**Reconstruct Logical Hierarchical Sitemap for Related Entity Finding**

Qing Yang[1], Peng Jiang[1], Chunxia Zhang[2], Zhendong Niu[1]

[1]School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[2]School of Software, Beijing Institute of Technology, Beijing 100081, China
{yangqing2005, jp, cxzhang, zniu}@bit.edu.cn

**Abstract.** This Paper presents the work done for the TREC 2010 entity track. We concentrate on constructing enriched anchor text model by exploiting hierarchical information presented in web pages to retrieve promising pages, and heuristic rules to extract potential candidate entities by zooming in the right section.

## 1. Introduction

Different form traditional information retrieval, both content and structure information of web pages are critical to the success of web information retrieval. In recent years, many relevance propagation techniques have been proposed to propagate content information between web pages through web structure to improve the performance of web search. A large number of web sites contain lots of hierarchical information. In this paper, we first propose a novel method Logical Hierarchical Sitemap(LHS) to reconstruct logical sitemap and apply it in Related Entity Finding Task to find the relevant pages by integrating additional site level information . The result shows that the reconstruction of logical sitemap is effective.

## 2. Related Work

URL, as a Uniform Resource Locator for each web page, usually contains meaningful information for measuring the relevance of a web page to a query in web search. Some existing works utilize URL depth priors (i.e. the probability of being a good page given the length and depth of a URL) for improving some types of web search tasks. URL usually contains meaningful information for measuring the relevance of the web page to a query. Related works can be roughly divided into three categories: The first category is to use the length or depth of a URL as query-independent evidence in ranking[1-4]; the second cateogory is to use URL-based sitemap to enhance topic distillation[5, 6]; the third category addresses the issue of word break in URLs[1, 7]. Aixin et al. [8] Propose a web unit mining problem and construct site directory structure based URLs to mine web units.

But the problem is URL usually contains the physical web page organization structure instead of logical organization. Its name convention is at will of web authors. And there are many sites which are organized in a flat structure. Additionally, although most sites have sitemaps, those sitemaps are catered to search engines rather than to human. These result in the challenge of identifying the logical relation between of pages in a specified web site.

From observations from 5 billion datasets, actually it is possible to detect hierarchical information from web pages by deriving from menus, navigational bars, breadcrumbs etc. Seung etc.[9] proposes to construct hierarchical information structure f of sub-page level HTML documents to capture the hierarchical nature of the web. Rupesh etc.[10] Propose a web content structure based on visual representation which is called Vision-based Page Segmentation (VIPS) to improve relevancy and remove redundancy. In his seminal work on complex system, Simon argued that all systems tend to organize themselves hierarchically[11].

A typical site has a navigator bar at the top of page as the figure 1. It indicates obvious hierarchical information for a human observer. Benefit of Member Service consist Insurance Programs, Vehicle Discounts, Industrial Supplies, Travel & Entertainment, Products, and Services and are separated in five pages whose URLs

resides under the root of http://www.mdfarmbureau.com/. The relationship of the five pages cannot be predicted singly from their URLs.


Figure 1: A typical use of the navigation menu bar to describe the hierarchical relationship of pages in the same site. The hierarchy here indicates the organization of pages and is obvious to a human observer. Benefits of Member Service consist Insurance Programs, Vehicle Discounts, Industrial Supplies, Travel & Entertainment, Products, and Services.

This paper describes the details of the reconstruction of logical sitemap, whose goal is to capture and leverage the useful hierarchical information on the specified site to get the site level information. Our first contribution is that we propose a novel method LHS to reconstruct the logical hierarchical sitemap for a site. We employ the clueweb09 English part which is about 5 billion pages. Our second contribution is to integrate extracted site hierarchical information which is modeled as logical sitemap into search engines. We made a comparative study of the relevance propagation in the context of Related Entity Finding task in TREC2010[1].

## 3. Logical Hierarchical Sitemap

Our goal is to extract hierarchical information to construct logical sitemap from the raw Web in a site. A HTML list always presents hierarchical information of the navigation and organization of pages in a site. This hierarchical information also indicates the content of the responding pages explicitly. Actually, anchor text is the most direct information for the target page and is already used in broad way. But it just predicts the target page and has no information for the relation of pages.

From observations of the dataset, the most common pattern to present hierarchical information is HTML lists. The most common HTML lists are ordered and unordered lists: An unordered list starts with the <UL> tag. Each list item starts with the <LI> tag. An ordered list starts with the <OL> tag. Each list item starts with the <LI> tag. From observations from the corpus, the most common pattern is to use <DIV> section to partition sections as demonstrated by Figure 2.

Also, another popular pattern is to use <table> tag which use <br> in their <td> sections to present hierarchical information. Of cause, there are some other formats to present hierarchical information by using CSS style. We only considerate HTML lists in this paper.

[1] http://trec.nist.gov/act_part/tracks.html


Figure 2: DOM tree illustration of an example

It is obvious that the categories (*Member Services->Benefits->Insurance Programs*) indicate the content of the target URL (“*/Insurance.asp*”). But, if we just browse the URL (http://www.mdfarmbureau.com/Insurance.asp), it is not explicit to tell what the page is about.

From this figure, we devise a method LHS – Logical Hierarchical Sitemap to extract such hierarchical information of a site. This method is bottom-up. We first extract links from page, and then parse hierarchical information for the extracted links. The pseudo code of parse hierarchical information is showed as the following:

```
// parse hierarchical information for a link node
category parseHierarchy(node,category){
      // Locate category node from the current node
      while ( (parent-node = parent node of linknode) != null    &&
                  isCategorynode(parent-node));{
      // Get the current category label
        category = category + “ < “ +parent-node.text;
        parent-node = parent-node.getparent();
      }
}
```

The function of isCategorynode is not easy to define for the free styles of web designers. From observation of the dataset, we take into consideration textnode, linknode and img node which has a link.

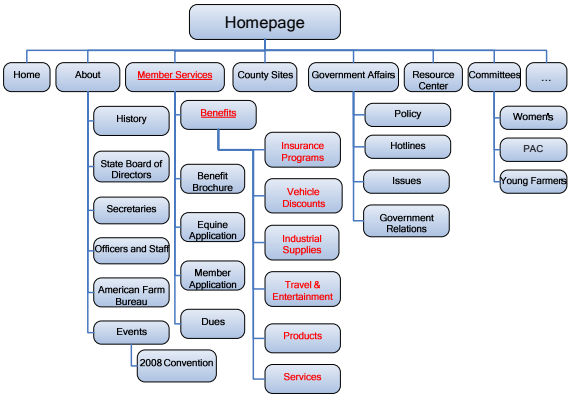


Figure 3: Result of Logical Hierarchical Sitemap

After we get all the pages’ categories, the logical hierarchical sitemap of a site can be illustrated by Figure 3. The logical sitemap represents the logical relation of pages which reside in this site.

Additionally, for some sites which have no such hierarchical information available, we employ their URLs which indicate sites directory structure, or similarity of page structure in the same site to construct virtual groups, especially some small sites which are organized in a flat structure.

## 4. Integration of Existing Search Engine.

After we reconstruct the logical hierarchical sitemap, it is easy to integrate it into existing search engines. Actually, the hierarchical information enhances the anchor text for links. For simplicity, the hierarchical categories are used as index terms for the target pages which act the same as its anchor text. We use indri[2] to index the dataset. First, we extract the hierarchical information and index the dataset by adding *inlink* field in the configure setting by adding the hierarchical information for the responding page.

## 5. Related Entity Finding Task

The related entity finding task is proposed from 2009 and continues this year. This task is defined as the following:

Given an **input entity**, by its name and homepage, the **type of the target entity**, as well as the **nature of their relation**, described in free text, **find related entities** that are of target type, standing in the required relation to the input entity[3].

This task shares similarities with both expert finding (in that we need to return not "just" documents) and homepage finding (since entities are uniquely identified by their homepage). However, approaches to address this task need to generalize to multiple types of entities (beyond just people) and return the homepages of multiple entities, not just one. Also, the topic defines a focal entity to which returned homepages should be related.

For this year, our goal is to construct logical hierarchical sitemap to integrate the hierarchical information into existing search engine to get the relevant pages which have the results residing in tables or lists especially.

### 5.1 System Overview

We complete our experimental system architecture as a pipeline architecture.

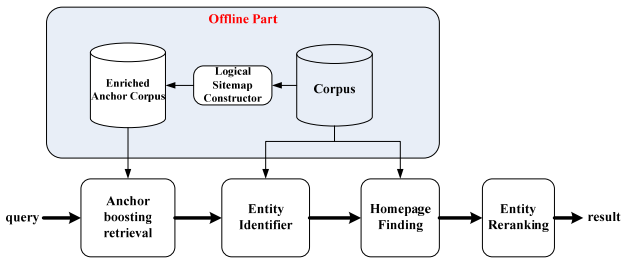


**Figure 4: the Related Entity Finding System Architecture.**

We outline the retrieval framework as above. From TREC-supplied query topics, we first analyze the narrative of every query topic and extract keywords and terms. Then we send query strings to the indri search engine, and get result pages. From the result pages, we employ some OpenNLP components and Stanford's parser to identify target typed named entities. We rank entities by counting number of entity occurrences in a single result page. It means that the more entity occurrences in a single page, the more likely the requested entities. We get the top 150 entities at most and post the top 150 entities' names to Google[4] and Realnames[5] search engine respectively. Then, we rank the result entities by the score of their homepages from five sources, which are anchor text's target link, the directed URLs, top 10 of Google's results, top 10 of Realnames' results and the URLs with the same domain in the Clueweb09 Category A English part.

---

[2] http://www.lemurproject.org/indri/
[3] http://ilps.science.uva.nl/trec-entity/guidelines/
[4] http://www.google.com
[5] http://www.realnames.com

### 5.2 Query Topic Parsing

As for a QA, Ephyra[6] spends much effort to analyze question syntactically and semantically. To identify answer type, it employs machine learning scheme to train answer patterns and identify answer types. This tricky phase is not necessary for the related entity track because the target entity type is explicitly given. As for the explicit entity name, we just add it to the query string without expanding it. Additionally, we prefer to pages in the site where the supplied source entity's homepage resides.

To take the 39th query topic for an example, query string is focus words i.e. *benefits, members and "Maryland Farm Bureau"*. The query to be sent to indri is as the following:

#combine (#1(mdfarmbureau com).url benefits members "Maryland Farm Bureau").

### 5.3 Named Entity Identification

In this task, the type of target entity is restricted in four types: person, organization, location, and product. Generally speaking, the first three types are easier to be identified. However, for the product type, it is rather difficult to be identified correctly. To deal with this issue, we resort to Wikipedia online knowledge database whose pages always have a category label. We made a hardworking to find that those *introductions*, *productions*, *products*, *games*, *software*, *hardware* etc. category labels are almost classified into product type. It helps us to extract 43,393 product names. Also, by using the same method, we extracted 18,181 organization names and 118,002 person names.

Additionally, inspired by [12], we discriminate extracted entities by their locations in DOM tree. Therefore, our method is biased to extract multiple entities in tables and lists, but it is not restricted those entities which reside in tables or lists. In the experiment setting, $\theta$ is set as 20 and $\alpha$ as 10.

**Table 1. Heuristic Characteristics of Identifying Relationship**

| Characteristics | Description | Span |
|---|---|---|
| Title | Page title | global |
| URL | Page address | global |
| Anchor text | Enriched anchor text of page | global |
| Headings | Heading sections | local |
| Emphasizing strings | Em, strong, u, I, b, font size, background color | local |
| Table's *th* field | Header column of table | local |
| Selection selected option | First indicating option or not | local |
| Length of identified string | 1 <= length <= $\theta$ words | local |
| Similarity with relation r | Cosine similarity | local |

**Table 2. Heuristic Characteristics of Identifying Entities**

| Characteristics | Description | Span |
|---|---|---|
| Formatting tag | Strong, em, b, I,fontsize | Intra-page |
| Link text | Link text of link node | Intra-page |
| Repeated patterns | Tag path, string pattern | Intra-page |
| Parallel relationship | Parallel relation between instances | Intra-page |
| Length of identified string | 1<=length <= $\alpha$ | Intra-page |
| **Not complete sentence** | optional | Intra-page |
| Distance of ***r and e*** | Relational position distance | Intra-page |
| Site page frequency | Difference of navigational items | Inter-page |
| Formatting tag | Strong, em, b, I,fontsize | Intra-page |

### 5.4 Related Entity Candidates Ranking

It is well known that the search results ranking is not necessary responding to those extracted entities ranking. We apply the following formula to rank related entity candidates.

[6] http://www.ephyra.info/

$$Er = \sum_{i} Pi(\sum_{e \in E} occurence(e))$$

P refers to Web page which the entities are extracted from. E refers to the set of extracted entities. *Pi*(Occurrence(*e*)) refers to the occurrence of *e* extracted from Web page *Pi*. Under the hypothesis that the more entities are extracted from a page, the more likely the extracted entities are the results, we rank the entity by the sum of the occurrence of entities in the pages which the entity resides in. The rank function is biased for those result entities presenting in tables or lists.

### 5.5 Entity Homepage Finding and Ranking

The procedure is to identify the extracted entities' homepages. We consider five sources for finding homepages ordered by confidence by assuming that the candidate entities are all correct.

**Table 3. Sources of homepages of entity**

| | |
|---|---|
| #1. | Target URL link of entity name as anchor text |
| #2. | The redirected URL of the target link by web server |
| #3. | The top 10 URL returned by Google with the query of entity name |
| #4. | The top 10 URL returned by Realnames with the query of entity name |
| #5. | The URLs of the above URLs within the same domain identified by homepage classifier |

We rank the target URLs by the order of the sources which Table 1 indicates.

### 5.6 Experimental Setup

The Indri Search Engine[7] was used to index both collections by removing a standard list of 418 INQUERY[13] stopwords and applying Krovetz stemmer.

In a separate process, we make use of Java Class WarcRecord [8] and WarcHTMLResponseRecord[9] for reading WARC record and by parsing pages with HTMLParser[10] extract hierarchical information of the site-level knowledge and create an enriched anchor text index by using Indri Search Engine without removing stopwords and stemming. We did not take special consideration on those Wikipedia documents in the corpus.

### 5.7 Results and Discussions

The official test set contained only 70 queries including 20 of last year. Given the facts that this is a new task, a new collection, and we have a relatively small number of topics, evaluation will primarily focus on analysis of the results and runs on a per-topic basis, rather than on average measures.

We submit 3 runs as official runs: *bitDSHPRun*, *bitDSRRun* and *bitRFRun*. All three runs use the same extracted entities list, and differ in ranking settings. *bitDSHPRun* prefer homepages from Google than from Realnames, and *bitDSRRun* vice verse. *bitRFRun* normalize person names with the rule for the format like "last name, first name" as the canonized format as "first name last name" on the base of *bitDSHPRun*. The 2010 edition of the track creates 50 new topics besides 20 topics of last year. The submitted runs include the 70 topics. But the old 20 topics are not taken into account for the official ranking systems. Additionally, the final evaluation just assessed 47 topics and leaved alone three topics: 35, 46 and 59. The probable reason is these topics have no results in the dataset.

---

[7] http://www.lemurproject.org/indri/

[8] http://boston.lti.cs.cmu.edu/clueweb09/wiki/tiki-download_file.php?fileId=2

[9] http://boston.lti.cs.cmu.edu/clueweb09/wiki/tiki-download_file.php?fileId=3

[10] http://htmlparser.sourceforge.net/

The following measures are used[11]:

- NDCG@R is calculated to rank R, where R is the number of primary homepages for that topic, where a primary homepage gets gain 3 and a relevant homepage gets gain 1. R-precision is computed likewise.
- P@R and MAP, computed for relevance level 1 (both relevant and primary accepted) and 2(only primary accepted)

For all metrics, only previously unseen entities will be rewarded; i.e., if a primary/relevant homepage has already been returned at earlier ranks for the same entity, then it will count as non-relevant. We summarized the official result as Table 4.

Table 4: Results on Related Entity Finding

| Run | P10 | nDCG_R | Map | Rprec |
|---|---|---|---|---|
| bitDSHPRun | 0.3766 | 0.3694 | 0.2726 | 0.3075 |
| bitDSRRun | 0.3766 | 0.3694 | 0.2726 | 0.3075 |
| bitRFRun | **0.3936** | **0.3897** | **0.2876** | **0.3209** |

Figure 5 show our system's per-topic performance in terms of nDCG_R, alongside with the per-topic median and best performance in all attended groups. The dots indicates the topics of which bitRFRun obtains the best performance.

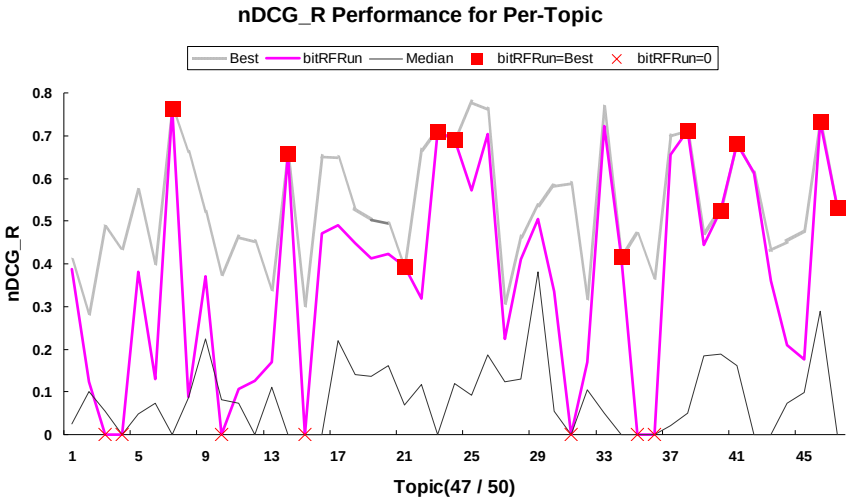


Figure 5. nDCG_R performance for each topic

From Figure 5, it indicates that there is 11 topics get the top score in the pool, 7 topics zero and others much better than median results. It avails much from tables or lists totally. For the evaluation is end to end, it requires much efforts to decide which parts gain more than others, retrieving promising documents, zooming in the right sections in documents or finding homepages and ranking.

In summary, all these stages gain the final scores. Intuitively, these stages may have their contributions. To explore the stages separately is our next work.

In future work, there are a number of things for us to explore. First, we will explore more efficient way to automatically construct queries. We observed there are some inverse relationships. Take the topic 3 "Students of Claire Cardie" for example, it is effective to query by "advisor Claire Cadie" to get more relevant results. The second is that the target type is defined in a general sense. When we decide the target entity it is more suitable to be constrained as a concrete type. The third is to extract more detailed logical hierarchical site level knowledge from structures in HTML files. In addition to HTML lists, there are many kinds of structure on the Web, including the deep web, tables, tagged items, ontologies, XML documents, spreadsheets, and even extracted language parses[14].

[11] http://ilps.science.uva.nl/trec-entity/guidelines/

**Acknowledgments.** This work is supported by the grant from Chinese National Natural Science Foundation (No: 60705022).

## 6. References

1. Ogilvie, P. and J. Callan, *Combining Structural Information and the Use of Priors in Mixed Named-Page and Homepage Finding.* In Proceedings of the Twelfth Text Retrieval Conference (TREC-12, 2003: p. 177-184.
2. Nick, C., et al., *Relevance weighting for query independent evidence*, in *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*. 2005, ACM: Salvador, Brazil.
3. Wessel, K., W. Thijs, and H. Djoerd, *The Importance of Prior Probabilities for Entry Page Search*, in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. 2002, ACM: Tampere, Finland.
4. Westerveld, T.H.W., W. Kraaij, and D. Hiemstra, *Retrieving Web Pages using Content, Links, URLs and Anchors*, in *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*. 2002, National Institute for Science and Technology (NIST): Gaithersburg, Maryland, USA.
5. Qin, T., et al., *A study of relevance propagation for web search.* In SIGIR 28, 2005: p. 408-415.
6. Ji-Rong Wen, R.S., Deng Cai,Kaihua Zhu, Shipeng Yi,Shaozhi Ye and Wei-Ying Ma, *Microsoft Research Asia at the Web Track of TREC 2003.* 2003.
7. Chi-Hung, C., D. Chen, and L. Andrew, *Word segmentation and recognition for web document framework*, in *Proceedings of the eighth international conference on Information and knowledge management*. 1999, ACM: Kansas City, Missouri, United States.
8. Aixin, S. and L. Ee-Peng, *Web unit mining: finding and classifying subgraphs of web pages*, in *Proceedings of the twelfth international conference on Information and knowledge management*. 2003, ACM: New Orleans, LA, USA.
9. Seung Jin Lim, Y.-K.N., *Constructing Hierarchical Information Structures of Sub Page Level HTML Documents.* CIKM, 1999(1999): p. 466-484.
10. Rupesh, R.M., M. Pabitra, and K. Harish, *Extracting semantic structure of web documents using content and visual information*, in *Special interest tracks and posters of the 14th international conference on World Wide Web*. 2005, ACM: Chiba, Japan.
11. H.A.Simon, ed. *The Sciences of the artifical*. 3rd ed. 1981, MIT Press,Cambridge: MA.
12. Valter, C., M. Paolo, and M. Paolo, *Clustering web pages based on their structure.* Data Knowl. Eng., 2005. **54**(3): p. 279-299.
13. Broglio, J., J.P. Callan, and W.B. Croft, *An Overview of the INQUERY System as Used for the TIPSTER Project.* 1993, University of Massachusetts.
14. Jayant Madhavan, A.H., Shirley Cohen, Xin (Luna) Dong, Shawn R. Jeffery, David Ko, Cong Yu, *Structured Data Meets the Web: A Few Observations*. 2006, Data Engineering Bulletin.