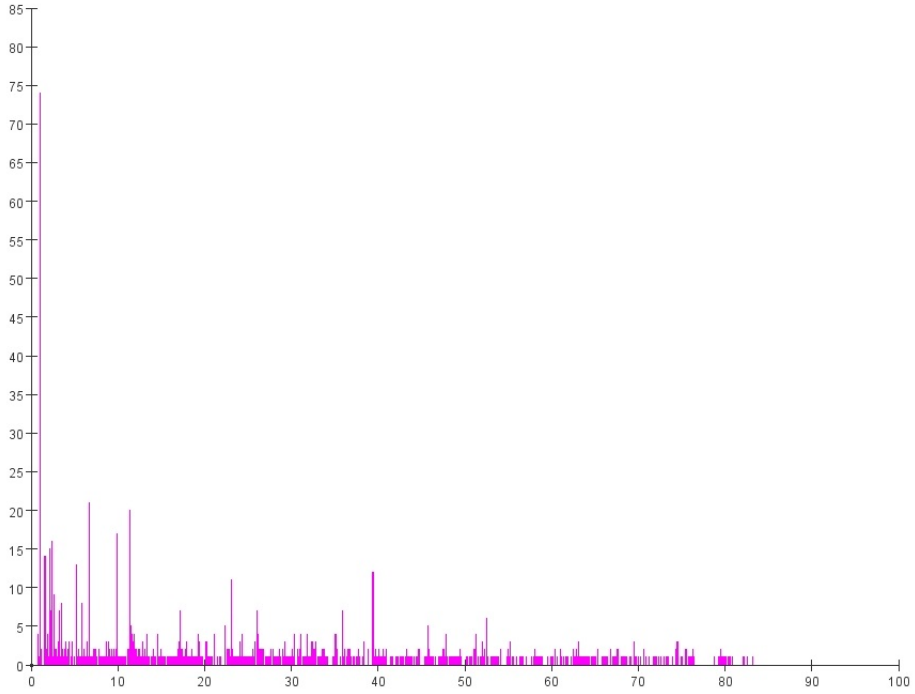


Figure 5.1: Histogram of Term Distribution

5.1.2 Test Data of Event Detection

In order to test the effectiveness of event detection, the comparison results of the chosen factors and the event detection results are first stored in database for further evaluation. As next, the beginning time and tagging relevance are used to evaluate the effectiveness and efficiency of our event detection. The chosen Wikipedia pages are from different categories such as person, location, which are related to social events of different types (see Appendix A.1).

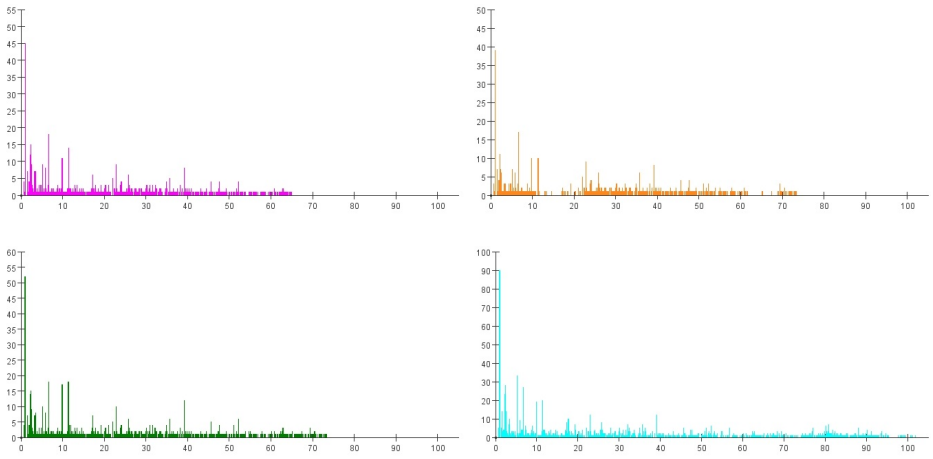


Figure 5.2: Histogram of Four Adjacent Revisions

5.1.3 Detecting Approach

First of all, the *Page title(concept)* from the test data and *revision ID* are inputted into the prototype. There are three parts of output data, first is the triaxis diagram, in which the *x-axis* represents timestamp and the *y-axis1 (left side)* shows the content distance value whereas the *y-axis2 (right side)* depicts the time interval of the revisions. Last but not least, the *y-axis3 (right side)* stands for the changes of link intensity. Second part of the output data is the increased keywords of events which are extracted by RAKE. Third part gives the extracted title of increased external links of events to describe the event content in brief. If the comparison method is not specially defined in the following test, statistical method Goodman is employed as the main comparison method of content distance.

5.1.4 Detection Factors

In this section, the visualization of three factors (time interval, content distance, and external links intensity) will be first illustrated separately, and then the meaning of the diagram and the development of event are explained in detail.

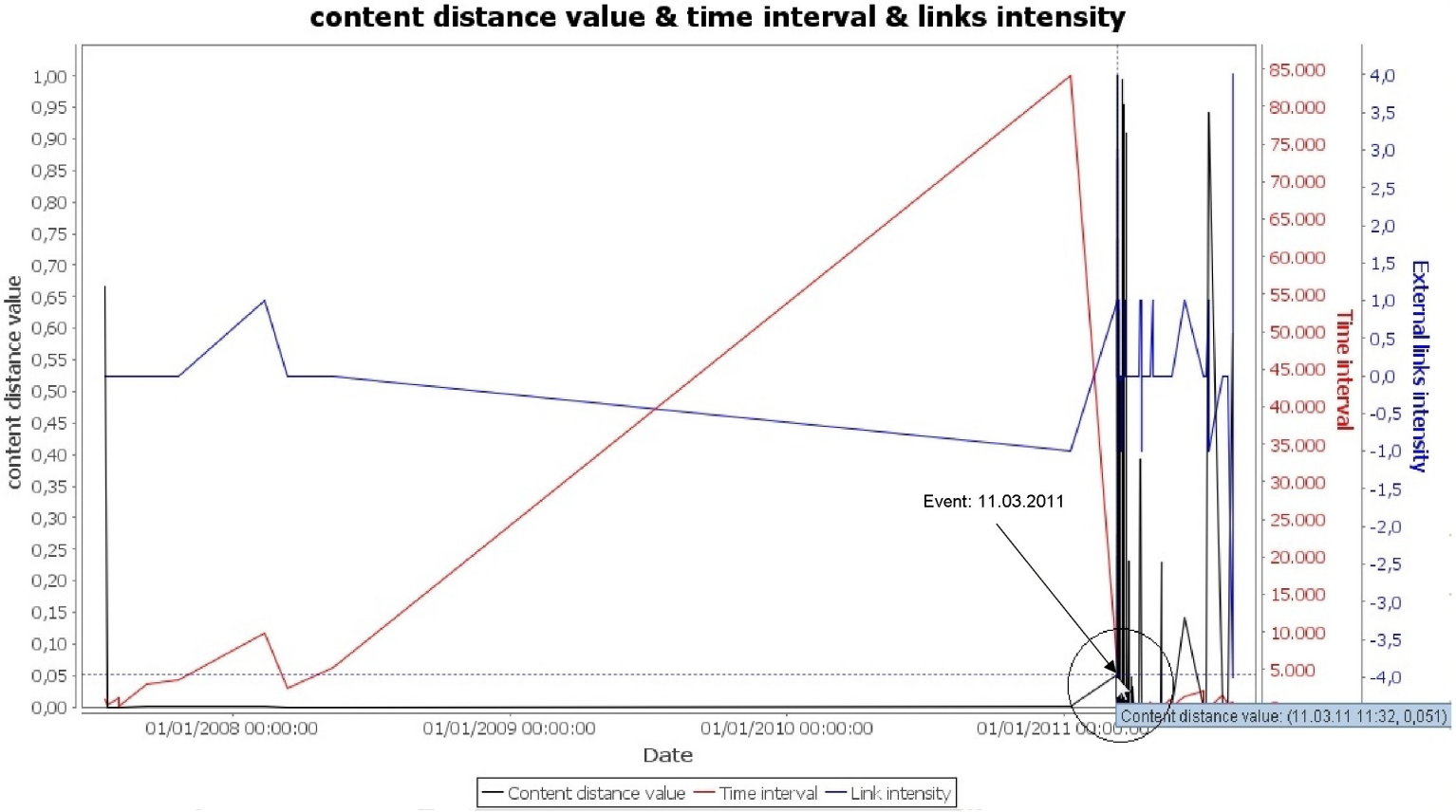


Figure 5.3: Fukushima Daini Nuclear Plant Represented by Three Factors

The Figure 5.3 represents the entire editing history of page *Fukushima Daini Nuclear Power Plant* (see Appendix A.1), which has been established and modified over the period from *July 2006* to *July 2011*. As can be seen in the diagram, the black curve is the connection line of all the values of content comparison, while the red curve joins all the points of time interval value together. The blue curve stands for the variations in the intensity of external links. The fluctuation of the time interval (red line) reveals the change of editing time distance, namely editing frequency, whereas the rise and fall of the content distance value (black line) and the link intensity (blue line) shows the changes of revision content and modifications of external links in corresponding revisions.

It can be seen that in the entire editing history of this page, only a few modifications has been made during a quite long period after page establishment, which finds expression in large time interval of editing, minor content distance and small change of links intensity between revisions. After this long period, significant change took place on *March 11, 2011*, on which day the 2011 To-hoku earthquake and tsunami struck. It can also be seen from the single factor diagram that, this event has led to obvious changes in each factor: the content distance sharply went up and down from *March 11, 2011* to *April 01, 2011* whereas the time interval reached the bottom from the event beginning and remained steady to the null point during the entire period. With regard to link intensity, the intense changes can also be seen clearly, whose positive and negative value have indicated the insertion and deletion of external links, respectively.

As can be seen from Figure 5.4, typical examples of event beginning point are circled out. By two factor detection (time interval and content distance), which is shown in black ellipse, the event beginning, i.e. *20:00:29 of March 11, 2011* is recognized with large content distance and little time interval value close to zero. The points in the red circle which appear short after the black circle at *23:50:06, March 11, 2011* represent the three factor detection results. Both the link intensity and the content distance show peak values whereas the time interval remains close to zero. Consequently, in contrast to two factor detection, the introduction of the third factor (link intensity) might postpone the recognition of event beginning due to the incoherent appearance of external links.

As indicated in Figure 5.5, the entire process of an event is detected based on the weekly span, which indicates the sustained content change with the

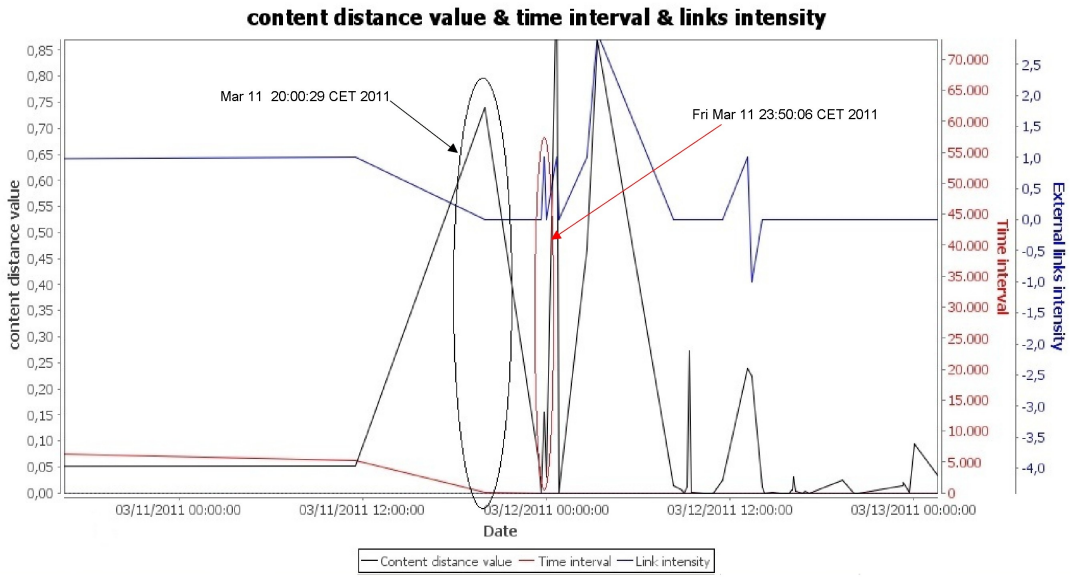


Figure 5.4: Event Beginning (Zoomed excerpt from Figure 5.3)

maximal one week editing pause. The external links are the references which have been cited from other web sources to enrich and approve the page content. Although the external links from news agencies or websites are good evidences of event occurrence, the synchronization of external links depends completely on the timely updates of Wikipedia users. For some events the external approvals appear or are cited with delays in various time lengths, so using external links as an additional detection condition might more or less damage the precision of the event beginning detection. Therefore, in the following section the effectiveness for detection with these three factors will be further evaluated, i.e. suitable detection of Wikipedia page and detection conditions will be discussed.

5.1.5 Event Tagging

After the boundary and development process of an event are fixed, the newly emerging keywords extracted by (RAKE) (see Figure 5.6) and titles of external link pages (see Figure 5.7) are extracted to tag the content of corresponding event.

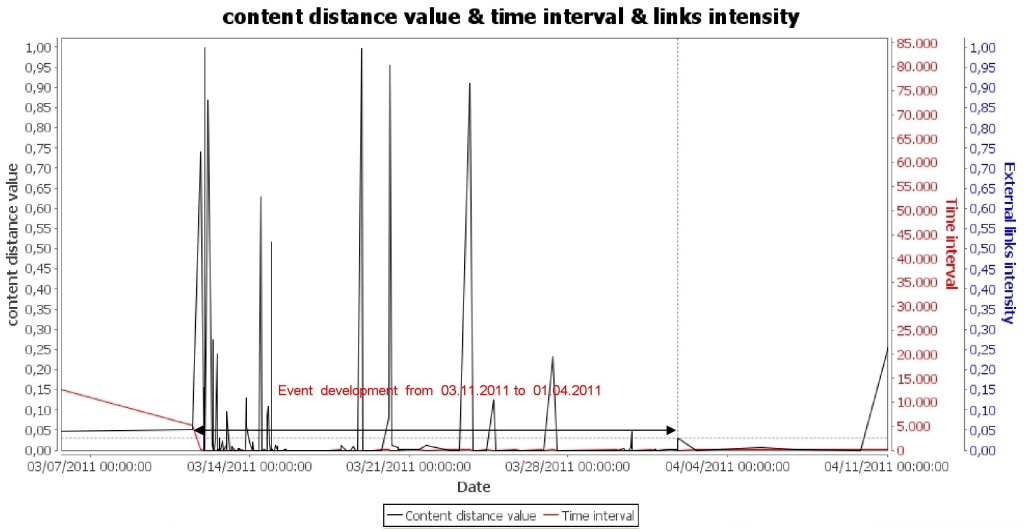


Figure 5.5: Event Development (Zoomed excerpt from Figure 5.3)

Fukushima No.2 plant reactors safely halted-- essential service water system transferring heat--Reactor supplier Architecture Construction Containment--civilian nuclear accidents --Fukushima Daiichi Nuclear Power Station--Fukushima II automatically shut down--Fukushima II nuclear accidents--International Nuclear Event Scale--worst nuclear accident occurred--maximum horizontal ground accelerations--Tokyo Electric Company--Fukushima Daini nuclear plant-- Japanese nuclear accidents--International Atomic Energy Agency--transmission line connection environment--Fukushima faced 14-metre tsunami--boiling water reactors--cooling systems remained operational--high pressure coolant injection--Japan initiates emergency protocol--Nuclear Power Plant--Fukushima II NPP--advanced Electrical connections--supplies electrical power--Fukushima Daini Plant--Nuclear Engineering International--Fukushima Daiichi plant-- collects outdoor water--radioactive incidents--Japanese nuclear plant--Hitachi Hitachi Shimizu Takenaka Mark--Fukushima engineering challenges--ultimate heat sink--Boiling Water Reactor--Officials made preparations--crane operating console--criticality Installation costs--recommence Coolant temperatures--selected radiation levels-- Japanese nuclear incidents--Tokyo Electric

Figure 5.6: Keywords for Event in Fukushima Daini Nuclear Plant

External links: TEPCO : Press Release | Occurrence of a Specific Incident Stipulated in Article 15, Clause 1 of the Act on Special Measures Concerning nuclear Emergency Preparedness (Unit 1)

External links: TEPCO : Press Release | Occurrence of a Specific Incident Stipulated in Article 15, Clause 1 of the Act on Special Measures Concerning Nuclear Emergency Preparedness (Unit 2)

External links: TEPCO : Press Release | Occurrence of a Specific Incident Stipulated in Article 15, Clause 1 of the Act on Special Measures Concerning Nuclear Emergency Preparedness (Unit 4)

External links: Emergency declared at 2nd Japan nuke plant after cooling fails

External links: 東京電力ホームページ

Figure 5.7: Example of the Titles of External Links

5.2 Detection Results Comparison

In this section, the results of event detection and evaluation methods are presented, and the different configurations will also be explained.

5.2.1 Automatic Generation of Benchmarks

In order to generate an independent event benchmark from third party, the events in "event by month" page from Wikipedia[1] is extracted in accordance with the page title in dataset (see Appendix A.1). This "event by month" page is composed by the big events occurred every day of the month from Wikipedia current events on portal site. These events are described with time, internal links to the related Wikipedia pages and a brief description of the events. The month which the event occured should first be defined, and then the page titles are used as keywords to extract page related events. The benchmarks are also clustered with weekly span. That is to say, if the two events (starting from the first one) have a time interval less than seven days, they will be treated as one event.

5.2.2 Test Scores

In order to assess the performance of the event detection, a suitable method is required. Hence, a detection cost function is introduced, which is inspired by the detection cost function of [All02]. Since the normalized detection cost function is developed to assess the performance of TDT (topic detection and tracking) system, the cost function is defined in terms of probabilities of missed detection and false alarm errors P_{Miss} and P_{Fd}. The function is assigned with pre-specified C_{Miss} and C_{Fd} and a priori probability of a target P_{Target}, which means target topics discussed by a sequence of stories are annotated as targets and non-targets. If the test system fails to detect the targets, it will be labeled as missed detection; if the system detects the non-targets as targets, then it will be then judged as false detections; otherwise the results are accepted as a correct one. P_{Target} shows the percent of target events in the corpus. In other words, the proportion between the targets and non-targets in the corpus are represented by this value. The aim

[1]For example, http://en.wikipedia.org/wiki/May_2006, accessed on September 09, 2011

of this evaluation method is to build a global standard so as to compare the performance between different TDT systems.
According to the characteristics of the events in Wikipedia, the events in Wikipedia can not be easily segemented as the standard TDT corpus, because the event related editing will be made continuously after event occurrence, so the end of topics can not thus be defined clearly. As a result, our evaluation is focusing on the first story detection in each topic. The aim of evaluation is to compare the system performance under different detection factors instead of comparing the system performance with other similar detection systems, Therefore our cost function is:

$$C_{Det} = C_{Miss} \cdot P_{Miss} + C_{Fd} \cdot P_{Fd}$$

$$P_{Miss} = \frac{\left|Misseddetections\right|}{\left|Events\right|}$$

$$P_{Fd} = \frac{\left|Falsedetections\right|}{\left|Events\right|}$$

In this function, the performance of detection system is characterized by the probabilities of the missed detection and false detection rate (P_{Miss} and P_{Fd}). These two probabilities are also linearly combined into one cost function by assigning costs to missed detection and false detection error. Different from [All02]'s function, the P_{target} is not used, since the abundance of topics and their stories in Wikipedia can not be clearly defined and measured. Instead, the extracted benchmarks (number of events) are used as test domain, so that the P_{Miss} and P_{Fd} can be calculated on equal event amount under the same conditions. As described in the two formulations, the P_{Miss} and P_{Fd} are calculated by the number of missed detections and false detections divided by the sum number of events. In addition, according to the concept and system uses, the missed detection should be heavily punished in contrast to the false detection, since the most important feature of a detection system is to find the related events as many as possible, which ensures the high recall and provides the possibility for further improvement on the detection precision. If an event is missed in detection, the improvement of the precision can never be conducted. Hence in our function, the constant C_{Miss} and C_{Fd} are set to 1 and 0.1 respectively so that the factors of high precision and a low missed detection rate can be stressed.

The results of time beginning detection using different factors are compared with the benchmarks. If the difference between detection time and benchmark

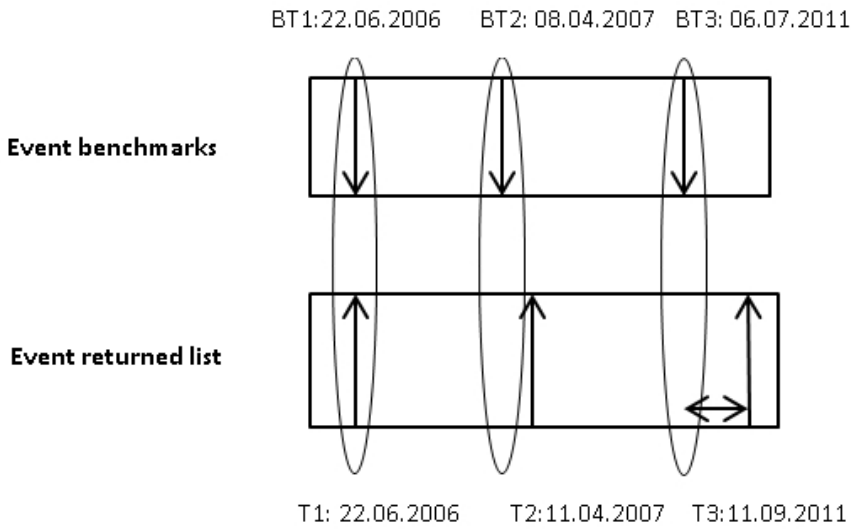


Figure 5.8: Evaluation Methods

time is less than two days, the detection results will be accepted to be correct. The false detections are those results which shows a time difference larger than two days but under one month; otherwise the results will be labeled as missed detections. That is to say, there are two types of missed detection situation: the first type is the returned result has a time difference larger than one month which deviates too much from the benchmark and can be seen as two independent time points. The other type is the detection with no detection results which is also classified into missed detection. The following example describes the calculation of cost function: as shown in Figure 5.8, the detection results are compared with the event benchmarks, There are three benchmarks in this page, so the three detected events are picked out according to the ranking rule(editing intensity). In other words, the three most edited events are chosen and compared with the benchmarks. The comparison indicates that, the event T1 is matched with benchmark BT1, the event T2 has a three day distance from the benchmark BT2, and the event T3 deviates from the benchmark BT3 for more than 2 months, so the calculation is $C_{Det} = 1 \cdot \frac{1}{3} + 0.1 \cdot \frac{1}{3} = 0.433$. If the returned result falls into the interval of more benchmarks, the best matched result is used. In this case, one result can only match one benchmark.

Test Type	C_{Det}	P_{Fd}	P_{Miss}
Time Interval	0.3694	0.2245	0.3470
Content Distance	0.4326	0.2448	0.4081
Link Intensity	0.5061	0.1633	0.4898
Time Interval and Content Distance	**0.3653**	**0.1836**	**0.3469**
Time Interval and Link Intensity	0.5490	0.1836	0.5306
Content Distance and Link Intensity	0.6041	0.1224	0.5714
Time interval and Content Distance and Link Intensity	0.6612	0.0816	0.6531
Content distance with χ^2	0.4878	0.18367	0.46939
Time Interval and Content Distance ranked with Sum of Content Distance	0.5224	0.1224	0.5102
Time Interval and Content Distance ranked with Sum of External Links	0.5857	0.1428	0.5714

Table 5.1: Cost Function for Detection Factors and Their Combinations I

Test Type	C_{Det}	P_{Fd}	P_{Miss}
Derivative of Time Interval and Derivative of Content Distance	0.4490	0.2041	0.4286
Value of Time Interval and Value of Content Distance	0.3326	0.2653	0.3061
Value and Derivative of Time Interval and Value of Content Distance	**0.2979**	**0.3265**	**0.2653**
Value and Derivative of Content Distance and Value of Time Interval	0.4469	0.1836	0.42857

Table 5.2: Cost Function for Detection Factors and Their Combinations II

5.2.3 Result Analysis

The detection of the event beginning using different factors and their combinations are evaluated by the cost function 5.2.2. 49 events in 18 pages are tested with the evaluation method introduced in the last section. The smaller cost value shows a better detection effectiveness.
In first step, the detection with the three factors defined in Chapter 3.5.3 is performed, the result is shown in Table 5.2.2. The combination of the time interval and the content distance has achieved the lowest detection cost, which is followed by the single time interval. At the third place is the detection with factor content distance. The detection with factor link intensity and their combinations generally lead to worse results than the previous two factors, which suggests factor link intensity has hurt the general precision of event beginning detection results. One of the reasons is the problem of external links in timeliness: some websites report the event related news with a long time delay, and the other reason is the timeliness of the adding of external links by the Wikipedia users. In some cases, the editing of the event related external links are postponed for quite a long time after the event beginning. As a result, the factor external links is more likely to be used as an external approvement for event, whereas the combination of factor time interval and content distance has the highest sensitivity to detect event beginning. The factor content distance is more editing-oriented and it shows the exact trends of public attention on certain page.

The second step of testing aims to find a better factor, which brings lower detection cost than the factors in the first step. Hence the "test winner" from first step, namely time interval and content distance, is further factorized and combined to be new factors, which can be seen in the first column of Table 5.2.2, the combinations of the derivative and value of the content distance and time interval are further tested on the same data set (see Appendix A.1) as the first step. As shown in Table 5.2.2, the new factor, namely value and derivative of time interval and value of content distance, has achieved the lowest detection cost (0.29) with 0.2653 missed detection and 0.3265 false detection. There are two reasons why content distance and time interval are chosen to conduct the optimization, first is this factor combination has won the test in first step, which arouses the interest of further optimization. Second, the bad performance of factor link intensity in first step has given enough evidence to exclude the link intensity and its combinations in second step and the detection with only time interval or content distance has also

been proved to have no clear enhancement in the test, so these two factors are intuitively chosen to perform the further test on the combinations of the sub factors.

In order to test the other on hand ranking factors, the detection factors with the lowest detection cost are used to run with the new ranking factors, the sum of content distance value and the sum of the positive link intensity are empolyed as weighting factors. The large content distance value or link increasement indicates the high degree of change, it might reflect the order of event in a more edit-oriented way. Yet, as shown in Table 5.2.2, the two new ranking approaches have brought no positive influence on detection result and the detection cost has doubled by using new ranking factors. From this standpoint, the editing intensity is the most suitable ranking factor for the current detection approach.

Moreover, it is noteworthy that, the biggest event in each page can always be detected with high precision using the time interval and content distance. For example, the tsunami earthquake in Japan on *March 11, 2011*, dual attacks in norway on *July 22, 2011* or the plagiarism of Guttenberg. These big events are all detected without missing. Normally, these events are also ranked at the first place according to the editing intensity which reflects the positive correlation between real world events and the Wikipedia editing intensity. However, some of the event benchmarks can be detected by none of the factor combinations. There are two reasons for this situation. One is the Wikipedia page has actually not included this event. The other lays on the ranking of this event, which is conducted according to the editing intensity is too low. Although the event has already been detected, it can not be included in the return list for the low editing intensity. The first reason is the page content itself and it is related to the page classification which will be further discussed in the following sections. As far as the second reason is concerned, a better ranking mechanism based on machine learning and the NLP approach should be introduced, so that the target event can be returned at first place according to the page classification and more detailed event related features from the revision content.
Based on the test scores of event detection, the following problems are further discussed.

5.2.3.1 User Editing and Event Page Classifications

According to the test results and the test set, which have been used, page editing is varied in forms. In terms of our concept, the event detection is based on the user editing. However, not all the user editings are suitable for the real world event detection. The type of page editing and suitable editing types for the event detection is discussed as follows. Except the one-time or whim editing and vandalism, most editing is aroused by an event or caused by change in public attentions.

On the basis of the test results and observations, the editing relating to these events is made in different time sequences. For instance, most events related editings will be conducted soon after event occurrence, which are called synchronous editing. As the opposites, there are also three types of asynchronous editing, which are conducted at a time differing from the event occurrence. The first type of asynchronous editing is delay editing and random editing, which is postponed with a certain time length after the event occurrence. In our dataset there are two typical asynchronous pages *Madrid Atocha railway station* and *London Buses route 30*, which are related to the 2004 Madrid train bombings and 2005 London bombings. However, these two pages are not timely updated to present the event occurrence or development. The event related content is not added until after quite a long-time delay. The detection of the real world event beginning time using these two pages are proved to be completely inaccurate. The result can only reflect the editing activities on these two pages. The second type is periodical editing. For some event related page, editing is made near the anniversary of the event in a periodical way. The extension and additional explanation to the event are continuously be added to the page for commemorating or memorializing. The last type is named as associate supplementary editing, which is mainly related to the situation when the old event content is edited during the occurrence of new events of the same kind. For example, the page of *Megathrust earthquake* was edited when 2011 To-hoku earthquake and tsunami struck on *March 11, 2011*, yet the editing concerns not only To-hoku earthquake but also previous earthquakes including 2010 Maule Earthquake and 2004 Indian Ocean Earthquake. In this way, some events can lead to a series of associate asynchronous editings on previous content which are related to the extension of background information for current event or views on similar perivous events.

However, concerning the real world event detection based on the editing event, only the synchronous editing can be directly used, for the asynchronous

Statistical methods	**AD or Goodman**	χ^2
Guttenberg page: 488 revisions	313 results >0	196 results >0
Fukushima page: 256 revisions	104 results >0	97 results>0
Daiichi page: 1384 revisions	865 results>0	526 results>0

Table 5.3: Statistical Methods Comparison

editing generates different kinds of time delays and deviations, which hurt the precision of the event beginning detection and content tagging as well as make it impossible to be used as the time evidence for further reasoning.

5.2.4 Detection with different Statistical Methods

In this section the results of the comparison with three different statistical methods including χ^2 (Chi-squared), Anderson-Darling, Goodman are presented and analyzed. The aim is to find a suitable method for the content distance comparison, which is highly effective in detecting the newly emerging vocabulary as well as discovering the differences between two distributions. For a start, the test results on single page *Karl-Theodor zu Guttenberg,Fukushima Prefecture* and *Fukushima Daiichi Nuclear Power Plant* are presented in Table 5.3, the first column of the table shows the revision number of each page, the second and the third column represent the non-zero valid comparison results which returned from Anderson, Darling Goodman and χ^2 respectively. It is clear to see that Anderson Darling and Goodman return more non-zero valid results than χ^2.
According to the visulizations in Figures 5.9, 5.10 and 5.11, Goodman and Anderson Darling have higher sensitivity. Hence the first two comparison methods yield more details and minor changes between revisions, which are different from χ^2 that pays more attention to big change and ignores some of the minor editings. As a result, Goodman and Anderson Darling are the better choice for the detection of multiple events with some small changes and details between contents. To detect the big difference and single event, the χ^2 method can ignore some of the noise automatically and fulfill the requirement. In order to further test the detection effectiveness with the two methods, Goodman and χ^2 are chosen to perform the detection task on the same dataset (see Appendix A.1). The content distance is naturally selected as detection factor.
Based on the detection results in table 5.2.2, the event detection with χ^2

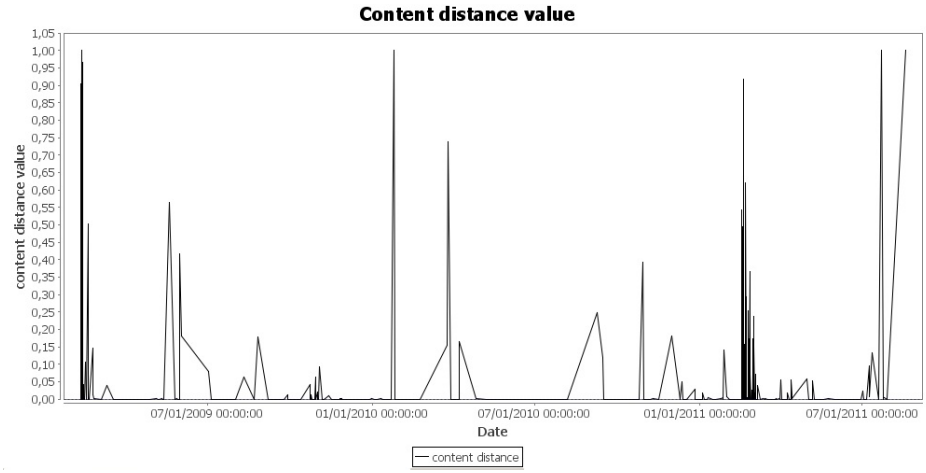


Figure 5.9: Statistical Comparison with Goodman

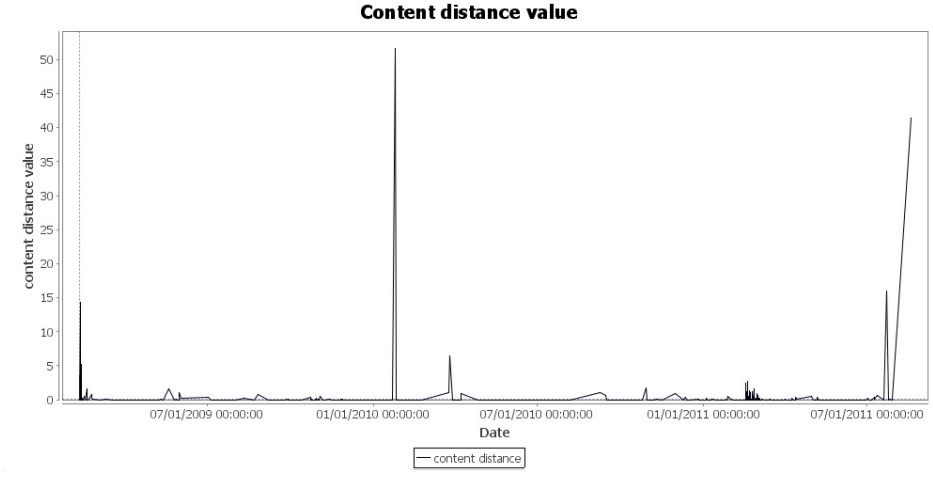


Figure 5.10: Statistical Comparison with Anderson Darling

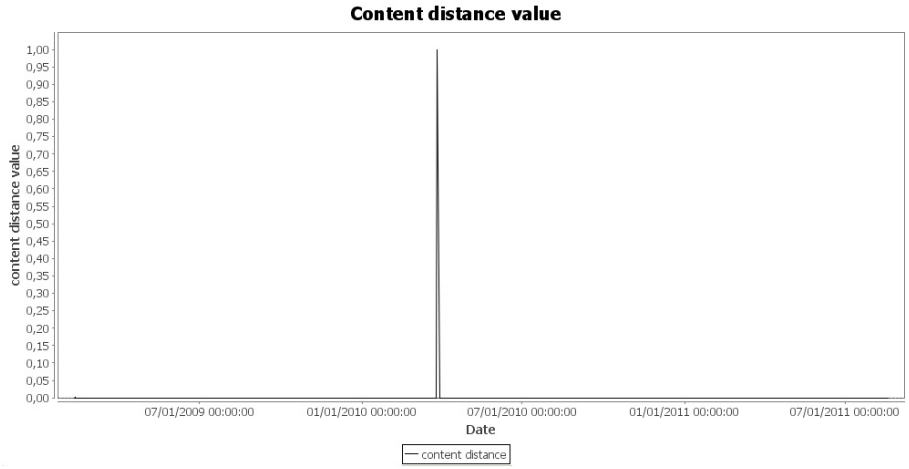


Figure 5.11: Statistical Comparison with χ^2

shows a 0.05 (about 12.7%) higher detection cost than the Goodman. With a comparison of the difference between each benchmark, it is clear that the detection of a single event pages with the two methods returns the same results, and the difference comes mainly from the detection of subordinate events in multiple events situations, For example, on the page of *Dominique Strauss-Kahn*, the biggest event is detected correctly with both methods. Yet in detection of second benchmark on *July 28, 2011*, the χ^2 method has missed to deliver the correct event beginning time; instead, it returns another detected event beginning according to the ranking method (editing intensity). On the opposite, Goodman returns the results with deviation on *July 1, 2011*, which is found to be the editing activities with a short delay. Although Goodman has returned a slightly better result, but χ^2 has also detected all significant events in dataset with fewer non zero result points, and the missed detections are due to the ranking methods using link intensity, since the event on *July 28, 2011* has also been detected in the event results instead of at the right place which fails to match the benchmark. From this perspective, there is no obvious answer to the question of "which statistical method is the best for event detection". The answer effectively depends on the concrete detection situation and different ranking methods; at the same time, pre-/post processing should be taken into consideration as well. However, according to our evaluation result, the detection of multiple events page with large number of editing details in Wikipedia event detection scenario, Anderson Darling and Goodman are naturally the right choice.

5.2.5 Configuration of the RAKE Algorithm

Due to the exponential growth in the time consumed for keywords extraction with RAKE, the scope of application of our prototype is restricted to short pages with small amount of revisions (below 400 revisions). The most time consuming part of RAKE is the establishment of co-occurrence matrix (see Appendix A.1), which takes nearly 13 minutes for one revision with about 2000 words. For a page with 3000 revisions, it is impossible to test the complete revision history on a single personal computer. Therefore we decide to establish the bag of words model by using all the tokens in one revision instead of using RAKE keyword extraction, so that the bottleneck of the keyword extraction can be avoided. An incremental algorithm is also adopted to establish the histogram. Only the newly emerging keywords are added to the bag vocabulary. As depicted in Figure 5.13, in contrast to the establishing histogram with RAKE, the time consumed in histogram establishment with the pure bag of words model is much smaller and the growth curve of the time consuming also gets flatter. After improvement, the comparison efficiency shows the

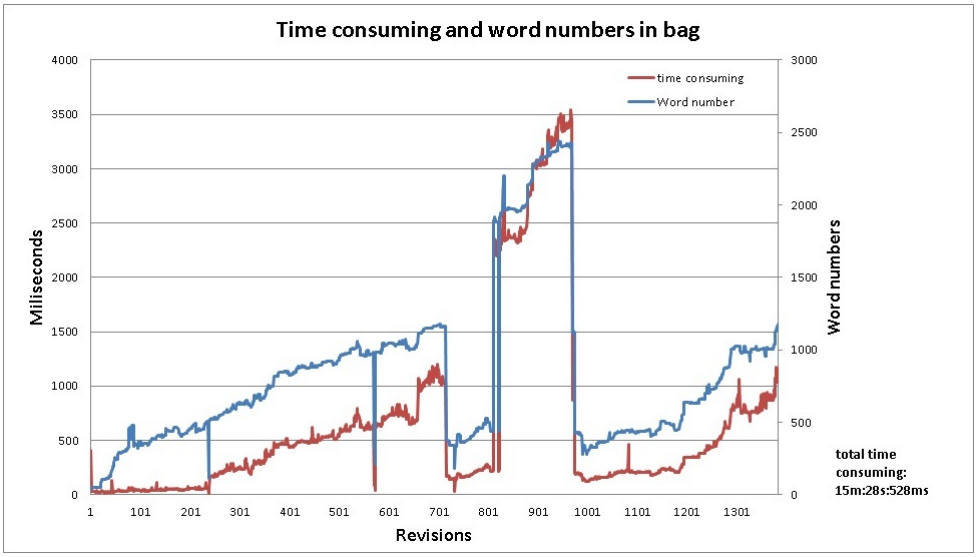


Figure 5.12: Time Consumed with only Bag of Words

remarkable enhancement. For example, page *Tsunami* with 6935 revisions can be analyzed in 4h:14m:2s:374ms, which offers a better possibility of conducting the analysis on generic Wikipedia pages. As can be seen in Figure 5.12, the time consumed for a page with 1384 revisions is 15m:28s:528ms, and the maximum single revision time consumed is near 3500ms. The blue

curve represents the total word number in each revision whereas the red curve stands for the time consumed for the corresponding revision. Moreover, according to the similar trends of changes of the two curves, the word number and time consumption are proved to be positively correlated. The sharp decrease of time consumed during the histogram establishment is due to the deletion editing or the page decomposition, which leads to the generation of new pages based on a certain part of the current page. As can be seen from

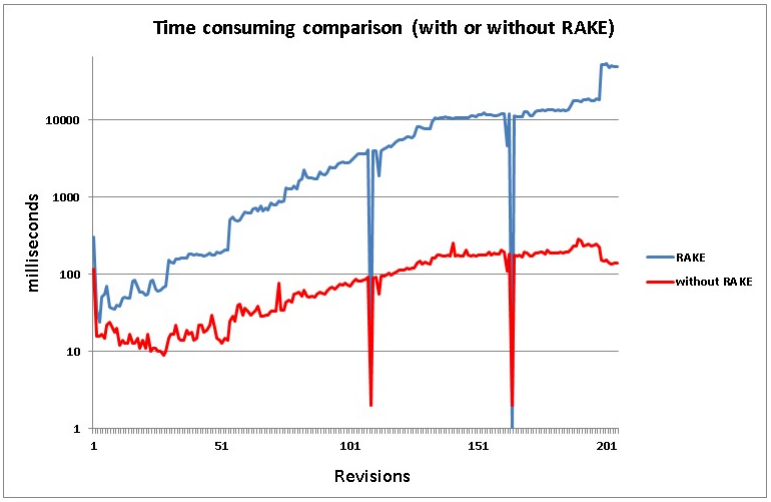


Figure 5.13: Comparison of RAKE and Bag of Words

Figures 5.14 and 5.15, the differences between the detection with RAKE and with direct bag of words model are clear: with the direct bag of words model more editing events have been detected, while detecting with RAKE word change in the same situation is not considered as event, because some of the word change have not really influenced the keywords ranking.

From the shape of the curve for the content distance values, it is clear to see that, detection with direct bag of words model is more easily influenced by the token change than RAKE, because all the word tokens are used as statistical objects and the change of the vocabulary in the revision can be detected at the very beginning of the token change. In contrast to direct bag of words model, detection with RAKE is more content-oriented, and only the keywords changes will be accepted as new comparison objects. Therefore the change peaks come always later than detection methods with direct bag of words model.

In reference to our concept, the content distance is used as one of the features

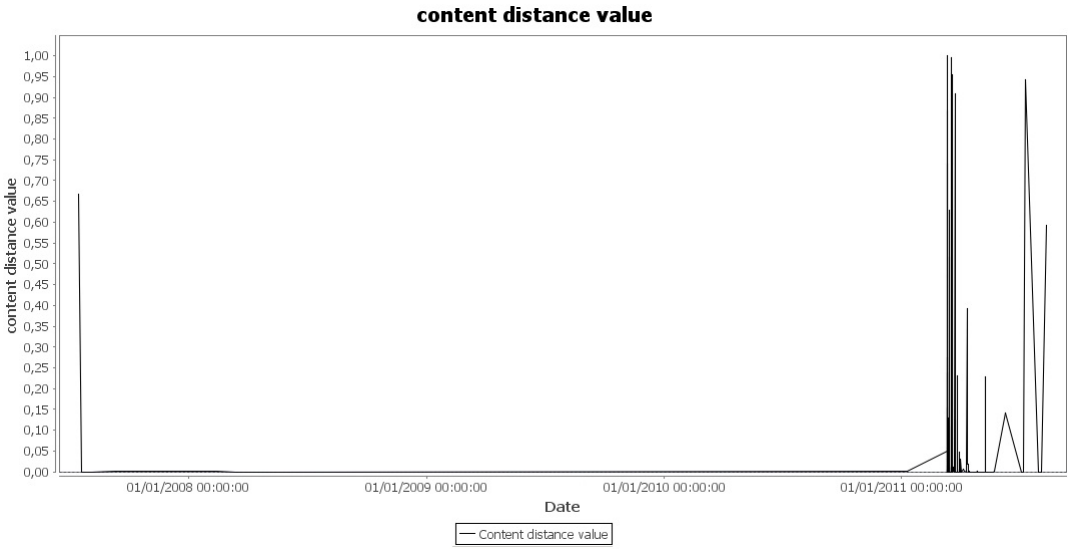


Figure 5.14: Detection without RAKE

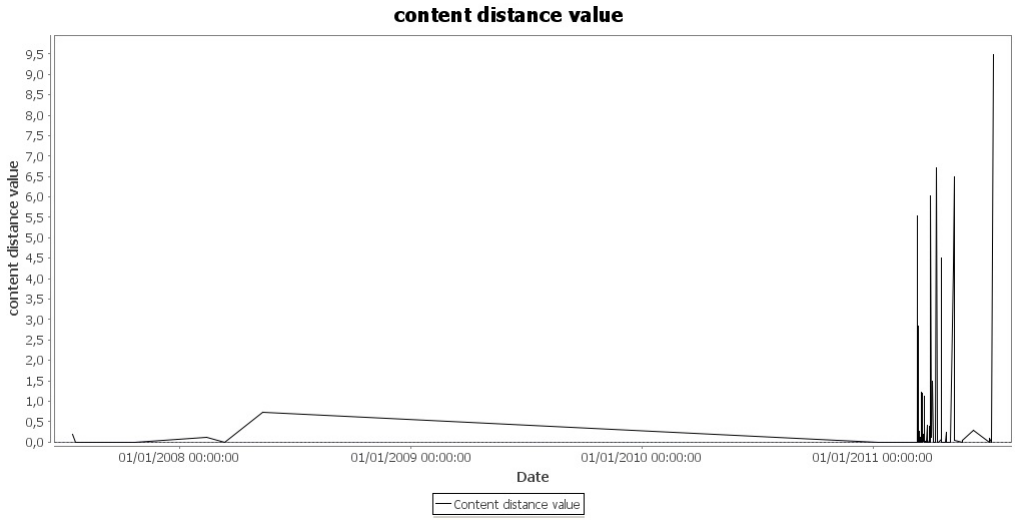


Figure 5.15: Detection with RAKE

to determine the event boundary. Through the comparison results, it can be proved that detection without RAKE can also deliver the right event boundary with a high precision.

Chapter 6

Conclusions and Future Work

In this thesis, the change in Wikipedia has been analyzed. Based on the change analysis, the event detection using different change factors is performed, the detection results and corresponding evaluation are presented.

- Chapter 1 gives a basic introduction to the history and principles of wiki system and Wikipedia followed by a description of Wikipedia's features and uses. After the explanation of the editing history mechanism in Wikimedia in Section 1.1, the motivations and research questions are presented in Section 1.2.
- Chapter 2 describes the technologies and theories used in this thesis. Current research progresses and results related to Wiki changes tracking are categorized and compared in the section of state of the art 2.5.
- Chapter 3 presents the concepts of design for our prototype. The basic architecture of our prototype and important components are described and illustrated in detail. The analyzing procedures such as page crawling, parsing, features extraction and comparison are explained. Based on the extracted features, the language model is established for each revision using RAKE keywords extraction and the bag of words model, which is introduce in Section 3.5, so that the content distance can be detected and measured by statistical methods. With other two features (time interval, link intensity) the approach 3.5.3 for event detection is presented.
- Chapter 4 explains the implementation details of the prototype step by step, which mainly focuses on the technical details of important

components, methods and interfaces, for example, wiki page crawling using Wikimedia crawler based on Palladian and Wikipedia Parsing using JWPL (see Section 4.2), histogram generation with JAIDA (see Section 4.4.1). Last but not least, the visualization of extracted features using the Jfreechart which described in Section 4.4.5 is also described.

- Chapter 5 is the evaluation part, which provides the assessment of the implementation results and the validation of the design in concept (see Section 5.1). The detection results based on different factors and statistical methods configurations are compared and analyzed in Section 5.2. The effectiveness and the efficiency of different modeling configurations are also tested in Section 5.2.5.

- The sixth chapter gives the summary and outlook for the thesis. Within the summary, the main aspects of each chapter are described. And the future work is expected to focus on the extension possibilities and improvements in different aspects.

6.1 Main Results

The research results of the questions raised in Chapter 1 and the extra knowledge obtained from the experiments are briefly summarized as follows:

1. Through our prototype, the inner structure of Wikipedia and change types are modeled. Besides, the content distance, time interval as well as link intensity between revisions have also been quantized. The change based on these three factors can therefore be localized and visualized as needed. Moreover, the change of the other features at each level corresponds to the timeline can also be further analyzed by various methods. Our prototype provides not only more details and flexibilities than the diff function offered by Wikimedia, but also gives an overview of the entire change history which helps us to grasp the trends of the change at different levels.

2. The event detection with different factors has different effectiveness and application scopes. According to the test results in the Chapter 5, the factor combination value of content distance and the derivative and value of time interval has the highest effectiveness in detecting the

event beginning ($C_{Det} = 0.2979$, $P_{Fd} = 0.3265$, $P_{Miss} = 0.2653$), which means about 74% of the events in test data set have been detected and about 68% of these detected events (74%) are recognized correctly. As for the characteristics of the single factors, the content distance represents the change inclined to user activities and represents content-orient fluctuation of revision. The detection with time interval has the highest sensitivity to detect the event occurrence, for the event-related editing always reflects on shortening of editing interval first. However, the detection with content distance always shows some delay in contrast to time interval detection, as the content change needs some time to accumulate. The last factor link intensity has also its special time and sensitivity features, because external links can be seen as the external confirmation of event from other web medias, it appears with even larger delay than the content distance, however, it brings naturally more evidence to approve the event occurrence. Based on these features, the selection of event detection factors should consider the specific event types and the types of analysis, which will further be performed on the output of event detections. For example, if detection of the beginning time of the editing event is the main purpose, the suitable choice is using single factor time interval; if the hot spot event of the public attention is needed and further NLP analysis based on the content will be applied for further investigation, the factor content distance is then the best choice. Yet if some external confirmation about certain event becomes the focus of event detection, factor link intensity is the most suitable choice.

3. The editing behavior of users in Wikipedia is found to be merely partly synchronous with the real world event. As can be seen in the evaluation, some of the Wikipedia pages are not been edited or timely updated during the event occurrence. Editing is delayed at times at arbitrary length. Apparently the asynchronously edited pages of this kind are not suitable for the real world event detection. Only the pages with the event-synchronous editings can be used as data sources of the event detection. From this standpoint, Wikipedia is more likely to be treated as a history of public attention on certain events, or the development record of mass viewpoints; at the same time, it obtains the evolution of conclusive comments from Wikipedia users, which is quite different from normal news portals.

4. By the establishing of a language model for revisions, the keyword

extraction method RAKE is found to be an unsupervised, domain-independent method for extracting keywords from individual documents. Although RAKE performs well on the short plain texts from Wikipedia page at first, the establishment of co-occurrence matrix in RAKE leads to exponential growth in the time consumption and makes it impossible to process long pages with a large number of revisions on single a personal computer. Experiment and configurations come out that first story detection with only the bag of words model shows the same degree of precision as the detection with RAKE based on the current data set and the time consumption with the bag of words is also far lower than the time consumption with RAKE.

6.2 Future Work

1. In event detection, the current ranking for event candidates is using a rule based mechanism which grounds on the the three factors and their derivatives as well as the editing intensity. In some situations, the ranking mechanism is unable to detect the noisy editing before real event beginning, or target event with small editing intensity. Hence an alternative ranking mechanism could employ further NLP methods and the machine learning approach to extract the event related named entities through an extraction component, so that the event related time, name and places can be found and the revisions are weighted according to the metrics derived from the features of these extracted entities. Based on this weighting value, the revision can then be ranked according to the target event. Furthermore, a corresponding annotated feature set for event in Wikipedia is required, which offers a start point for machine learning approach.

2. A link based detection approach, which is based on page ranking algorithm and clustering approach is another possible extension point. As is known, each revision has its own internal and external links. Based on the outgoing links, which shows the importance of the revision, a numerical weighting can be assigned to each revision with the purpose of measuring the relative importance within the set. And the changes in the numerical weighting together with the linkage situation reveal the new appearance of certain links, which further indicate the occurrence of certain events.

3. Moreover, the research on social property of Wikipedia is also a possible extension point to the Wikipedia change tracking, for one user can edit more than one page and one page can be edited by an arbitrary number of users. All pages and users can be considered as nodes in this network. If we connect all these nodes with edges which indicates the editing action of users on certain pages such as deletion, insertion and reversion, for example, the frequency of the action according to the editing types or the count of each type of the actions could be used as the weighting factors. All the features can be extracted from the revision history. Since the editing change can also influence the shape of the networks and their weights. Reversely, if the shape of certain networks are traced, the change in Wikipedia pages can be detected as well.

Bibliography

[All02] James Allan. *Topic detection and tracking: event-based information organization*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[AOH04] Taweesup Apiwattanapong, Ro Orso, and Mary Jean Harrold. A differencing algorithm for object-oriented programs. In *In Proceedings of the 19th IEEE International Conference on Automated Software Engineering ASE 2004*, pages 2–13, 2004.

[CL96] Jim Cowie and Wendy Lehnert. Information extraction. *Commun. ACM*, 39:80–91, January 1996.

[CN10] Marek Ciglan and Kjetil Nørvåg. Wikipop: personalized event detection system based on wikipedia page view statistics. In *CIKM*, pages 1931–1932, 2010.

[CSSE10] Si-Chi Chin, W. Nick Street, Padmini Srinivasan, and David Eichmann. Detecting wikipedia vandalism with active learning and statistical language models. In *Proceedings of the 4th workshop on Information credibility*, WICOW '10, pages 3–10, New York, NY, USA, 2010. ACM.

[Dbp] Sir tim berners-lee talks with talis about the semantic web. `http://talis-podcasts.s3.amazonaws.com/twt20080207_TimBL.html`, retrieved on July 9, 2011.

[FBA10] Peter Kin-Fong Fong and Robert P. Biuk-Aghai. What did they do?: deriving high-level edit histories in wikis. In Phoebe Ayers and Felipe Ortega, editors, *Int. Sym. Wikis*. ACM, 2010.

[Fpw] Five pillars of wikipedia. `http://en.wikipedia.org/wiki/Wikipedia:Five_pillars`, accessed on July 11, 2011.

[GG07] Jakub Gawryjołek and Piotr Gawrysiak. The analysis and visualization of entries in wiki services. In *Advances in Intelligent Web Mastering*, pages 118–123. Springer Berlin / Heidelberg, 2007.

[Gil05] Jim Giles. Special report internet encyclopaedias go head to head, December 2005. `http://www.nature.com/nature/journal/v438/n7070/full/438900a.html`, accessed on July 11, 2011.

[Glo06] P. A. Gloor. Swarm creativity : Competitive advantage through collaborative innovation networks. *Oxford University Press, USA*, January 2006.

[Gom] Google-diff-match-patch api. `http://code.google.com/p/google-diff-match-patch`, accessed on July 9, 2011.

[Hec78] Paul Heckel. A technique for isolating differences between files. *Commun. ACM*, 21:264–268, April 1978.

[Hgp] Project page of history flow. `http://www.research.ibm.com/visual/projects/history_flow/index.htm`, accessed on July 9, 2011.

[HSLA09] Sebastian Hellmann, Claus Stadler, Jens Lehmann, and Sören Auer. Dbpedia live extraction. In *Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part II*, OTM '09, pages 1209–1223, Berlin, Heidelberg, 2009. Springer-Verlag.

[KK08] Aniket Kittur and Robert E. Kraut. Harnessing the wisdom of crowds in wikipedia: quality through coordination. In *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, CSCW '08, pages 37–46, New York, NY, USA, 2008. ACM.

[LBN10] Dustin Lange, Christoph Böhm, and Felix Naumann. Extracting structured information from wikipedia articles to populate infoboxes. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1661–1664, New York, NY, USA, 2010. ACM.

[MI04] Y. Matsuo and M. Ishizuka. Keyword extraction from a single document using word co-occurrence statistical information.

International Journal on Artificial Intelligence Tools, 13:2004, 2004.

[MWB10] Jacob Kogan Michael W. Berry, editor. *Text Mining: Applications and Theory*. Wiley, 2010.

[NRD08] Sérgio Nunes, Cristina Ribeiro, and Gabriel David. Wikichanges: exposing wikipedia revision activity. In *Proceedings of the 4th International Symposium on Wikis*, WikiSym '08, pages 25:1–25:4, New York, NY, USA, 2008. ACM.

[PMD+05] M. Pia, B. Mascialino, S. Donadio, S. Guatelli, A. Pfeiffer, et al. A statistical toolkit for data analysis. pages 381–383, 2005.

[PSG08] Martin Potthast, Benno Stein, and Robert Gerling. Automatic vandalism detection in wikipedia. In *Proceedings of the IR research, 30th European conference on Advances in information retrieval*, ECIR'08, pages 663–668, Berlin, Heidelberg, 2008. Springer-Verlag.

[SCKP08] Bongwon Suh, Ed H. Chi, Aniket Kittur, and Bryan A. Pendleton. Lifting the veil: improving accountability and social transparency in wikipedia with wikidashboard. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1037–1040, New York, NY, USA, 2008. ACM.

[SGV08] K. Smets, B. Goethals, and B. Verdonk. Automatic vandalism detection in wikipedia: Towards a machine learning approach. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy (WikiAI08)*, pages 43–48. AAAI Press, 2008.

[Sot09] Jos´e Felipe Ortega Soto. *Wikipedia: A quantitative analysis*. PhD thesis, Universidad Rey Juan Carlos Madrid, Spain, 2009.

[Spe] Special export interface of wikipedia. `http://en.wikipedia.org/wiki/Special:Export`, accessed on July 7, 2011.

[Van] Vandalism example. `http://www.flickr.com/photos/erikrasmussen/3195274862/`, accessed on July 9, 2011.

[Vos05] Jakob Voss. Measuring wikipedia. In *Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics*, Stockholm (Sweden), July 2005.

[VWD04] Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 575–582, New York, NY, USA, 2004. ACM.

[WIPR10] Qinyi Wu, Danesh Irani, Calton Pu, and Lakshmish Ramaswamy. Elusive vandalism detection in wikipedia: a text stability-based approach. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 1797–1800, New York, NY, USA, 2010. ACM.

[WKL10] Andrew G. West, Sampath Kannan, and Insup Lee. Stiki: an anti-vandalism tool for wikipedia using spatio-temporal analysis of revision metadata. In *Proceedings of the 6th International Symposium on Wikis and Open Collaboration*, WikiSym '10, pages 32:1–32:2, New York, NY, USA, 2010. ACM.

[Wma] Wikimedia api. `http://www.mediawiki.org/wiki/API:Main_page`, accessed on July 9, 2011.

[ZMG08] Torsten Zesch, Christof Müller, and Iryna Gurevych. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *Proceedings of the Conference on Language Resources and Evaluation (LREC)*, 2008.

List of Figures

List of Tables

Appendix A

A.1 Dataset

The chosen datasets are:

- Fukushima Prefecture
  Fukushima Prefecture is a prefecture of Japan located in the To-hoku region on the island of Honshu. The capital is the city of Fukushima. The 2011 Great East Japan Earthquake, the tsunami that followed, and the resulting Fukushima I Nuclear Power Plant disaster caused significant damage to the prefecture. `http://en.wikipedia.org/wiki/Fukushima_Prefecture`

- Onagawa Nuclear Power Plant
  The Onagawa Nuclear Power Plant is a nuclear power plant located on a 1,730,000 m2 (432 acres) in Onagawa in the Oshika District and Ishinomaki city, Miyagi Prefecture. On April 8, 2011, a leak of radioactive water spilled from pools holding spent nuclear fuel rods following the 2011 To-hoku earthquake. `http://en.wikipedia.org/wiki/Onagawa_Nuclear_Power_Plant`

- Fukushima Daini Nuclear Power Plant
  The Fukushima II Nuclear Power Plant, is a nuclear power plant located on a 1,500,000-square-metre (370-acre) sitem,After the 2011 To-hoku earthquake and tsunami, the four reactors at Fukushima II automatically shut down. `http://en.wikipedia.org/wiki/Fukushima_Daini_Nuclear_Power_Plant`

- Mount Lokon

Mount Lokon, together with Mount Empung, is a twin volcano (2.2 km/1.4 mi apart) in the northern Sulawesi, Indonesia. The volcano erupted on 15 July 2011, forcing thousands of people to evacuate. http://en.wikipedia.org/wiki/Mount_Lokon

- London Buses route 30
  London Buses route 30 is a Transport for London contracted bus route in London, United Kingdom. The service is currently contracted to First Capital.On 7 July 2005 at 09:47, a Dennis Trident 2 double-decker bus, fleet number 17758, registration LX03 BUF, was involved in a terrorist attack perpetrated by Hasib Hussain, a bomb in whose rucksack exploded, killing 13 other passengers as well as himself. http://en.wikipedia.org/wiki/London_Buses_route_30

- Moscow – Saint Petersburg Railway
  The Moscow to Saint Petersburg Railway is a 649.7-kilometre railway running between the two largest Russian cities of Moscow and Saint Petersburg,On November 27, 2009 four cars from train No. 166 derailed while travelling between Moscow and St. Petersburg. The derailment was a terrorist act caused by the detonation of 7 kilograms TNT equivalent. On August 13, 2007 an intercity passenger train heading to St. Petersburg from Moscow derailed shortly before reaching Malaya Vishera after a bomb explosion. There were 30 injuries and no deaths. http://en.wikipedia.org/wiki/Moscow_-_Saint_Petersburg_Railway

- Pyeongchang
  Pyeongchang is a county in Gangwon province, South Korea. It is located in the Taebaek Mountains region, and is the most popular winter sports location in South Korea.On 6 July 2011, Pyeongchang was announced as the host city for the 2018 Olympic Winter Games. http://en.wikipedia.org/wiki/Pyeongchang

- Madrid Atocha railway station
  Madrid Atocha is the largest railway station in Madrid. On March 11, 2004, packed arriving commuter trains were bombed in a series of coordinated bombings, killing 191 people and wounding 1,800. http://en.wikipedia.org/wiki/Madrid_Atocha_railway_station

- Peter Häberle
  Peter Häberle is a German legal scholar, specialising in constitutional law. He supervised the dissertation of Karl-Theodor zu Guttenberg.

Guttenberg's dissertation was later shown to contain copies of texts from different sources including Häberle himself. http://en.wikipedia.org/wiki/Peter_Häberle

- Dominique Strauss-Kahn
  Dominique Gaston André Strauss-Kahn, often referred to in the media as DSK, is a French economist, lawyer, and politician, and a member of the French Socialist Party, Managing Director of the International Monetary Fund from 2007 to 2011, In May 2011, Strauss-Kahn was arrested in New York City and charged with the sexual assault of a housekeeper who entered his Sofitel hotel suite, which leds to his resignation of IMF Director. http://en.wikipedia.org/wiki/Dominique_Strauss-Kahn

- Fukushima Daiichi Nuclear Power Plant
  The Fukushima Dai-ichi Nuclear Power Plant is a disabled nuclear power plant located on a 3.5-square-kilometre (860-acre) site,The plant suffered major damage from the 9.0 earthquake and subsequent tsunami that hit Japan on March 11, 2011 and is not expected to reopen. http://en.wikipedia.org/wiki/Fukushima_Daiichi_Nuclear_Power_Plant

- Karl-Theodor zu Guttenberg
  Karl-Theodor Freiherr zu Guttenberg is a German politician of the Christian Social Union (CSU),the discovery and widespread criticism of extensive plagiarism in his doctoral thesis started from 16. February. 2011. http://en.wikipedia.org/wiki/Karl-Theodor_zu_Guttenberg

- Utøya
  Utøya is an island in the Tyrifjorden lake in Hole municipality, in the county of Buskerud, Norway. The island is 10.6 hectares (26 acres) situated 500 metres off the shore, by the E16 road, 38 kilometres (24 mi) driving distance north-west of Oslo city centre.On 22 July 2011,after car bomb explosion in Regjeringskvartalet at 15:25:19 (CEST), a mass shooting took place at the AUF's summer camp. http://en.wikipedia.org/wiki/Ut%C3%B8ya

- Workers' Youth League (Norway)
  Workers' Youth League is the youth organization affiliated with the Norwegian Labour Party. On 22 July 2011 an AUF camp at Utøya was the scene of a massacre carried out by a right-wing terrorist dressed up as a police officer http://en.wikipedia.org/wiki/Workers%27_Youth_League_(Norway)

- Leopold Cafe
  The Leopold Cafe is a large and popular restaurant and bar on Colaba Causeway, in the Fort area of Mumbai, India, located across from the Colaba Police station.The cafe was an early site of gunfire and grenade explosions during the 2008 Mumbai attacks by terrorists(26.November.2008). The restaurant was extensively damaged during the attacks. http://en.wikipedia.org/wiki/Leopold_Cafe

- Tim Cook
  Timothy D. Tim Cook is the chief executive officer of Apple Inc. having joined the company in March 1998.His primary responsibility is managing day-to-day operations at the company. He was named CEO after Steve Jobs announced his resignation on August 24, 2011. http://en.wikipedia.org/wiki/Tim_Cook

- Lech Kaczynski
  Lech Aleksander Kaczynski (18 June 1949 – 10 April 2010) was the President of Poland from 2005 until his sudden death in 2010 http://en.wikipedia.org/wiki/Lech_Kaczy%C5%84ski

- Goran Hadžic
  Goran Hadžic born on 7 September 1958 is a former president of the Republic of Serbian Krajina who was in office during the Croatian War of Independence. He is accused of crimes against humanity and of violation of the laws and customs of war by the International Criminal Tribunal for the former Yugoslavia. http://en.wikipedia.org/wiki/Goran_Had%C5%BEi%C4%87

- Patrick Tracy Burris
  Patrick Tracy Burris (August 8, 1967 – July 6, 2009) was an American spree killer responsible for at least five known murders in Cherokee County, South Carolina in 2009 http://en.wikipedia.org/wiki/Patrick_Tracy_Burris

- Wolfgang Schneiderhan(general)
  Wolfgang Schneiderhan (born 26 July 1946) is a German general who served as Chief of Staff of the Bundeswehr, the German armed forces, from 2002 to 2009.
  http://en.wikipedia.org/wiki/Wolfgang_Schneiderhan_(general)

- Foxconn
  The Foxconn Technology Group is a multinational business group anchored by the Hon Hai Precision Industry Co., Ltd. `http://en.wikipedia.org/wiki/Foxconn`

- Times Square
  Times Square is a major commercial intersection in the borough of Manhattan in New York City, at the junction of Broadway and Seventh Avenue and stretching from West 42nd to West 47th Streets. `http://en.wikipedia.org/wiki/Times_Square`

- Sendai
  Sendai is the capital city of Miyagi Prefecture, Japan, and the largest city in the To-hoku Region. In 2005, the city had a population of one million, and was one of Japan's 19 designated cities. `http://en.wikipedia.org/wiki/Sendai`

A.2 Time Consuming of RAKE

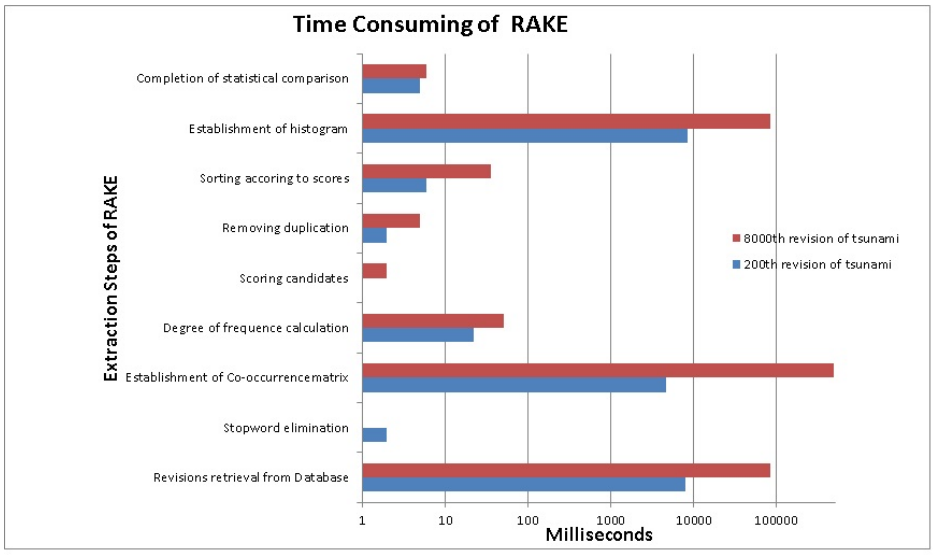


Figure A.1: Time Consuming of RAKE