

Automatic Wikibook Prototyping

Jen-Liang Chou, Shih-Hung Wu

*Department of Computer Science and Information Engineering,
Chaoyang University of Technology, Taiwan R.O.C.
{s9527633, shwu}@cyut.edu.tw*

Abstract: Wikipedia is the world's largest collaboratively edited source of encyclopedic knowledge. Wikibook is a sub-project of Wikipedia. The purpose of Wikibook is to enable a free textbook to be edited by various contributors, in the same way that Wikipedia is composed and edited. However, editing a book requires more effort than editing separate articles. Therefore, how to help users cooperatively edit a book is a new research issue. In this paper, we investigate how to automatically extract content from Wikipedia and generate a prototype of a Wikibook. Applying search technology, our system can retrieve relevant articles from Wikipedia. A table of contents is built automatically based on link analysis and. Our experiment shows that given a topic, our system can generate a table of contents, which can be treated as a prototype of a Wikibook.

Keywords: Wikipedia, Wikibook, table of content generation

Introduction

The ability to quickly construct a free encyclopedia such as Wikipedia has shown that the Web 2.0 has been successful. Wikipedia is useful in college education, both for general topics [6] and specific topics such as physics [7]. In this paper, we focus on another project of Wikimedia Foundation: Wikibook, which is also useful in the classroom [12]. Wikibook provides free textbooks on the Internet via the Wiki system, letting global users edit the contents of textbooks. However, creating a book without supporting data is difficult. An expert or a system that can provide a general framework and useful references for a book is of much help. Thus, we propose an automatic Wikibook prototyping system that can pick relevant articles from Wikipedia and output a hierarchy as the table of contents (TOC) for a given topic. Our system consists of information retrieval, machine learning, and natural language processing technology.

As a first study, [16] uses the TOC and anchor text of a Wikipedia entry to form a TOC of a Wikibook. A relevant research area is Topic Maps. Topic Maps are analogous to the Table of Contents of a textbook. Users can realize and memorize the relevant concepts of a topic. "Topic Maps for learning" (TM4L) [3] is an application of Topic Maps. However, these methods rely on humans mostly without using information retrieval technology that can provide massive relevant information. Studies in the knowledge acquisition provide some hints. Semi-automatic methods can aggregate a domain ontology via Internet search [11].

1. Methodology

Our framework for Wikibook prototyping involves three modules. These modules can be replaced to fulfill the need of different requirements. For example, we might customize the system for different languages, users of different ages, or topics with different contexts. The first module is the preparation of a corpus. We can use the whole Wikipedia or certain language versions or subsets. The system then extracts and analyzes contents from the corpus. As a knowledge source, Wikipedia provides not only the content but also a lot of links to contexts, which are also valuable. The second module is a search engine. As we mentioned above, relevant topics can be found out not only from a full text keyword search, but also from links in Wikipedia. This module is very important. Our system searches relevant topics and their hyponyms and hypernyms using general information retrieval technology for hierarchical construction. We discuss this in more detail in the following sections. The third module is hierarchical construction. Given relevant topics, our system then generates a hierarchy. This hierarchy can be edited manually as the table of contents of a Wikibook. During the hierarchical construction, further searching might be necessary to find more relevant contents.

2. Implementation issues

Our methodology gives a general idea on how to generate a Wikibook automatically within a flexible framework. In the following sections, we discuss our system and experiments on computer science topics in the English version of Wikipedia as our corpus because it is an ample resource of information which is available for potential Wikibook editors. Also, there is more professional knowledge in Wikipedia than in general Websites.

2.1 Search Strategy

Our search strategy is an automatic iterative search. The system takes keywords as the search input and performs context searching via the standard information retrieval process. We use pseudo-relevance feedback as our searching algorithm. The resulting set from the first search will be used as the corpus for the second search.

The system then extracts keywords of important entities from the resulting set. Entities can be the titles of high frequency articles and anchor texts. We rank these keywords with Lucene's [2] TF-IDF scoring function to rank an article. Where $Score_i$ is the score of an article title. Let i denote a article, j denote a term occurring within the title, and T denote the number of terms in the title. TF_j is the frequency of a term occurring within the title, TF_{ij} is the frequency that a term occurs in an article. We assume $token = |\{D_i - j\}|$, where D_i is the number of terms.

$$Score_i = \sum_j^T \left(\frac{TF_j \times IDF_j}{\sqrt{\sum_j (TF_j \times IDF_j)^2}} \right) \times \left(\frac{TF_{ij} \times IDF_j}{\sqrt{token}} \right) \quad (1)$$

where $IDF_j = \log\left(\frac{D}{D_j} + 1\right)$, D is all the articles in the index, D_j is articles that contains j .

It means the frequency of term occurs in the article is important. The higher of term frequency, the higher of the score of the title.

After filtering out the noise in the resulting set, such as redirected pages, our system maintains top 20 documents as the resulting set for further search. Our system thus finds the first level of relevant topics and then outputs them. These relevant topics can be treated as the backbone of a Wikibook.

2.2 Sub-topics Finding & Hierarchy Construction

Since the extracted topics from the first search often contain the original query term, our system removes the original query term string and uses the reduced topics to do the second search based on the result of the first search. For example, if we have retrieved the topic “Linux Operating System,” the shortened topic will be “Linux.” Our system extracts keywords from the search result as the sub-level topics of the output TOC. This is a recursive method; we can find the sub-sub-topics in the same way. For example, we can further search the sub-topic of “Linux.” Figure 1 shows the architecture of our system.

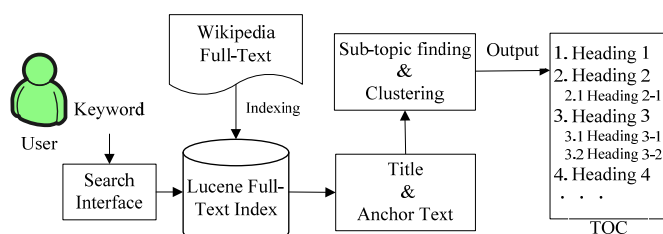


Figure 1: System Architecture

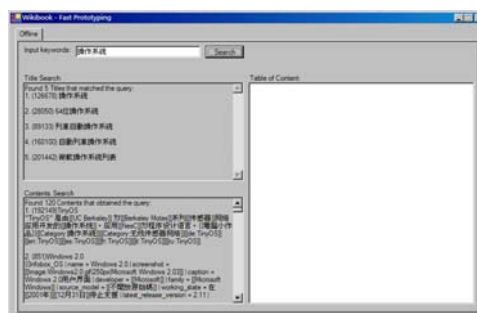


Figure 2: User Interface

3. Experiment

3.1 Dataset and UI

We download the dump data in English version of Wikipedia from the Wikimedia downloads website¹. We build a search tool based on the Lucene open source information retrieval API. A user interface is also built using Visual C# for the experiment. See Figure 2.

3.2 Result and Discussion

Currently, there is a Wikibook entitled “Operating System Design.” We take part of the TOC shown in Table 1. The first four level topics in this TOC are general concepts, which are independent of the title of the Wikibook. However, the sub-topics of the fifth topic are very similar to the automatically generated TOC. We observe that the sub-topic of the fifth topic, “Kernel Architecture,” states the concept of “Kernel Architecture” or gives some illustrations. Our system can help to automatically generate this part of the Wikibook. As the first example, we input a topic, “Operating system,” into our system. Then the system automatically generates a corresponding TOC, shown in Table 2. The sub-topics are ranked according to formula (1) discussed in Section 2.2. Next, we use the scheme described in

¹ <http://download.wikimedia.org/enwiki/>

Section 2.3 to perform a search again. We use another topic “Algorithm” to verify our system is validated, shown in Table 3.

Table 1: Current Wikibook TOC of “Operating System Design”	Table 2: Automatically generated TOC of “Operating System” via our system	Table 3: Automatically generated TOC of “Algorithm” via our system
1. Preface 2. Introduction 3. Case studies 4. History 5. Kernel Architecture 5.1 Monolithic Kernels 5.1.1 Solaris 5.1.2 Linux 5.1.3 Windows 9x 5.2 Microkernels 5.2.1 QNX 5.3 Exokernel 5.3.1 XOK 5.4 Hybrid Kernels 5.4.1 Windows NT/XP 5.4.2 Mac OSX 5.4.3 BeOS 6. Initialisation 6.1 Boot Loaders 6.2 Hardware Initialisation 7. Processes ...	1. Operating system 2. THE operating system 3. IBM AIX (operating system) 4. CPM operating system 5. Disk operating system 6. Kent Applicative Operating System 7. Linux operating system 7.1 Distro 7.2 Flask operating system 7.3 Sabayon 7.4 Red hatter 8. Operating system advocacy 9. Real-time operating system 9.1 Rubus (disambiguation) 9.2 RTMOS (Real-Time Multiprogramming Operating System) 10. VSE (operating system) 10.1 Exec 10.2 Protected procedure call 11. Computer network operating system 11.1 Faceless process ...	1. Algorithm 2. A* algorithm 2.1 List of algorithms 2.2 Timeline of algorithms 3. Asymmetric algorithm 3.1 List of algorithms 4. Euclidean Algorithm 4.1 List of algorithms 4.2 Timeline of algorithms 4.3 Euclidean 5. Merge algorithm 5.1 List of algorithms 5.2 Timeline of algorithms 6. Online algorithm 6.1 List of algorithms 6.2 Competitive analysis 6.3 Competitive analysis (online algorithm) 6.4 List of algorithm general topics 7. Sorting Algorithm 7.1 List of algorithm general topics 7.2 List of algorithms ...

We find that the sub-topics at the second level are adequate in this case. For example, the sub-topic of “Linux operating system” in Table 2 is instance of the Linux operating system. The scheme in Section 2.3 is effective in this case. We believe that the second search in the resulting set from the first search can really help find topics with a stronger relationship.

4. Conclusion and Future Work

We proposed an automatic process that can generate a Wikibook by searching the content of Wikipedia. Our method involves document searching, keyword extraction, and hierarchy construction technology. We built an experimental system and conducted primary experiments. The results showed that automatically generated TOC can help the online community edit a Wikibook more rapidly.

4.1 Identification of Hypernym/Hyponym Relation

To know the relations between relevant topics is very important in this application, especially the relation between the upper and lower concepts, known as the hypernym/hyponym relation. With the relation, a tree structure can be built and a hierarchy formed [8]. Supervised [1] or semi-supervised method [5] of hierarchical clustering algorithm is a promising method. After finding a set of relevant documents, a system may be clustered into a hierarchy according to the content. The titles of these articles can be treated as the TOC. A knowledge-based approach is also possible. We can try to identify the hypernym/hyponym relation between relevant titles by using WordNet [4] or SUMO.

4.2 Importance of an Article

Currently our system ranks relevant documents according to the scoring function of Lucene. We can combine the result with Google’s PageRank [9]. Each page of the Wikipedia entry

contains a link to a page that reports how many entries link to this entry. The more inward the link, the higher the importance of the entry is. With the analysis of the link relation, importance can be ranked [15]. This information helps to decide whether the TOC should contain this entry or not. The relatedness of words in Wikipedia might also help [10].

Acknowledgement

This research was partly supported by the National Science Council under NSC 96-2221-E-324-046.

References

- [1] Aggarwal, C. C., Gates, S. C., and Yu, P. S. (1999). On the merits of building categorization systems by supervised clustering. In *Proceedings of the Fifth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*. KDD '99., 352-356.
- [2] The Apache Software Foundation. (2008). Lucene Project. Retrieved December 26, 2007, <http://lucene.apache.org/>
- [3] Dicheva, D., Dichev, C. (2005). Authoring educational topic maps: can we make it easier? In *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*. ICALT'06., 216-218.
- [4] Farreres, J., Rodríguez, H., and Gibert, K. (2002). Semiautomatic creation of taxonomies. In *Coling-02 on Semanet: Building and Using Semantic Networks - Volume 11* International Conference On Computational Linguistics., 1-7.
- [5] Huang, R., Zhang, Z., and Lam, W. (2006). Refining hierarchical taxonomy structure via semi-supervised learning. In *Proceedings of the 29th Annual international ACM SIGIR Conference on Research and Development in information Retrieval*. SIGIR '06., 653-654.
- [6] Lally, A. M., & Dunford, C. E., (2007, May/June). Using Wikipedia to Extend Digital Collections, *D-Lib Magazine*, 13.
- [7] Muchnik, L., Itzhack, R., Solomon, S., and Louzoun, Y., (2007) Self-emergence of knowledge trees: Extraction of the Wikipedia hierarchies, *The American Physical Society*, Phys. Rev. E 76, 016106.
- [8] Nguyen, Dat P.T., Matsuo, Yutaka. and Ishizuka, Mitsuru., (2007) Subtree Mining for Relation Extraction from Wikipedia, In *Proceeding of NAACL/HLT 2007*, Companion Volume, 125-128.
- [9] Page, L., Brin, S., Motwani, R., and Winograd, T., (1999) The pagerank citation ranking: Bringing order to the web. *Technical Report*, Stanford University.
- [10] Ponzetto, S. P., & Strube, M., (2007) An API for Measuring the Relatedness of Words in Wikipedia, *Proceedings of the ACL 2007 Demo and Poster Sessions*, 49-52.
- [11] Roberson, S. and Dicheva, D. 2007. Semi-automatic ontology extraction to create *draft* topic maps. In *Proceedings of the 45th Annual Southeast Regional Conference*. ACM-SE 45., 100-105.
- [12] Sajjapanroj, S., Bonk, C., Lee, M. & Lin, G. (2006). The Challenges and Successes of Wikibookian Experts and Want-To-Bees. In T. Reeves & S. Yamashita (Eds.), *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2006*, 2329-2333.
- [13] Silberschatz, A., Galvin, P. B., Gagne, Greg., (2001). *Operating System Concepts, Sixth Edition*. John Wiley & Sons.
- [14] Völkel, M., Kröttsch, M., Vrandecic, D., Haller, H., and Studer, R. (2006). Semantic Wikipedia. In *Proceedings of the 15th international Conference on World Wide Web*. WWW '06., 585-594.
- [15] Wissner-Gross, A. D. (2006). Preparation of Topical Reading Lists from the Link Structure of Wikipedia. In *Proceedings of the Sixth IEEE international Conference on Advanced Learning Technologies*. ICALT'05., 825-829.
- [16] Yang, J., Han, J., Oh, I., and Kwak, M. (2007). Using Wikipedia technology for topic maps design. In *Proceedings of the 45th Annual Southeast Regional Conference*. ACM-SE 45, 106-110.