Corpus Exploitation from Wikipedia for Ontology Construction

Gaoying Cui, Qin Lu, Wenjie Li, Yirong Chen
Department of Computing, Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
E-mail: csgycui@comp.polyu.edu.hk, csluqin@comp.polyu.edu.hk, cswjli@comp.polyu.edu.hk, csyrchen@comp.polyu.edu.hk

Abstract

Ontology construction usually requires a domain-specific corpus for building corresponding concept hierarchy. The domain corpus must have a good coverage of domain knowledge. Wikipedia(Wiki), the world's largest online encyclopaedic knowledge source, is open-content, collaboratively edited, and free of charge. It covers millions of articles and still keeps on expanding continuously. These characteristics make Wiki a good candidate as domain corpus resource in ontology construction. However, the selected article collection must have considerable quality and quantity. In this paper, a novel approach is proposed to identify articles in Wiki as domain-specific corpus by using available classification information in Wiki pages. The main idea is to generate a domain hierarchy from the hyperlinked pages of Wiki. Only articles strongly linked to this hierarchy are selected as the domain corpus. The proposed approach makes use of linked category information in Wiki pages to produce the hierarchy as a directed graph for obtaining a set of pages in the same connected branch. Ranking and filtering are then done on these pages based on the classification tree generated by the traversal algorithm. The experiment and evaluation results show that Wiki is a good resource for acquiring a relative high quality domain-specific corpus for ontology construction.

1. Introduction

Ontology construction is a research area which gets off to a flying start in recent years. In the area of natural language processing, the basic requirement of ontology construction is an appropriate corpus. Building an ontology usually requires domain-specific corpus for acquiring concepts and building corresponding hierarchy of one domain. The domain corpus must have a good coverage of domain knowledge for generating a comprehensive ontology. Existing works have exploited different sources as corpus for ontology construction. Some early works used manually established corpora by domain experts (Collin F. Baker et al, 1998). But manual work is usually time consuming and labor intensive. The texts selected are often from books, magazines and news organizations automatically or semi-automatically (Latifur Khan & Feng Luo, 2002). But these corpora are not so easy to extend because knowledge contained in the corpora is fixed by time and regions and cannot be easily updated. Others try to exploit corpus from internet, such as using the results of Google search engine (P Cimiano et al, 2004). In fact, internet is a good source to collect corpus data. But, ontology construction requires domain-specific information. Classification information of articles over the internet may not be very clear and it is not so easy to obtain appropriate domain-specific corpus from internet search results.

Wikipedia(Wiki), the world's largest online source of encyclopedic knowledge, is a better candidate as domain corpus. Wiki has the following characteristics, open-content, collaboratively edited, and free of charge. It covers millions of articles and still expands continuously. Since established in 2001, Wiki has had tremendous growth both in size and public popularity. As of April 21st 2005 the English Wiki boasted more than 500,000 articles (Besiki Stvilia et al, 2005) and expanded from around 1 million articles (November, 2006) to more than 2 millions or more till now. A lot of researches have been done since Wiki was established including statistic works on readers and editors (A Lih, 2004), cultural biases analysis(F Bellomi & R Bonato, 2005), network structure analysis (F Bellomi & R Bonato, 2005) and some attempts for extending Wiki to the level of semantic web (M Völkel et al, 2006). The characteristics of Wiki make it a good candidate as domain corpus resource in ontology construction. However, the collection of articles selected from it must meet the challenges both in terms of quality and quantity.

In this paper, a novel approach is proposed to identify articles in Wiki as domain-specific corpus by making use of the classification information available in the article pages. The main idea is to generate a domain hierarchy from the hyperlinked pages of Wiki. Only articles strongly linked to this hierarchy are selected as the domain corpus. The proposed approach makes use of linked category information in Wiki pages to produce the hierarchy as a directed graph for obtaining a set of pages in the same connected branch. Ranking and filtering work of acquired nodes is then done on these pages based on a breadth first search algorithm. The experiment and evaluation results show that Wiki is a good resource to acquire a relative high quality domain-specific corpus for ontology construction.

The remaining of this paper is organized as follows. Section 2 reviews related works. Section 3 describes the proposed methodology for exploiting Wiki. Section 4 gives the experiment and evaluation details on different domains along with some analyses. Section 5 concludes

this paper and shows possible directions of future works.

2. Related Work

Many research works require the appropriate selection of corpus resources. Some researchers prefer to use manually constructed corpus by linguistic experts, such as the British National Corpus (BNC) (Collin F. Baker et al, 1998), a 100-million-word text corpus of written and spoken English from a wide range of sources. BNC was compiled as a general corpus (text collection) in the field of corpus linguistics.

A lot of research works on ontology construction make use of existing corpora acquired from books, magazines and news organizations automatically or semi-automatically, such as the text document corpus of Reuters (Latifur Khan & Feng Luo, 2002). Reuters released a corpus of Reuters News stories from the year 2000 to 2004 (named as Reuters21578 corpus) for research and development use such as natural language-processing, information-retrieval (IR) and information extraction (IE). The Reuters21578 corpus has previously been seen as a standard real-world benchmarking corpus for the IR/IE etc community which is marked up in XML.

Attempts are also made to collect corpus for ontology construction by making use of search engines over the Web and selecting corpus from search result (P Cimiano et al, 2004). They developed system based on the corpus collected from internet for trying to implement a self-annotating web.

All preceding mentioned methods had shortcomings in one aspect or another. Such as cost of time and human resource for manual corpora, time and region limitation for periodical and journal corpora, lack of appropriate classification information for corpora from internet.

Wikipedia as a Web resource is being used in many studies. Statistics and analysis had been done over Wiki on the ratio between number of edits and unique editors (A Lih, 2004). Statistics on structure and content of Wiki were also conducted (Jakob Voss, 2005). The work of analyzing Wiki's link structure and cultural bias had already been studied in (F Bellomi & R Bonato, 2005) which used two metrics, HITS and PAGERANK to gain insights on the macro-structure of the organization of the corpus and on cultural biases related to specific topics.

Some researchers have also tried to add semantic relation links and attributes to Wiki (M Völkel et al, 2006) and measure semantic relatedness using Wiki as resources (L Denoyer & P Gallinari, 2006). The former provided an extension to be integrated into Wiki which allowed the input of links between articles and the specification of typed data inside the articles in an easy-to-use manner. The latter presented methods on using Wiki for computing semantic relatedness and compared it to WordNet on various benchmarking datasets. Results showed that computing semantic relatedness from Wiki performed better than a baseline given by Google counts. There were also studies in which Wiki was used as the corpus in content-oriented XML retrieval area (L Denoyer & P Gallinari, 2006). The corpus used in this research from Wiki was composed of 8 main collections corresponding to 8 different languages: English, French, German, Dutch, Spanish, Chinese, Arabian and Japanese. In addition to these 8 collections, different additional collections were also provided for other IR/Machine Learning tasks like categorization and clustering, machine translation, multimedia IR, entity search, etc. The analysis on Wiki categories was also shown in (Sergey Chernov et al, 2006) to extract semantic relationships between them for building a semantic schema for Wiki to improve its search capabilities and provide contributors with meaningful suggestions for editing Wiki pages..

Since Wiki is an open-content and collaboratively edited online encyclopedia, it has expanded from around 1 million articles (November, 2006) to more than 2 millions or more till now. Wiki is an information-rich resource with hyperlinks to other entries and relevant classification information declared by contributors manually. Also Wiki contains large volume of articles in science and technology making it a good candidate for domain corpus extraction in fields like IT, biology, physics, etc.. The structure of Wiki can be considered as an interconnected network of articles. Each article is connected through hyperlinks in its main body to other Wiki entries. However, simply following these hyperlinks to find related articles in a domain is not appropriate because hyperlinks do not necessarily point to articles in the same domain. The category information of each article declared by contributors manually, on the other hand, provides more relevant information on domain specificity in classifying article types. However, each article can belong to different categories. Suppose we are trying to build an IT domain corpus. As an example, the article entry for "*Women, girls and information technology*" has categories "*Category:Women*", "*Category:Computing and society*", and "*Category:Information Technology*", etc.. Obviously this article is related to IT. Yet, it only touched the issue of IT, but not a qualified IT domain-specific article because most of the content are political rather than technical. Thus, in this work, we need to develop a method to identify the relevance of articles to a specific domain.

3. Methodology

A novel approach is proposed in this paper to identify articles in Wiki as domain-specific corpus by making use of the classification information in pages available. The main idea is to traverse and generate a connected domain hierarchy from the hyperlinked pages of Wiki, only articles strongly linked to this hierarchy are selected as the domain corpus. The linked category information in Wiki pages is used in the proposed approach as a directed graph

to produce a set of pages in the same connected branch, referred to as a *classification tree*. Ranking and filtering work is done on these pages during the traversal of the classification tree for final corpus acquisition. Evaluation has been done by using sampling method on the coverage and comparison to the comparable parts of the Library of American Congress Classification (LACC).

3.1 Basic Concepts

Among the 6 basic types of Wiki pages (ordinary article pages, category pages, image pages, template pages, talk pages and Wikipedia pages), only article pages and category pages are relevant in this work. Along the category information declared in each article page, Wiki can be considered as a directed graph where the articles are nodes and the category information as edges/links to other articles and category nodes. In this study, we use basic terms used in graph theory to define related objects in Wiki as follows:

First, all ordinary articles and category pages are considered as ***nodes*** in a graph and are named by their web page titles.

Definition 1: A ***directed edge***, is defined by a 2-tuple ***edge*(P_i, P_j)**, where P_i and P_j are two *nodes* and P_i contains the category information link to P_j. edge(P_i, P_j) is called the ***out-edge*** of P_i and also the ***in-edge*** of P_j.

Definition 2: A ***Wiki-graph G***, is a directed graph defined by a 2-tuple, $G = <V, E>$, where V is a not empty set containing Wiki ordinary articles and category pages as nodes, called the ***node set***; E is a set of directed edges, called the ***edge set***.

Definition 3: In a Wiki-graph, the ***in-degree*** of one node is defined as the number of in-edges of this node, and ***out-degree*** is defined as the number of out-edges of one node.

Generally speaking, a directed graph forms a network topology. According to the connectivity theory, if we start from a node P_r in the graph, all nodes connected to P_r can be reached following the directed edges. Generally speaking, given a Wiki-graph G, all the traversed nodes of P_r form another graph G' with a set of V' and E' such that $G' = <V', E'>$, which is a connected branch of G.

Definition 4: A ***Classification Tree*** $T = <V', E''>$ with a selected root node P_r is defined as a spanning tree of G', $G' = <G', E'>$, where G' is a connect branch of G and $E'' \subseteq E'$, $P_r \in V'$. For $\forall P_i \in V'$ and $\forall P_j \in V'$, edge(P_i, P_j) $\in E''$ if and only if edge(P_i, P_j) can be reached along the in-edges starting from P_r.

For example, if the Wiki-graph is traversed from a category node (say "*Category:Information Technology*" for IT domain and "*Category:Biology*" for biology domain) P_r, the article nodes that can be reached through category nodes are in fact the terminal nodes and all the traversed nodes form a tree-like structure, which can be considered as a *classification tree* starting from a properly selected node. All the edges connected to a node P_i are either out-edges from P_i to the categories pages which P_i belongs to or in-edges pointed from category pages to P_i. $N_{out}(P_i)$ denotes the total number of out-edges from P_i and $N_{in}(P_i)$ denotes the total number of in-edges to P_i.

A fragment of a *Wiki-graph* is shown in Figure 1 using P as the current node. As node P belongs to 3 categories A, B, C and pages E, F belong to category P, which number of out-edges $N_{out}(P)$ is three ($N_{out}(P) = 3$) and in-edges $N_{in}(P)$ is two ($N_{in}(P)=2$). The right part of Figure 1 shows the content of a terminal node F. Different starting nodes

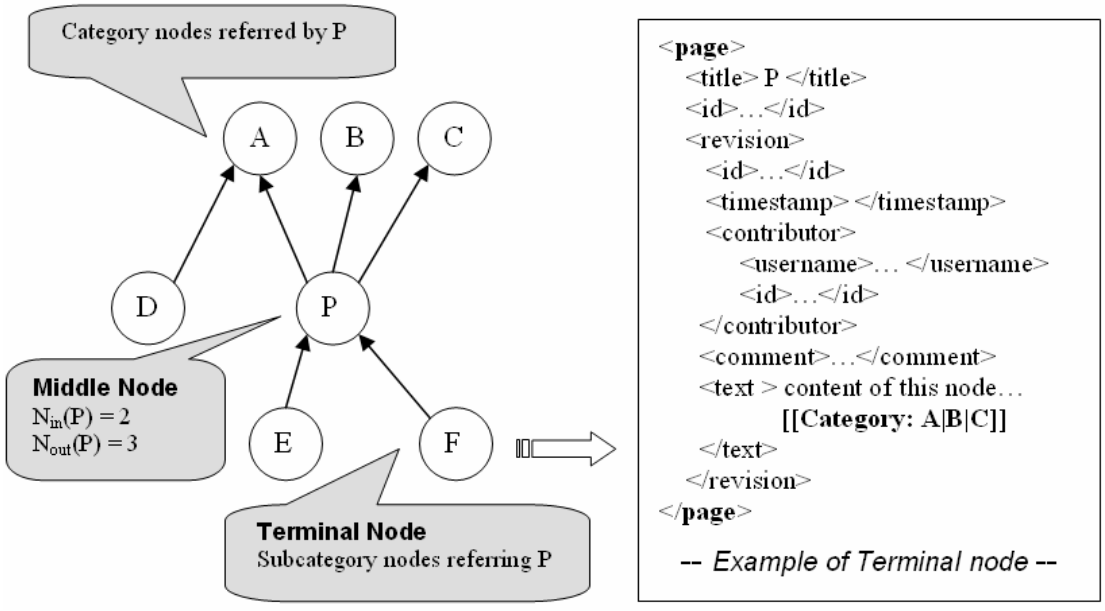

Figure 1 A Fragment of Directed Category Graph from Wikipedia

will lead to different classification tree structures. If P is a terminal node, its $N_{in}(P)$ should be zero.

3.2 Classification Tree Traversal

In this work, the process of traversing the classification tree is in fact growing a spanning tree of a connected branch in *Wiki-Graph* from a specified root node. Theoretically, both depth-first search and breadth-first search can be used for traversing a spanning tree. Breadth-first search is used in this algorithm because it is traverses the tree one level at a time starting from the root which makes it easy to show the relations between visited nodes and the root node in a naturally hierarchical way. First, two hash tables are used to store all the positive and negative pairs of pages with their categories. Then, the below pseudo code in Figure 2 shows the algorithm to generate the classification tree of *Wiki-Graph* using breadth-first search (**CT-BFS**).

```
Algorithm: CT-BFS
Input: Wiki-graph G and a root node Pr
Output: classification tree T with ranked nodes
    For each node Pi in G {
        visited(Pi) = False;
    } EndFor
    visited(Pr) = True;
    T.V = {Pr};
    T.E = { };
    W(Pr) = 1;
    Push Pr into queue Q;
    While (Q.empty( )==false) {
        Pc = Q.pop( );
        For each in-edge Ei of Pc {
          Pn is the other end node of Ei;
          If (visited(Pn)==false and Pn not in Q){
                T.E = T.E ∪ {Ei};
                Push Pn into Q;
            }Endif;
          } EndFor
        W(Pc) = Scoring();
        Visited(Pc) = True;
        T.V = T.V ∪ {Pc};
    } EndWhile
    Return tree T with ranked nodes.
```

Figure 2 Pseudo codes of CT-BFS

As given in the algorithm in Figure 2, for a selected root node P_r (how this node is selected will be discussed later), all other node pages in the same connected branch of P_r must be pointed to P_r either directly or transitively through the in-edges of P_r. Therefore, the BFS-WG algorithm starts from P_r to traverse the spanning tree for all the nodes along the in-edges, one level at a time, to all reachable terminal nodes. No circle can form because no node will be visited twice. Using breadth-first search, the shortest route from the root node to each terminal node will always be selected if there are multiple routes between them. Each node is given a score after its visit according to the different scoring schemes to be discussed in Section 3.3. The scoring is based on relevance calculation to the domain.

3.3 Ranking Nodes in the Classification Tree

During the classification tree traversal, each node is given a score on the relevance of the node to the specific domain. Once the traversal is completed, the terminal nodes, which are the article pages, are ranked according to the domain relevance scores. Pages over a certain threshold is considered domain relevant. The threshold value of ranking is an experiment dependent algorithm parameter. The score can consider either in-edges or out-edges, even though Wiki pages can belong to multiple categories, it is easier to see that the more out-edge nodes a node P_c have that are pointing to the classification tree, the more likely the node is domain specific. For a given P_c, suppose it has a total of $N_{out}(P_c)$ number of out-edges. Among them, m out-pages point to the classification tree. P_i is the i^{th} out-page of P_c in the classification tree where $i = 1, \ldots, m$ with P_i's score W_i obtained from the previous iteration of the BFS-WG algorithm. To initiate, $W_r = 1$ for node P_r, and $W_i = 0$ if P_i is not on the traversal path to this level. Three scoring schemes are proposed with consideration of different proportions of $N_{out}(P_c)$ and $N_{in}(P_i)$, where $N_{out}(P_c)$ means how many nodes are P_c's category nodes and $N_{in}(P_i)$ means how many nodes take P_i as one of their category nodes. The numbers of in-edges and out-edges of each node in Wiki are independent of the classification tree and they are acquired and stored in two hash tables and are used during the classification tree traversal. Below shows the formulas for calculating the score W_c of P_c during traversal.

$$\mathbf{S_1}: \quad W_c = \frac{1}{N_{out}(P_c)+1} \times \sum_{i=1}^{m} W_i \tag{1}$$

$$\mathbf{S_2}: \quad W_c = \sum_{i=1}^{m} \left(W_i \times \frac{1}{N_{in}(P_i)+1} \right) \tag{2}$$

$$\mathbf{S_3}: \quad W_c = \sum_{i=1}^{m} \left(W_i \times \frac{1}{(N_{in}(P_i)+1) \times (N_{out}(P_c)+1)} \right) \tag{3}$$

In the scoring scheme $\boldsymbol{S_1}$, the W_c of P_c takes the sum of the scores of all its out-edges (all the P_is) that are pointing to the classification tree against the total number of out-edges that P_c has. In $\boldsymbol{S_2}$, the score of P_c is considering the summation of its out-edges in the classification tree against the total number of their in-edges. $\boldsymbol{S_3}$ scores P_c according to the summation of the out-edge nodes in the classification tree divided by both the total number of its out-edges and the total numbers of in-edges of those upper level nodes in the classification tree. In summary, $\boldsymbol{S}_1$ and $\boldsymbol{S_2}$ consider the influence of out-edges and in-edges separately, whereas $\boldsymbol{S_3}$ combines both factors.

4. Experiment and Evaluation

The experiments are done on the English Wiki with the

cut off date of November 30th, 2006 containing about 1.1 million article and category nodes. In order to valid that the method can work on different domains, two domains are selected in the evaluation of the proposed algorithm and ranking schemes. They are the domains on IT and biology. In the IT domain, the root node is "*Category:Information Technology*". Using the CT-BFS traversal algorithm, the obtained classification tree can reach 549,486 nodes of the 1.1 million nodes. As for the biology domain, the root node is "*Category:biology*" which can reach 549,433 of nodes when the classification tree is traversed. Considering that Wiki has articles in many other domains, it is obvious that there is a heavy overlapping of the two sets of pages. Thus, a selection scheme must be applied to choose the most relevant pages of a single domain.

4.1 Scoring Scheme Selection

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
S_1	7	0	0	0	0	0	0	0	0	0	0	0	7	0	0	7	0	9	8	0	10	1	3	10	0
S_2	10	10	10	10	9	7	9	10	10	10	10	10	1	10	10	0	0	9	8	7	0	10	0	0	0
S_3	10	10	10	10	9	10	10	10	10	10	8	10	1	10	10	10	7	10	10	10	8	10	0	0	0

Table 1 Evaluation Result of Different Schemes in the IT Domain

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
S_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	2	10
S_2	10	10	10	8	10	6	7	6	7	4	10	10	0	8	10	9	0	5	9	5	10	0	0	0	0
S_3	10	10	10	10	9	9	9	6	10	10	9	7	10	10	3	6	9	9	8	9	2	0	0	0	0

Table 2 Evaluation Result of Different Schemes in the Biology Domain

The scoring schemes are applied in separate experiments from selected domain-specific root category nodes for the two domains. The results are then ranked, respectively. For evaluation, the ranked nodes are sampled at the interval of 1,000 for the first 20,000 nodes, then for the interval of 20,000 from 20,000 to 100,000. At each sampling point, 10 consecutive nodes are taken manually by two people knowledgeable in both IT and biology. Thus, for each domain, there are a total of 250 samples for evaluation. Precision is used as the measure of performance. Table 1 shows the evaluation results on the IT domain and Table 2 for the biology domain. The first 20 columns show the interval of 1,000 nodes of the top 20,000 nodes and the last 5 columns are results from every 20,000 nodes from the 20,001th to 100,000th nodes.

As shown in Table 1 for the result on the IT domain, the descending tendency is quite apparent in S_3 and S_2 according to the ranks, yet, it is not so apparent in S_1. This means that the consideration of out-edges (in S_1) is not sufficient. This tendency is even more apparently in Table 2 for the biology domain because among the sampling results from top 20,000 nodes of S_1 in Table 2, there are no domain relevant pages until after 60,000. In fact the results from both tables show that, S_2, which consider the influence of in-edges to upper level nodes, performs better than considering of out-edges from current node only. The results of S_3 have fewer fluctuations than that of S_2 which means in most probability that S_3 is better than S_2 because both the total number of in-edges and total number of out-edges are factored in. On the whole, the sampling results in Table 2 are in a very similar situation to that in Table 1, but there are a few differences between Table 1 and Table 2. Besides the difference in S_1, the number of biology relevant items is less than that of the IT domain. After the top 60,000 nodes in IT domain, not many nodes can be really qualified as domain-specific articles. While in the biology domain, the boundary has been advanced to 40,000.

Generally speaking, a good scheme should show the results in a descending order and has a good overall precision of domain relevant pages. The overall precision of both IT and biology domain using different schemes on the top 20,000 nodes are shown in Table 3. Results show that S_3 is the best scheme for getting the pages which are most relevant to the selected domain. On the basis of this fact, experiments show that the first 20,000 articles can be taken to form domain corpus with good confidence using S_3 for both corpora. For the IT domain, the generated corpus size is 98M. While for biology domain, the size of the generated corpus is 101M, which are reasonable as domain corpus without any need for manual selection.

Schemes	Average IT Coverage	Average Biology Coverage
S_1	19.0%	0.0%
S_2	76.5%	72.0%
S_3	92.5%	86.5%

Table 3 Overall Precisions of Different Schemes

4.2 Root Node Identification

The selection of the root node is vital to the quality of the corpus acquired using this algorithm. Taking the IT domain as an example, the node "*Category:Information Technology*" is the starting node in this work. However, two other nodes "*Category:Communication*" and "*Category:Electronics*" were also taken as the root node to be applied to the algorithms. It is interesting to note that almost the same number of nodes (549,486, 549,485 and 549,483) is reached by all three different starting nodes although the classification trees are different, and the ranking results are different. Taking S_3 as the scheme, the node "*Information technology management*" ranked 33 if the starting node is "*Category:Information Technology*", the same node will be ranked 425,335 if the starting node is "*Category:Electronics*" which in fact is not appropriate. What this experiment told us is that the choice of the root node does made a difference, as out of the 549,486 reachable nodes, only about 20,000 top ranked pages are used. It is understood that the classification information provided in Wiki are supplied by contributors manual with strict rules to follow any reference classification. In fact, because the hyperlink structure for these classification links in Wiki is a network, it is likely that almost all the domain-specific nodes can be reached if the traversal starts from just any node. This does give rise to the need to further validate the appropriateness of the selected root node.

For evaluation, the classification provided by the Library of American Congress Classification (LACC), is used as the external reference and is assumed to be a correct classification. It should be noted that LACC as a classification for books, has a rather flat structure. For the 32 IT related categories in LACC, for example, the hierarchical structure is not complete and there are partial trees involved as given in Appendix A for reference. The relations refer to the links defined in LACC and there is a total of 26 such relation links for the IT category. The validation compares the produced classification tree from this work using S_3 in terms of (1) the coverage of the with respect to all the domain tree in LACC and (2) violation of the hierarchy with respect to that of LACC. The comparison is done by manual check so that abbreviations and plurals are considered.

Root Node	IT		Biology	
	Terms	Relations	Terms	Relations
LACC	32	28	31	25
Wiki	26	23	21	17
Domain Classification Tree	21	20	20	15

Table 4 Comparisons of Classification Trees with Root Nodes from Respective Domains

Table 4 shows a summary of the evaluation result. Out of the 32 IT relevant categories in LACC, 26 of them appear in Wiki as either a category nodes or article nodes. The S_3 algorithm identifies 21 of these 26 categories in its classification tree. The categories in LACC but do not appear in Wiki (a total of 6) are because the LACC category names are more general high level names which are not used by the contributors directly. However, the corresponding lower level categories or names are in Wiki. For example, for the LACC category name "Internet domain names", category nodes "World wide web", "Internet protocols" and other more detailed terms are used instead in Wiki. Out the 28 pairs of categories in LACC that hold hypernym relations, 23 of them appear in Wiki. Using S_3, 20 such relations are maintained in the same order in the acquired classification for the IT domain. The other 3 relations do not exist in the classification tree. For example, in LACC, "*Usenet*" is under the classification of "*Networks*", on the other hand, the page titled as "*Usenet*" in Wiki can link to "*Networks*" by following the category nodes "*Wide area networks*", "*Networks by scales*", "*Computer networking*" and "*Networks*". That means the relation between "*Usenet*" and "*Networks*" from LACC and Wiki are consistent.

As to biology domain, there are 31 relevant nodes in LACC. Among them, 21 occur in Wiki as article nodes and 20 are contained in the classification tree. Although some items which cannot be found in Wiki, such as "*Cytology*" and "*Animal biochemistry*", their synonyms can be found in the classification tree from Wiki, called "*cell biology*" and "*biochemistry*" and are contained in the classification tree. Out of the 25 hypernym relations in LACC, 17 uch relations are in Wiki, and 15 are maintained in the acquired classification tree. The other 2 can not be found in the classification tree. For example, the item "*Cyanobacteria*" is below the classification branch of "*Microbiology*" and "*Microbiology*" is a major subfield of "*Biology*" in LACC. In Wiki classification tree, the node "*Cyanobacteria*" can also be linked to "*Biology*" according to the sequence of category nodes "*Bacteria*", "*Prokaryotes*", "*Microorganisms*" and "*Microbiology*". Obviously, the relation between "*Cyanobacteria*" and "*Microbiology*" acquired from the biology domain classification tree of Wiki also conforms to that in LACC. Comparing the two experimented domains, both the node and relation coverage of the biology domain are lower than that of the IT domain. That may be because biology is a more traditional and stable domain. While IT, as a new area of science and technology, can involve more interdisciplinary and applied areas.

Further examination was also done to compare the classification tree with "*Category:Electronics*" as root node to the IT domain and the electronics domain in LACC as shown in Table 5. When compared with the electronics classification in LACC (See Appendix B for details), "*Category:Electronics*" is a good choice as root node because most of the nodes as well as the relations

can be found in LACC. However, when this classification tree is compared to the IT categories in LACC, it can be seen that only 14 nodes out of the 26 in Wiki are found. Furthermore, out of the 23 category relations, only 2 of them are maintained in the same way. In other words, this classification tree is much more screwed when compared to the IT domain's classification.

Root Node	Electronics			
	For Electronics		For IT	
	Terms	Relations	Terms	Telations
LACC	34	42	32	28
Wiki	30	36	26	23
Domain Classification Tree	23	30	14	2

Table 5 Comparisons of Classification Tree Structures with LACC with Root Node: Electronics

The comparisons with the LACC classification indicate that the choice of root node must be representative of the domain. Otherwise, the generated classification tree would not be representative of the domain, putting the quality of the acquired articles at risk. However, if one choose a general term popularly used to represent the domain, the classification hierarchy of LACC is generally maintained by the obtained classification tree and thus the quality of the acquired corpus is domain-specific.

5. Conclusion and Future Work

In this paper, a novel approach is proposed to select appropriate domain-specific data from Wiki for ontology construction. A classification tree is acquired from traversing the category nodes in Wiki using the proposed BFS-WG algorithm. Three different schemes are evaluated with consideration of in-edges, out-edges, and their combinations. Two domains, the IT domain and the biology domain, are selected for evaluation. Results show that the scheme taking into consideration of both types of edges gives the best performance and the corpus using this scheme are of good quality and can be readily used without the need for manual selection. This confirms that Wiki is a good web resource as a domain-specific corpus if proper selection and scoring algorithms are applied. Also, the quality of the algorithm is dependent on the choice of the root node in the traversal. A general rule of thumb is to use the most representative term used to name the domain and the result should be quite reasonable.

The current work makes use of the category information in category pages and articles pages only. In the future, it would be interesting to explore the selection of domain-specific pages by making use of page content and other hyperlinks provided in the article pages.

6. Acknowledgements

This project is partially supported by CERG grants (PolyU 5190/04E and PolyU 5225/05E) and B-Q941 (Acquisition of New Domain Specific Concepts and Ontology Update).

7. References

Besiki Stvilia, Michael B. Twidale, Les Gasser and Linda C. Smith. (2005). Information Quality Discussion in Wikipedia. *In Proceedings of the 2005 International Conference on Knowledge Management* (ICKM05), 2005.

Collin F. Baker, Charles J. Fillmore and John B Lowe. (1998). The Berkeley FrameNet Project. *In proceedings of COLING/ACL 98*, 1998, pp.86-90.

Latifur Khan and Feng Luo. (2002). Ontology Construction for Information Selection. *In proceedings of International Conference on Tools with Artificial Intelligence*, 2002.

P Cimiano, S Handschuh and S Staab. (2004). Towards the SelfAnnotating Web. *In proceedings of the 13th International Conference on World Wide Web*, 2004.

A Lih. (2004). Wikipedia as Participatory Journalism. Reliable Sources? Metrics for evaluating collaborative media as a news resource. Symposium. *In proceedings of the International Symposium on Online Journalism*, 2004.

Jakob Voss. (2005). Measuring Wikipedia. *In proceedings of the 10th International Conference of the ISSI*, 2005.

F Bellomi and R Bonato. (2005). Network Analysis for Wikipedia. *In proceedings of Wikimania*, 2005.

M Völkel, M Krötzsch, D Vrandecic, H Haller and R Studer. (2006). Semantic Wikipedia. *In proceedings of the 15th International Conference on World Wide Web*, 2006.

Michael Strube and Simone Paolo Ponzetto (2006). WikiRelate! Computing Semantic Relatedness Using Wikipedia. *In proceedings of AAAI*, 2006.

L Denoyer and P Gallinari (2006). The Wikipedia XML Corpus. *ACM SIGIR Forum*, Vol. 40, No.1, June 2006.

L Denoyer and P Gallinari. (2006). The Wikipedia XML Corpus. *ACM SIGIR Forum*, Vol. 40, No.1, June 2006.

Sergey Chernov, Tereza Iofciu, Wolfgan Nejdl, and Xuan Zhou. (2006). Extracting Semantic Relationships between Wikipedia Categories. *In proceedings of SemiWiki 2006 at the ESWC*, 2006

Library of American Congress Classification (LACC) http://www.loc.gov/catdir/cpso/lcco/

Appendix A IT Relevant Parts of LACC

QA 76.	Computer Science
QA 76.625.	Internet Programming
QA 76.73.	Programming Languages
QA 76.73.J38.	Java
QA 76.73.J39.	JavaScript
QA 76.76.	Computer Software (special topics)
QA 76.76.H94.	Hypertext. HTML
QA 76.76.H94.	RSS
QA 76.76.O63.	Operating Systems. Unix
QA 76.9.	Computer Science(other topics)
QA 76.9.D3.	Databases
QA 76.9.W43.	Web Databases

TK.	Electrical Engineering
TK 5105.	Telecommunications:Data Transmission Systems
TK 5105.565.	CGI
TK 5105.73.	Electronic Mail Systems
TK 5105.875.	Special Networks and Systems
TK 5105.875.I57.	Internet
TK 5105.875.U83.	Usenet
TK 5105.882.	Browsers
TK 5105.8835.	Internet domain names
TK 5105.884.	Search engines
TK 5105.886.	Internet Relay Chat
TK 5105.888.	World Wide Web
TK 5105.8882.	Wikis
TK 5105.8884.	Weblogs

ZA.	Information Resources (general)
ZA 4080.	Digital Libraries
ZA 4201.	Internet
ZA 4226.	World Wide Web
ZA 4480.	Electronic Discussion Groups
ZA 4550.	Video Recordings

Appendix B Electronics Relevant Parts of LACC

TK1-9971 Electrical engineering. Electronics.

TK301-399	Electric meters
TK452-454.4	Electric circuits. Electric networks
TK1001-1841	Powerplants. Central stations
TK2000-2891	Dynamoelectric machinery and auxiliaries
	Generators,motors, Transformers
TK4125-4399	Electric lighting
TK4601-4661	Electric heating
TK5101-6720	Telecommunication
	Telegraphy, telephone, radio, radar, television
TK 5105.	Data Transmission Systems
TK 5105.565.	CGI
TK 5105.73.	Electronic Mail Systems
TK 5105.875.	Special Networks and Systems
TK 5105.875.I57.	Internet
TK 5105.875.U83.	Usenet
TK 5105.882.	Browsers
TK 5105.8835.	Internet domain names
TK 5105.884.	Search engines
TK 5105.886.	Internet Relay Chat
TK 5105.888.	World Wide Web
TK 5105.8882.	Wikis
TK 5105.8884.	Weblogs
TK7800-8360	Electronics
TK7885-7895	Computer engineering. Computer hardware
TK8300-8360	Photoelectronic devices (General)