A novel approach for building Domain-specific Lexical Repository with Chinese Wikipedia

Zhijian RUAN[1,a], Xiu LI[2,b]

[1]Department of Computer Science and Technology of Tsinghua University

Beijing, 100084, China

[2]Department of Computer Science and Technology of Tsinghua University

Beijing, 100084, China

[a]email: shadowkimi520@gmail.com, [b]email: lx@mail.tsinghua.edu.cn

Keywords: Domain-specific lexical repository, Domain Corpus, Domain Relatedness, Modified Explicit Semantic Analysis, Chinese-Wikipedia.

Abstract. Domain ontology is a collection of domain-specific concepts and their interrelationships, which provide an abstract view of the application domain and is used in many areas such as semantic mining(SM) and natural language processing(NLP). But the direct construction of Domain ontology manually is labor intensive and time consuming, while auto-generated Domain-specific Lexical Repository can be used to build domain ontology as an indispensable component. In this paper, we propose a two-stage method to build domain-specific lexical repository making use of the dump service of Chinese Wikipedia. The main idea is that only concepts strongly semantic-related to the multi roots we choose are incorporate into the repository. First we use the dump service for all pages(zhwiki-all-pages.xml) of Chinese Wikipedia to generate a graph of all Wikipedia concepts, we call it pre-stage. Then we enter stage one by selecting three top-level nodes as roots, traversing the graph generated in the pre-stage using BFS-like algorithm to form spanning trees and computing rough domain relatedness of these nodes at the same time. Finally, in stage two we use the novel Modified Explicit Semantic Analysis method combined with the results we got in stage one to compute the ultimate domain relatedness. The experimental results shows that our method could get a high-quality domain-specific lexical repository.

Introduction

Domain-specific ontology construction is a research area which was slowly emerging in recent years. In the area of natural language processing, the basic requirement for ontology construction are an appropriate lexical repository and a corpus related to the lexical repository. Domain-specific lexical repository is a vocabulary repository consists of almost all the concepts in a domain, and the concepts in the repository are ranked according to their domain relatedness. Domain-specific corpus is the language material corresponding to the lexical repository, which includes the concept's application context and gives a definition of the concept. Building an ontology usually requires domain-specific lexical repository for acquiring concepts and corresponding corpus for building concept hierarchy of the domain. The lexical repository must have a good coverage of domain knowledge for generating a comprehensive ontology. The existing method of establishing a domain-specific lexical repository can be divided into two categories: automatic and manual. The manual repository is revised by linguistic experts who can ensure the accuracy of the domain-relatedness and the category classification to be rational(such as WordNet). This leading to the prevalence of building the domain-specific lexical repository automatically or semi-automatically. The extraction of domain information from books, magazines and news organizations semi-automatically[1] is a rational way. There is other way by exploiting repository from internet, such as using the results of Google search engine[2].

Since established in 2001, Wikipedia has evolved into a comprehensive resource with very good coverage on diverse topics, domain-specific concepts, important entities, etc. As the world's largest

online source of encyclopedic knowledge, Wikipedia has the characteristics of open-content, collaboratively edited, and free of charge, it has a comprehensive coverage of domain-specific knowledge and it's a realtime online knowledge system can be handled by computers automatically. Many researchers use the different part of Wikipedia to measure semantic similarity and semantic relatedness, such as the link structure between all the Wikipedia pages[3][4], the category structure of Wikipedia concepts[5], the content of all the Wikipedia articles[6].

In this paper, a novel two-stage approach is proposed to construct domain-specific lexical repository making use of the Chinese Wikipedia. The main idea is that only concepts strongly semantic-related to the multi roots we chosen are incorporate into our repository. First we use the dump service for all pages(zhwiki-all-pages.xml) of Chinese Wikipedia to generate a conceptual graph of all Wikipedia concepts, we call it pre-stage. Then we enter stage one by selecting three top-level nodes as root nodes, traversing the graph generated in the pre-stage using BFS-like algorithm to form spanning trees and computing rough domain relatedness of these nodes at the same time. Finally, in stage two we use the novel Modified Explicit Semantic Analysis method combined with the results we got in stage one to compute the ultimate domain relatedness. The experimental results show that our method could get a high-quality domain-specific lexical repository.

Related Work

The existing method of establishing a domain-specific lexical repository can be divided into two categories: automatic methods and manual methods. The manual construction is that linguistic experts collect concepts in specific domain and build the relationships between them. It's time consuming and labor intensive, also the main point is that domain-specific experts are needed to building the domain-specific lexical repository, so the built repository will have large deficiencies in the coverage and real-time capability of domain-specific concepts.

Then, a lot of researchers pay emphasis on automatic or semi-automatic construction of domain-specific lexical repository. Latifur Khan & Feng Luo[1] proposed a method to construct lexical repository by processing books, magazines and news organizations semi-automatically. But, this method can't be used in most domains because of the time and region limitation and the poor coverage on domain-specific concepts of the gathered material. P Cimiano et al.[2] presented an automatic method by using the results of search engines, they also developed a system based on the collected lexical repository for trying to implement a self-annotating web. This can resolve the real-time problem of domain-specific concepts, but the results of the search engine lack of appropriate classification information, also the authority and quality of web pages can't be ensured, which leading to the incorporation of many domain-independent concepts and affecting the further use of the constructed domain-specific lexical repository.

Another type of methods use the online encyclopedia Wikipedia. As an information-rich resource with hyperlinks to other entries and relevant classification information declared by contributors manually, also Wikipedia contains large volume of articles in science and technology making it a good candidate for domain-specific concepts extraction in fields like IT, biology, physics, etc.. Each article is connected through hyperlinks in its main body to other Wikipedia entries. However, each article can belong to different categories, the collection of concepts selected from it must meet the challenges both in terms of quality and quantity. Thus, to use Wikipedia resources, people need to develop a method to identify the relevance of articles to the specific domain.

Gaoying Cui et al.[7] came up with the idea to construct domain-specific lexical concepts using the Wikipedia resources firstly. They identify concepts in Wikipedia as domain-specific by using available classification information in Wikipedia pages. The main idea is to generate a domain hierarchy from the hyperlinked pages of Wikipedia, and only concepts strongly linked to this hierarchy are incorporated into the domain-specific lexical repository. They treat the category concepts in Wikipedia as nodes, the category relationships between nodes as edges, so the Wikipedia concepts and the category hierarchy relationships between concepts can form a directed graph. To generating the domain-specific lexical repository, they choose a root concept in the

specific domain and start traverse the directed Wikipedia Graph from the chosen root concept by BFS algorithm to forming the spanning tree. During the traversing, the domain relatedness of the traversed nodes are computed with the relatedness value of the root node initialized to 1.0. This method has good performance in the authority of content, coverage and real-time capability of the domain-specific concepts, but it has limitations. Due to the complexity of the manually classification of Wikipedia, traversing from the single root node can't be sufficient to incorporate the comprehensive concepts of the specific domain, because a domain in our real life corresponds to different category branches in Wikipedia. We can choose high-level root node to solve this problem, but this can involve massive domain-independent concepts. Another deficiency is that during the computation of a node's domain relatedness all the parent nodes' domain relatedness value are used, but if the parent node is not in the spanning tree its value is assigned to 0, which is not rational. The last deficiency is that the method only uses category information in Wikipedia to compute domain relatedness, but the text content of Wikipedia pages is more important for an accurate computation.

Our previous work[8] solved the first two problems by traversing from well-chosen multi roots, and using the topological sort algorithm to compute the domain relatedness. We use a BFS-like algorithm to traverse along the multi-roots and generate the directed cyclic graph of Wikipedia. Then an algorithm mainly based on Kosaraju Algorithm is proposed to remove the cycles in the Wiki-graph. Finally, to compute all the parent nodes' domain relatedness before computing current node's relatedness, topological sort algorithm is used to traverse the cycle-removed Wiki-graph rather than the spanning tree growing algorithm used in [7], and ranking and filtering is done during the process. This method also has problems: we should have a memory representation of the traversed nodes, that is to generate the Wikipedia-graph in the memory, which is huge and should store the child-relationship and parent-relationship both to facilitate the domain relatedness computation; before the ranking and filtering procedure, the cycles in the Wikipedia-graph must be removed, which is time-intensive. Besides, which edge to be removed is hard to determine and can affect the accuracy of the ranking procedure. To solve all the problems and take the text content into the domain relatedness computation, we proposed a novel two-stage method described in Section 3.

Building Domain-specific Lexical Repository with the Two-Stage Method

The main point of our novel method is that only concepts strongly semantic-related to the multi roots we chosen are incorporate into the domain-specific repository.

1) **Pre-Stage: Building the conceptual Wikipedia Graph**

Not the same as other Wikipedia-based Domain-specific Lexical Repository construction methods, we use the dump service for all pages of Chinese Wikipedia, instead of crawling all the Wikipedia pages to build the hash tables to store all the positive and negative pairs of pages with their categories, which is time-consuming and traffic-intensive. By using the dump service of Chinese Wikipedia, we download the zhwiki-all-pages.xml documents and introduce a pre-stage step to build the conceptual Wikipedia graph and compute the in-degree and out-degree of all Wikipedia concepts. While the entire nodes in Wikipedia are tremendous and the Wikipedia Graph we built is very large, so we store the conceptual graph in separate files by each contains the concept's subcategory concepts and parent category concepts. But the Chinese Wikipedia is still not perfect, there are few Chinese Traditional concepts in the category page, so we need to convert traditional Chinese to simplified Chinese. Here, we use a python program for this task.

Our Pre-Stage procedures are as follows: (1). Initialize an empty set to store all the processed Wikipedia concepts. (2). Use xml.dom model in Python to parse all the Wikipedia pages, if the page's concept is in the set, which means we have previously processed the concept's ordinary page or category page, then add additional content into the corresponding .txt file, else new a txt file named by the concept. (3). After parsing all the article in the downloaded xml document, the results of our Pre-Stage is the conceptual Wikipedia Graph, which consists of many txt files with the Wikipedia concepts as titles, each containing the concepts' subconcepts and parent concepts(Also the in-degree and out-degree of the node).

2) Stage-One: Rough Domain-Relatedness Computing with Multi-roots

A specific domain in our real life can't be strictly mapped to single corresponding category node in Wikipedia. To include the domain-specific concepts as more as possible, we choose three specified root nodes which are highly related to the domain. Then, we use a BFS-like algorithm(Figure 1) to traverse the conceptual Wikipedia Graph and grow the spanning tree of each root node.

```
Algorithm: MR-BFS-like
Input: the conceptual Wikipedia Grpah generated
         in pre-stage(many .txt files)
Output: the ranked traversed nodes
Queue Q;
Map M;
Push root1 into Q;
Push root2 into Q;
Push root3 into Q;
While( !Q.empty() ) {
    current_node = Q.pop();
    read current_node's .txt file;
    get in-degree, out-degree from the .txt file;
    for each sub_category in current_node.txt {
        if( sub_category not in M
        and sub_category not in Q ) {
            Push sub_category into Q;}}
    if(current_node in root1, root2, root3) {
        tuple = make_tuple(in-degree, out-degree, 1);
    } else {
        score(current_node) = Scoring();
    tuple = make_tuple(in-degree, out-degree,
            score(current_node) ); }
    insert {current_node:tuple} into Map M;
}
```

Fig.1. The pseudo code of MR-BFS-like

During the generation of the spanning trees, each node is given a rough value on the relatedness of the node to the specific domain by all the parents' relatedness value. After the traversal, the gathered concepts are ranked according to the domain relatedness scores. It's easy to see that the more in-edge a node Pc has that are starting from the spanning trees, the more likely the node is domain specific. The scoring formula is as below, which is testified in [7].

$$W_c = \frac{1}{N_{in}(P_c)+1}\sum_{1}^{count\_parent(P_c)} W_i \times \frac{1}{N_{out}(P_i)+1} \qquad (1)$$

In formula (1), all the parent categories of current node Pc are take into consideration, the final score combines the both influence of the total number of the current node's in-edges and the total number of the upper level node's out-deges. Pi is the parent category node, and its score Wi is obtained from the previous traverse. In this step, all the root nodes' values are initialized to score 1.0. If Wi is not in the spanning trees, or deeper than the current node(that is not inserted into the Map M to this level) then its socre is 0.

3) Stage-two: Ranking and Filtering by MESA Domain Relatedness method

The roughly computing of domain relatedness of each node in Stage-one is somekind accurate, but didn't take the text content of each concept's Wikipedia page into consideration, which is more important in measuring the relatedness of a concept to the specific domain. So, we incorporated our novel Modified Explicit Semantic Analysis method to compute accurate semantic relatedness by using the content text.

To construct a comprehensive domain-specific lexical repository, we use the result of Stage-one to get further accurate results. We choose the first ranked 20000 nodes as word segmentation dictionary, and use the forward maximum matching method during the segmentation to keep the complete meaning of concepts, then we use this dictionary to process all the pages for the travsered concepts in Stage-one by using the zhwiki-all-pages dump xml document.

After word segmentation of related Wikipedia pages, we can represent the concept(page) as a set of the selected 20000 nodes. According to the frequency of each concept's occurrence, we can know how important the concept is to the page by computing the TF-IDF value. But, in this way, we didn't take the occurrence position of particular concept into consideration, which is important to our semantic relatedness computing. As a human-edited corpus resource, the different parts of Wikipedia page has its own function. Obviously, there is a difference of their role in the article,

hence we added a weighting scheme to the basic TF measure, based on the following parts of text:

See also text: This part of text is semantically related, which can also be called redirect text. Therefore, we assign the highest weight to the concepts in this type of text.

First paragraph text: Mostly, the first paragraph is an explanation paragraph whose text gives the main meaning of the concept, and the synonyms of the concept always occur here, which is highly related too. In this sense, we give the concepts in this text a relative high weight.

Info box text: Many Wikipedia pages contain the "info box" part, which gives the most basic information about the page concept. Therefore, we identify concepts in this part as the next most relevant concepts.

Category text: Each Wikipedia page belongs to some categories, while the direct child-parent relationship reflect some kinds of semantic relatedness, so the concepts in this part.

Regular text: The rest text in a Wikipedia page, which carry the least semantic information. In this paper we used the type weighting scheme shown in Table 1.

Table 1. Weights for different kind of text

The type of text	Type weight
See also text	6
First paragraph text	5
Info box text	3
Category text	2
Regular text	1

After word segmentation and WeightedTF counting, we map the concept(article) to words. Then we reverse the mapping to generate the inverted index which map the words to their occurred concepts(articles) by WeightedTF-IDF value.(That is the part above the dash line in Figure 2)

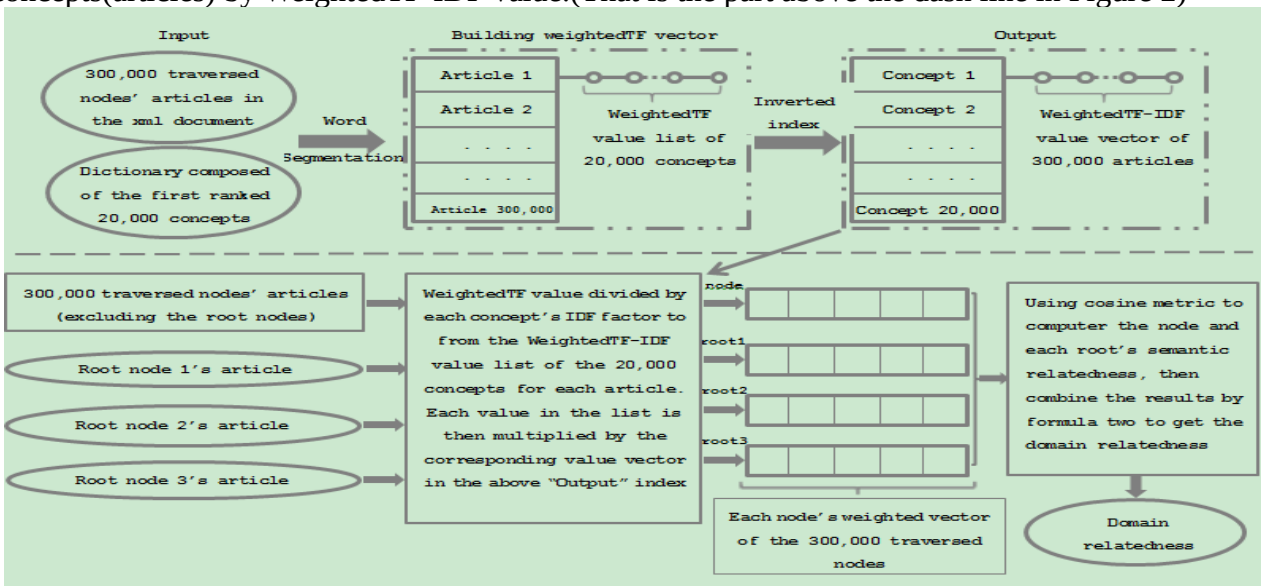

Figure 2. The procedures for Stage-Two

Then, we compute the traversed concept's semantic relatedness with the three root nodes we choosen. With regard to each traversed node a's article A, it can be represent by a set of concepts after word segmentation(A = $\{c_i\}$), and let $<v_i>$ be its WeightedTF-IDF vector, where v_i is the Weighted-weight of concept i. Let $<k_j>$ be an inverted index entry for concept i, where k_j quantifies the strength of association of concept i with traversed concept c_j, {c_j belongs to $c_1,...,c_N$}(where N is the total traversed nodes), so do root node r_1's article B. Then, the semantic relatedness between A and B can be computed by cosine metric of A's representation vector and B's representation vector.

Then, the final domain relatedness can be computed by formula 2, in which α, β are parameters. In our experiment, α is assigned to 0.6, β is assigned to 0.4.

$$R_{domain}(node) = \frac{Rel(node,root_1)+\alpha Rel(node,root_2)+\beta Rel(node,root_3)}{2}, \quad \alpha+\beta=1 \qquad (2)$$

Experimental Evaluation and Results.

1)Dataset and Evaluation Procedure

The experiment are done on the Chinese Wikipedia's dump service of March 01,2015, the downloaded xml document is about 41G. After the pre-stage processing we calculated that the entire document contains about 2 million article and category nodes. We choose three highly related and top root nodes to initiate our traversal procedure: computer science, information technology and networks. Our aim is to construct the lexical repository of domain Computer Science.

To compare the material quality of Chinese Wikipedia with its English counterpart, we use the algorithm proposed in [7], which noted as CT-BFS-en here, and get the test result as CT-BFS-zh. We use the same selection scheme as [7] to choose the most relevant pages of the specific domain. 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 samping point, 10 consecutive nodes are taken manually by a people knowledgeable in Computer Science, and whether the concepts are domain related is judged by 10 graduate students major in computer science to get the averaged result.

We take another criteria to test our method with our prevous work. Because we not only want to verify the built repository's concept coverage of specific domain, but also want to test the accuracy of our domain relatedness measures to facilitate our further learning material recommendation system. That is besides the "whether related" judgment, also the "how related" judgment. We sample the first 40,000 concepts of the ranked nodes, the sample interval is 2000 for the first 20,000 nodes, then for the interval 4000 from 20,000 to 40,000. 10 consecutive nodes are taken manually at each samping point, so we get 150 sample nodes. We choose 10 graduate student major in computer science, score the domain relatedness of each sample point. The results are averaged.

2)Results and Analysis

We can see from figure 1 that the Chinese-Wikipedia concepts of domain Computer Science we got are less than the English counterpart because the line declined more early, and the result line of Chinese Wikipedia shakes more(compare CT-BFS-en with CT-BFS-zh). That is mainly because the Chinese-Wikipedia is not as sound as its English counterpart, and the category structure in English Wikipedia is more accurate. For example the concept "semantic relatedness" occurs in English-Wikipedia but didn't occur in the Chinese-Wikipedia.

With our Multi-Roots traverse method, we can get more concepts and the relatedness shake is relieved, because we incorporate more concepts to traverse wider and in the domain relatedness computation we used multi-root metrics which can ease the tolerance in a situation that the node's parent category node is not incorporated into the spanning tree. The Two-Stage method gets the beset result, with more related concepts incorporated, and the shave is exactly small, which means the first ranked 20,000 nodes are highly related.

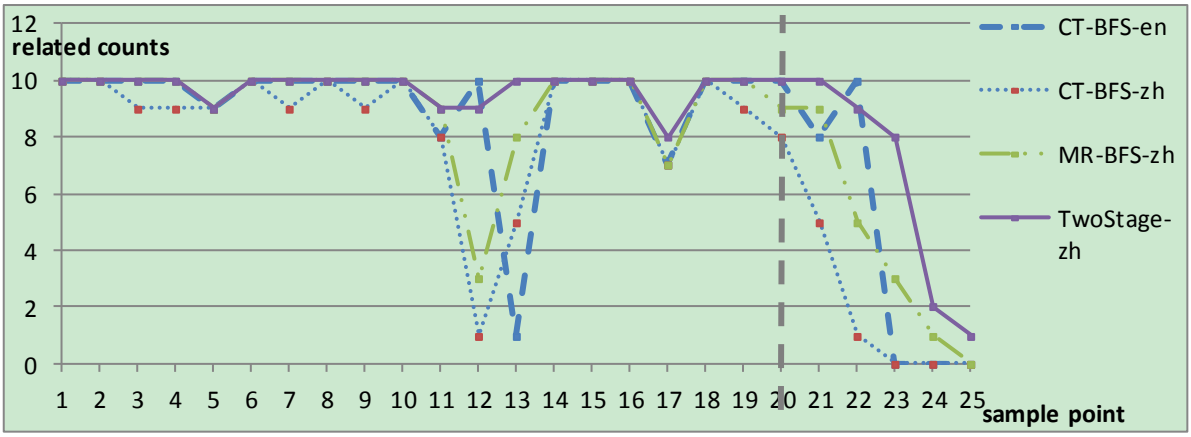

Fig.3. Evaluation Results for concept coverage and domain relatedness

Compared with our previous work, the novel two-stage method proposed in this paper get better result in the domain relatedness value test criteria. Also, it can show that the domain relatedness computation is more accurate, and the related concepts are more concentrated in the preceding part.

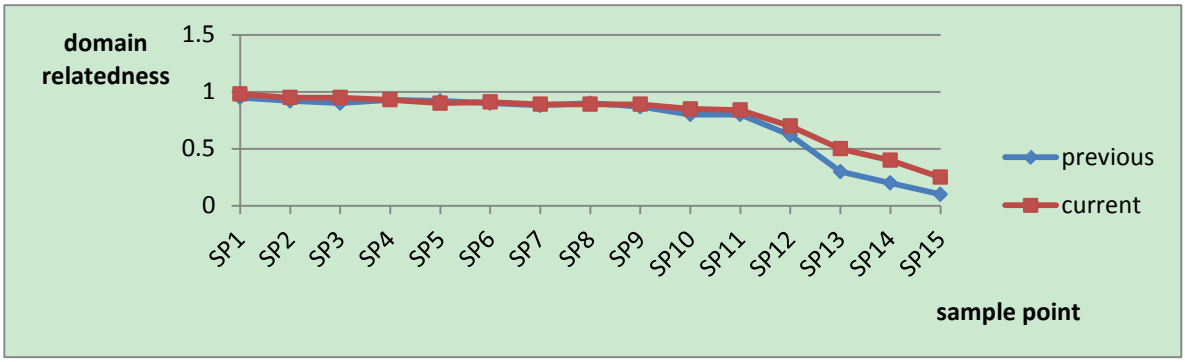

Fig.4. Domain Relatedness accuracy comparison with our previous work

Conclusion and Future Work

In this paper, a novel two-stage approach is proposed to building the domain-specific lexical repository for domain Computer Science. Our method takes multi-roots into consideration during the traverse procedure and use the text content in Wikipedia pages to compute the domain relatedness. We make different comparisons between our method and existing methods in two ways(whether related and how related), the result show that our novel method get excellent performance in building the repository not only in the concept coverage but also in the accuracy of the incorporated concepts. But the text processing workload of our method is heavy, which need to be optimized in the future work.

The method proposed in this paper can not only compute the domain relatedness of a concept to the specific domain, but also we can compute the semantic relatedness of arbitrary two nodes in the repository, which can be used to the recommendation of related studying materials. We can use the constructed domain-specific lexical repository to facilitate our studying material recommendation system and this will be discussed in our future work.

References

[1] Latifur Khan and Feng Luo. Ontology Construction for Information Selection. In proceedings of International Conference on Tools with Artificial Intelligence, 2002.

[2] P Cimiano, S Handschuh and S Staab. Towards the SelfAnnotating Web. In proceedings of the 13th International Conference on World Wide Web, 2004.

[3] David M. Computing Semantic Relatedness using Wikipedia Link Structure. Department of Computer Science, University of Waikato, Hamilton, 2007.

[4] Rui-Qin WANG, Measuring of Semantic Relatedness between Words based on Wikipedia Links. International Conference on Image, Vision and Computing(ICIVC 2012) vol.50.38.

[5] Strube M. and Ponzetto S.P. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In Proc. of the AAAI, Boston: IEEE, 2006:1419-1424.

[6] E. Gabrilovich and S. Markovitch, “Computing semantic relatedness using Wikipedia-based explicit semantic analysis”. In Proc. Of International Joint Conference on Artificial Intelligence(IJCAI 2007),pp.1606-1611,2007.

[7] Gaoying Cui, Qin Lu, Wenjie Li, and Yirong Chen. Corpus exploitation from Wikipedia for ontology construction. In European Language Resources Association, editor, Proceedings of the 6th International Conference on Language Resources and Evaluation. Marrakech, Morocco. (2008)

[8] Xinye Zhang, Xiu Li and Zhijian Ruan. A New Approach for Building Domain-specific Corpus with Wikipedia. The 2013 International Conference on Mechatronics and Industrial Informatics. (ICMII 2013, EI)