Querying and Clustering Web Pages about Persons and Organizations

Shiren Ye, Tat-seng Chua, Jeremy R. Kei
School of Computing, National University of Singapore, Singapore, 117543
Yeshiren2000@hotmail.com, {chuats, jkei}@comp.nus.edu.sg

Abstract

One of the most frequent Web surfing tasks is to search for names of persons and organizations. Such names are often not distinctive, commonly occurring, and non-unique. Thus, a single name may be mapped to several entities. The paper describes a methodology to cluster the Web pages returned by the search engine so that pages belonging to different entities are clustered into different groups. The algorithm uses a combination of named entities, link-based and structure-based information as features to partition the document set into direct and indirect pages using a decision model. It then uses the distinct direct pages as seeds to cluster the document set into different clusters. The algorithm has been found to be effective for Web-based applications.

1. Introduction

Queries about persons and organizations (PnOs) are common search requests during Internet surfing. The results returned for such queries are usually sufficiently accurate for its purpose and top ranked results usually include the target entity (TE). There are however several problems and issues with these search results as outlined below:

- The number of pages returned by a search engine may reach thousands. However, most users only have patience to browse the first few pages only.
- Search results may contain several target entities whose names are the same as the query string. It would facilitate user browsing if the search results can be grouped into different clusters, each containing pages about different entities.
- Some useless pages are completely irrelevant but are displayed nonetheless as return results because they contain phrases that are similar to the name of requested PnOs.
- The low-ranking pages listed at the rear of the result list may often be of only minor importance, but they are not always useless. In some cases, novel or unexpectedly valuable information can be found in these pages.

As shown in Figure 1, when we submit the query "*Sanjay Jain*" to Google, at least ten different persons named "*Sanjay Jain*" will be returned. Here, pages (a) and (b) are the homepages of two different persons: a computer scientist in Singapore and an economist in Virginia. Page (c) is an introduction of a book authored by the person in page (a). Page (d) is the description of another person, the Chairman of a company, but its style is different from that of earlier pages. It can be seen that the search engine returns a great variety of both related and unrelated results. If we are able to identify and partition the results into the clusters about different individuals, it will facilitate users in browsing the results.

The aim of this paper is to develop a search utility to support PnO searches on the Web. In particular, it partitions the search results returned by a PnO name query into distinct clusters, with each containing document pages about a particular target entity. For instance, for search on person named "Sanjay Jain", we expect to get one cluster about Sanjay Jain in Singapore, another about Sanjay Jain in Virginia etc. The unknown fragment pages are discarded into an unknown cluster. So it is different from general document and web clustering problems.

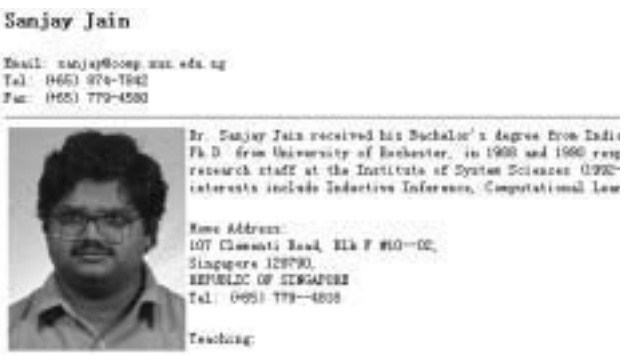
To support this process, we need to identify three types of pages from the returned pages:

- Direct page (DP): Its content is almost entirely about the users' focus. Examples of such pages include the homepages, profiles, resumes, CVs, biographies, synopsis, memoirs, etc. The relevance between them and the query is the highest and could be selected as the seed (center) of the corresponding cluster.
- Indirect page (IDP): In such pages, the target entity is only mentioned occasionally or indirectly. For instance, the person's name may appear in a page about the staff of a company, record of a transaction, or the homepage of his friend.
- Irrelevant page: the page is not about any target entity.

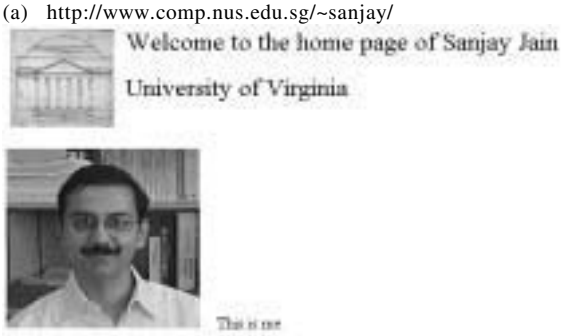
We use a combination of named entities, link and structure information extracted from the original content as features to perform the clustering. Our tests indicate that this approach is promising. The main contribution of this research is in providing an effective clustering methodology for PnO pages.

Briefly, the contents of this paper are organized as follows. Section 2 introduces related work and Section 3 discusses named entity based, link-based and structure-based document features. Section 4 presents the algorithm to identify DPs and seeds of the clusters. The method of delivering IDPs into clusters is described in Section 5. The results of our experiments and the conclusions are respectively presented in Sections 6 and 7.

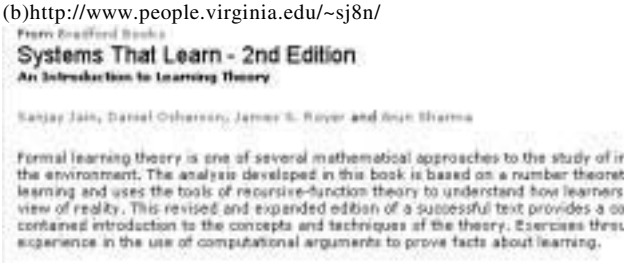
Proceedings of the IEEE/WIC International Conference on Web Intelligence (WI'03)
0-7695-1932-6/03 $17.00 © 2003 IEEE



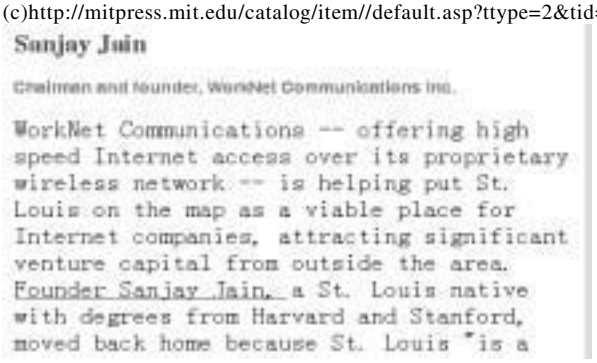
(a) http://www.comp.nus.edu.sg/~sanjay/



(b)http://www.people.virginia.edu/~sj8n/



(c)http://mitpress.mit.edu/catalog/item//default.asp?ttype=2&tid=727



(d)http://www.bizjournals.com/stlouis/stories/2000/01/31/focus37.html

Figure 1. Typical pages when "Sanjay Jain" is submitted to Google (Partial list)

2. Related Work

One of the important tasks in our research is to develop techniques to identify direct pages to PnO queries. Our direct page finding task is similar to but more complex than the home (entry) page and key resource finding tasks in TREC [1]. The homepage finding task [2] aims to find the home or site entry page about the topic. The home page usually has introductory information about the site and navigational links to other pages in the site. It is a subset of direct page as a direct page may include other type of PnO related pages such as the resume or profile. The key resource finding task [3] aims to find pages that contain lots of information, usually in the form of links to relevant pages, about the topic. A key resource page can therefore be located based on the number of out-links a page has to useful authority pages. In contrast, a direct page is more self-contained and includes useful information about a specific PnO with links to other pages within the sites.

The main approaches for finding homepages exploit content information as well as URL and link structure [4]. It was generally found that using only content information could achieve a mean reciprocal rank (MRR) score of only 30% based on the top 10 ranked results. However, combining content with anchor text and URL depth [5] could achieve an MRR of 77.4%, which is the best reported result in TREC10. Craswell [6] confirmed that ranking based on link anchor text is twice as effective as ranking based on document content. Kraaij [7] further analyzed the importance of page length, the number of incoming links and URL form such as whether it is of type root, sub-root, index or ordinary file. They discovered that URL form was a good predictor and reported a MRR of over 80% on a subset of WT10g corpus.

For key resource task, Zhang et al. [9] employed techniques based on link structure, link text and URL, especially the out-degree, of the pages. They achieved the best results in TREC-11 evaluation with a precision of 25% among the top 10 retrieved pages. However, the second best performing run [10] was a straightforward content retrieval run based on Okapi BM25, and achieved a precision of about 24%. The overall results reveal that page content is as good as non-content features in key resource finding task.

After we have found distinct direct pages for target entities (TEs), the second stage is to perform clustering to deliver IDPs for the corresponding TEs. PnO page clustering is a special case of web document clustering, which attempts to identify groups of documents that are more similar to each other than the rest of the collection. Several new approaches have emerged to group or cluster Web pages. These include association rule hyper-graph partitioning, principal direction divisive partitioning [11], and suffix tree clustering [12]. The Scatter/Gather technique [13] clusters text documents according to their similarities and automatically computes an overview of documents in each cluster. Steinbach et al. [14] compared a number of algorithms for clustering web pages on a variety of test corpuses. Their reported performance in terms of F_1 measure varies from 0.59 to 0.86.

Many of these traditional algorithms employ the bag of words representation to model each document. The resulting feature space tends to be very large, in the order of ten of thousands. As a result, most traditional clustering algorithms falter due to the problem of data sparseness when the dimensionality of the feature space becomes high relative to the size of document space. Because of the unpredictable performance of clustering methods, most search engines at present do not deploy clustering as a regular procedure during information retrieval.

3. Document Feature Representation

Most clustering approaches compute the similarity (distance) between a pair of documents using the cosine of the angle between the corresponding vectors in the feature space. Many techniques, such as TFIDF and stop word list [15], have been used to scale the feature vectors to avoid skewing the result by different document lengths or possibly by how common a word is across many documents. However, they do not work well for PnOs. For instance, given two resume pages about different persons, it is highly possible that they are grouped into one cluster because they share many similar words and phrases, such as "graduate", "university", "work", "degree" and so on. This is especially so when their style, pattern and glossary are also similar. On the other hand, it is difficult to group together a news page and resume page about the same target entity, due to the diversity in subject matter, word choice, literary styles, document formats and length among them. To solve this problem, it is essential to choose the right set of features that reflect the essential characteristics of target entities.

In general, we observe that in a direct PnO page (DP), there is typically a large number of NEs, such as the names of schools, contact information (phone, fax, e-mail, and address), working organizations and experiences (time and organizations). Here, PnO related NEs include person, location and organization name, time and date, fax/phone number, currency, percentage, e-mail and so on. To support our claim, we analyzed 1,000 PnO pages together with 1,000 other type of pages that we randomly obtained from the Web. We found that the percentage of NEs in PnO direct pages is at least 6 times higher than that in other types of pages, if we ignore NEs of type number and percentage. We could therefore use NEs as the basis to identity PnO pages.

The finding is quite consistent with intuition, as NEs play important roles in semantic expression and could be used to reflect content of the pages, especially when human activities are depicted. Our analysis also shows that NEs is good in partitioning pages belonging to different persons or organizations, and the use of frequent phrases and words, such as degree, education, work etc, is not effective for this task.

However, not all pages with many NEs are DPs. Examples of such pages include attendee lists of conferences and stock price lists etc. We thus need to further check the roles played by the NEs in this text. The rationale is that a DP is highly likely to repeat its name in its URL, title, or at the beginning of its page. In general, if the TE appears in important locations, such as in HTML tags <title>, <H1> and <H2>, or appears frequently, then the corresponding pages should be DPs and their topic is about the users' target.

Furthermore, we know from the TREC evaluations that URL, HTML structure and link structure tend to contain important heuristic clues for web clustering and information retrieval [16]. Links could be used to improve document ranking, estimate the popularity of a web page, and extract the most important hubs and authorities related to a given topic [17]. Moreover, links, URLs and anchors could improve the results of the content-only approach for IR [5]. A short DP, even though it may contain few NEs, usually has many links to those pages referring to TE. The positions of and the HTML markup tags around the NEs could provide hints to the role of these entities in the corresponding page. To better identify the role of links in DP, we further identify the form of URLs as: root (entry page of site), sub-root, index and ordinary file. The URL form has been found in [7] to be a particularly good predictor for finding home pages.

Based on the above discussion, we combine three categories of features to identify DPs and IDPs. They are the named entities, links and structure-based features. The resulting set of features, as listed in Table 1, can be considered as original feature transformation. As the number of such features is smaller than the number of tokens in the collection, there is considerable dimension reduction. This will alleviate the problem of low quality of clustering because of data sparseness when the sample size is small.

4. Identifying Direct Pages as Cluster Seeds

Direct pages (DPs) can be used as candidate seeds to divide the retrieved documents into clusters of distinct TEs. In case where there is more than one DP about a TE, we need to select the best one as the seed for clustering. To select the best DP of a TE, we therefore need to solve two problems. First we must be able to identify a DP from the collection. Second, in the case of multiple DPs for the same TE, we must be able to select the best one.

The process is carried out as follows. First we view the identification of DPs as a classification problem of dividing the document collection into the DP and IDP sets. Here we employ the decision tree to predict whether a page is a DP or IDP based on the feature set as listed in Table 1.

Table 1. Features of web pages representation

Feature	Explanation
|PER|	Number of persons
|ORG|	Number of organizations
|Mail|	Number of E-Mail addresses
|number|	Number of numeric; fax, phone number and zip code are included; but the series of number list are ignored
|NE|	Sum of the above NEs
|Word|	Number of words excluding the HTML tags
|Token|	Number of all tokens
|NE|/|Token|	Ratio of |NE| and |Token|
|NE|/|Word|	Ratio of |NE| and |Word|
Target_Title	Boolean; whether TE appears in the title, head or the beginning of the page
|in-Link|	Number of in-links
|out-Link|	Number of out-links
|allLink|	The sum of in-links and out-links
URL_Depth	The depth of URL
URL_Form	Four types of forms: root; subroot; index/path; file
|TE|	Number of TEs appearing in the page
Target_ URL	Boolean; Whether TE or its variant appears in URL. E.g. target is "Sanjay Jain" and URL is "http://www.comp.nus.edu.sg/ ~sanjay/"

Next, we need to resolve the case of multiple DPs found for a TE. We observe that if both the homepage and resume of the same person are selected as DP, then these two pages will share many similar NEs related to this specific person, such as the university graduated, employers, etc. Thus we could evaluate the similarity between two DPs by examining the overlaps in the instances of unique NEs. Here we use TFIDF to estimate the weight of each unique NE as follows.

$$W_{i,j}=tf_{i,j}*\log(N/df_i) \quad (1)$$

where $tf_{i,j}$ is the number of NE i in page j; df_i the number of pages containing NE i; and N the total number of pages.

The similarity of the DPs, p_i and p_j, could be expressed by their cosine distance as:

$$sim(p_i, p_j) = \frac{\sum_k (w_{k,i}{}^c * w_{k,j})}{\sqrt{\sum_k (w_{k,i})^2 * \sum_k (w_{k,j}{}^c)^2}} \quad (2)$$

If $sim(p_i,p_j)$ is larger than a pre-defined threshold τ_1, then p_i and p_j are considered to be similar. The page that has more NEs will be used as the seed and the other will be removed. Because the number of DPs is a small fraction of the search results, and the number of NEs in DPs is usually less than hundreds, thus the computational cost in eliminating redundant DPs is acceptable.

Algorithm 1 summarizes the procedure to identify seeds of clusters.

Algorithm 1:

```
Detect_seed(page_set) {
  page_set = {the set of all pages found};
  set seed_set=null; //the collection of candidate seeds
  for each (p in page_set){
    build transformed feature set of p_i
    if (decision_tree(p_i) == TRUE)
      move p_i from page_set into seed_set;
  }
  for each pair {p_i, p_j} in seed_set:
    if (Sim(P_i,P_j)> τ_1){
      if (N_NE in p_i >N_NE in P_j) then
        move p_j from seed_set into page_set;
      else
        move p_i from seed_set into page_set;
  }
  return seed_set;
}
```

At the end of the process, the pages remaining in the seed_set could be used as seeds for the clusters. They are representative of distinct entities named in the query.

5. Delivering Indirect Pages to Clusters

Compared to DPs, IDPs provide less information about the TE. Nevertheless, it does not mean that they are less important. Actually, the information extracted from IDP may be more novel and provide more valuable information to the users. In general, IDP could provide additional information such as the activity or experience of the TE; and support or oppose the content in DP irrespective of whether they are consistent or not. Most importantly, IDP may provide critical or negative information that is not contained in the DP. For instance, a report of a company involving in a fraud may be ranked at the bottom of thousands of returning pages, but such pages may be significant to users in correctly evaluating the worthiness of the company. It can thus provide important information to evaluate the TEs fairly and integrality.

Therefore, we must explore an approach to link DPs and IDPs properly. In other words, we want to add IDPs into clusters anchored by the seeds (DPs). We make the assumption that clusters do not overlap and an IDP can be assigned to only one cluster. This is reasonable as it is unlikely to have the same name of different entities being mentioned in the same page.

As discussed earlier, we use the entities extracted from the original sources to calculate the distance between two pages. In topic locality assumption theory [7], pages connected by links are more likely to be about the same topic than those that are not. It is therefore reasonable to extend cluster along links via spreading activation or to perform probabilistic argumentation. We can also assume that pages sharing more entities, including links, URL and NEs, should be grouped together. This is consistent with

the intuition that the TEs in two pages having same e-mail, birth date or birth place may have some intrinsic associations. Also, pages that link to the same root or each other may belong to the same TE. So these evidences provide support for them to be grouped together.

In addition, the similarity between two entities is beyond the simple exact matching. For instance, "Sanjay Jain" may be different from "Sanjay", but their similarity is not zero because the latter is an informal expression of the former. Here we map all NEs into the formal format before they are compared. The situation in URL and links are more complex. If the roots of URLs are the same (such as www.xxx.com and www.xxx.com/aa), or components of URLs are similar (such as www.xxx.com and www.aaa.xxx.com), there should have a non-zero similarity. Let S_a and S_b the number of segments of link a and link b that are separated by dot or slash ("www" is ignored), and S_{ab} be the number of identical segments among them. The similarity Sim(a,b) between a and b is calculated as

$$Sim(a, b)=S_{ab} / (S_a*S_b)^{1/2} \quad (3)$$

Finally, we derive the similarity between an indirect page i and seed j, $Sim(Page_i, Seed_j)$, by combining the similarities between NEs, links and URLs (Eqn.(3)). We now outline the algorithm to select and link IDPs to a seed cluster.

Algorithm 2:

```
Arrange_indirect_page (page_set, cluster_set)
//clusters are represented by their seeds
{
  set unknown_set=null;  //collection of unknown pages
  for each (page_i in page_set)
  {
     j = arg max sim(page_i, seed_j)
    if (j>τ2) j
      add page_i into cluster_j;
    else
      add page_i into unknown_set;
  }
}
```

6. Overall Procedure

We now outline the overall process of PnO searches on the web. The user first submits a target entity name as the query to the system. The system then downloads the list of pages P_{all} related to the target. This step may involve other meta search engines. Second, a classifier is initiated to partition P_{all} into two groups: the set of DPs, S_{DP}, and the set of IDPs, S_{IDP}. Third, only distinctive pages about different TEs in S_{DP} are used as seeds of the clusters. The other redundant pages in S_{DP} are moved to S_{IDP}. Fourth, each page p_i in S_{IDP} will be clustered to the closest cluster whose seed is the nearest to the current page. If p_i cannot be matched to a sufficiently similar seed, i.e. the similarity between them is less than τ_2, it will be discarded into an unknown set. Fifth, we use the name of organization (or person) that appears in the seed as the label to the corresponding cluster. The resulting set of clusters found is then presented to the users. Sixth, when user submits more constraints, for example, using the term "Virginia" to constrain the query "Sanjay Jain", the system will utilize the constraint to rank the clusters so that the more relevant cluster appears at the top. Lastly, information in each cluster can be extracted into a predefined template as concise summary to the users. This final step is the subject of further research.

7. Experiments and Discussions

7.1. Selecting Test Samples from the Web

Experiment of web information processing is a time-consuming task, where each search typically returns hundreds, or even thousands of pages. Moreover, evaluating the effectiveness of clustering is notorious even though there are many guidelines to measure the quality of clustering such as the entropy measures, clustering error, and average precision [18]. Because of the lack of comparable standard test data for our task, we derived a set of web pages for testing based on the following methodology.

a) We collected the names of 30 persons and 30 organizations from Yahoo (www.yahoo.com) and MSN (www.msn. com). In order to conduct meaningful tests, we removed PnOs that belong to large companies and famous persons (such as Microsoft). This is because there would be too many pages in the search results for such PnO names. For example, Google returns 2,880,000 pages for Microsoft, and the first hundreds of pages are about only one specific product. We also excluded those PnOs that return less than 30 pages.
b) We used every PnO name as the query string to Google. We downloaded the first 1,000 pages of each search, and we filtered out those files whose formats are not HTML and plain (i.e. PDF and DOC), and those whose lengths are less than 100 or more than 10,000 characters. The average number of pages returned per PnO is 227.
c) We manually examined and tagged the returned pages to provide the ground truth for the tests. We determined the number of distinct TEs for each PnO, and tagged all the DPs belonging to each TE.

The resulting set of web pages contains about 13,600 pages for 30 person and 30 organization names. We call this set of web page **WebPnO** collection.

In order to compare our results with other reported systems for general web searches, we adopted the **WT10g** data set used in the homepage finding task of TREC-2001 evaluations. It consists of 10-gigabyte subset of the VLC2

collection and is designed to have a relatively high density of inter-server hyperlinks.

The following sub-sections describe our evaluations in finding direct pages and target entity clusters. We also run our algorithm on the WT10g collection on the direct page and home page finding tasks.

7.2. Training of Direct Page Classifier

We used a subset of WebPnO collection to train and test our classifier for direct pages. We randomly selected 250 DPs and 250 IDPs each for both the person and organization categories. We used 4/5 of the pages for training, and the rest for testing. That is, we trained the decision model for finding DPs of persons (or organizations) using 200 positive and 200 negative samples. Each sample is represented using the 17 features as listed in Table 1 and one decision attribute. The learning component is built based on the machine-learning tool C5 (http://www. rulequest.com/see5-info.html).

In order to provide insights into the roles of features and the set of rules extracted for finding DPs, we list some of the decision rules found as follows.

1) Link <= 19 & PER <= 63 & NE > 67 -> Class DP
2) NE > 4 & NE_Word > 0.06883 & NE_Word <= 0.22727 & Word <= 91 -> Class DP
3) ORG > 1 & NE > 14 & NE <= 67 & URL_Depth > 3 -> Class IDP
4) Link > 19 & URL_Depth > 3 -> Class IDP
5) NE <= 4 -> Class IDP

Here, Rule 1 implies that good DPs should have many NEs but relatively few links and person names. Otherwise, they may be index pages or attendee lists. Rule 2 indicates that good DPs tend to be shorter, but contain a high percentage of NEs. In general, they are home pages of persons. Rule 3 and Rule 4 show that IDPs have deeper URL depth. In addition, Rules 5 indicates that those pages that have fewer NEs must be IDPs. These two rules reveal that NEs do play important roles in the classification of pages into DPs and IDPs.

We used the rest of 50 DPs and 50 IDPs from the person or organization categories to test the trained classifiers. We achieved an F_1 measure of about 91.3% (precision 89.7% and recall 92.9%). Our result is comparable to the best results reported for the homepage finding task (92%) in TREC-2001. We are encouraged by this result as we believe that DP detection is a more difficult task than homepage finding. This is because the latter deals only with a relatively simple task, where the decision depends mostly on URL length and whether the URL ends with a keyword or "/".

In order to compare the performance of our system with others on similar tasks, we run further tests using the WT10g collection. We first compared the performance of our decision model with that reported in [8] on the homepage finding task. [8] performed the document analysis by employing decision tree and regression analysis using the feature set based mostly on URL depth, number of in- and out-links, and keywords. They tested on a subset of WT10g collection and reported a F_1 measure of 92%. We conducted similar test using our algorithm based on our original feature set "without tuning", where a larger balanced test set rather than the unbalanced set in [8] was used. We obtained a F_1 measure of about 91%, which is comparable to that reported in [8]. Although the results are not strictly comparable, the results do indicate that our technique is effective, even for the home page finding task which our system is not tuned to perform.

In our second test, we randomly selected about 75 DPs and 75 IDPs for organization from the WT10g collection. We did not conduct a similar test for persons as there are very few (about 10) direct pages about persons. We used 2/3 of these pages for training and the rest for testing. Our classification shows that we could achieve an F_1 measure of about 94%. This is higher than that achieved using on our larger WebPnO collection. The test demonstrates that our WebPnO collection is representative and demanding, and we could obtain better results from the random subset of WT10g collection.

7.3. Web Page Clustering

We now discuss the full experiments on clustering web pages based on our WebPnO collection. We evaluated the performance of our clustering approach according to two aspects. First, we evaluate the quality of seeds. Table 2 gives the detailed performance of detecting seeds. The average number of clusters for persons and organizations is 4.57 and 2.17 respectively. As shown in Table 2, the average ratio of missing clusters and redundant clusters is lower than 10%. This indicates that the seeds are stable and reliable. The quality of seeds is pivotal because it controls the distribution of segmentation. Missing a seed will mean the lost of a cluster and cause some IDPs to be assigned into wrong or unknown set. On the other hand, if there are redundant seeds, IDPs about the same target may be delivered into different clusters. Fortunately, the results indicate that our technique is effective in differentiating between DPs and IDPs, and in removing redundant DPs.

Second, we evaluate the quality of the entire set of clusters. Table 3 lists the performance of assigning IDPs to clusters. The Table shows that we could deliver over 60% of IDPs to the clusters. The rest of less than 40% of pages are placed in the unknown set. We carried out manual check on 300 random IDPs assigned and found that we could achieve a precision of 83.1%, recall of 67.6%, and an overall F_1-measure of 74.6%. As there are no comparable results available on our specific task, it is hard to compare our results in comparison to other reported systems. However, the results reported in [14]

showed that the state-of-the-art clustering methods could achieve a performance of between 59% and 87% in F_1-measure on a range of test corpuses. This places the performance of our system towards the top end of the performance scale. This suggests that our approach is effective and reliable on practical web tasks.

Table 2. The performance of detecting seeds or DPs for distinct TEs.

Seeds	N	N_C	N_I	N_M	N_R	Prec.	Recall
Person	137	127	6	4	2	94.1%	95.5%
Org.	65	61	2	4	4	91.0%	88.4%
Overall	202	188	8	8	6	92.6%	91.8%

Note: N gives the number of samples, and N_C, N_I, N_M and N_R respectively denote the number of correct, incorrect, missing and redundant DPs found.

Table 3. The performance of assigning IDPs.

IDP page	N_{Total}	$N_{Avg.}$	$N_{Unknown}$	*Ratio of delivering*
Person	3,600s	70s(*30)	1,500s	58.3%
Org.	9,800s	220s (*30)	3,200s	67.3%

We conducted another experiment in clustering IDPs without using the NE features. We found that the F_1-measure decreased by nearly 15%. The results again show that the NE features are important for this task.

8. Conclusions

PnO is one of the most common types of queries posed by users when surfing the Internet. The problems with normal web search engines are that they return too many irrelevant pages and are unable to distinguish between different entities having the same name. We develop an effective PnO finder on the web where different target entities of the same name are clustered separately and presented to the users. Our tests on the actual web using the names of 30 persons and 30 organizations show that our method is effective for practical PnO retrieval. Our technique could achieve an F_1 measure of over 92% for finding the cluster seeds, which are direct pages of distinct target entities expressed in the query. Our method could also assign over 60% of indirect pages to the clusters with an F_1 measure of about 75%. Our approach is natural and the users could comprehend our clusters very well and accept it. It thus provides an effective approach for users to summarize information about specific target, and track the activity of entities in which they are interested in.

Further research can also be carried out as follows. First, we need to tune the algorithms in order to improve the effectiveness of DP classifier, and the clustering method. Second, we plan to perform information extraction on the clustering results and present the summary template to the users. This will facilitate user browsing. Third, we plan to extend our techniques to organize and extract information in other domains, such as the research documents. More research on the effective set of features for other domain needs to be carried out.

9. References

[1] Text REtrieval Conference (TREC) Home Page, http://trec.nist.gov/

[2] Ellen M. Voorhees, Donna Harman, Overview of TREC 2001, NIST, TREC 2001, pp1-15

[3] N. Craswell, D Hawking, Overview of the TREC-2002 Web Track, TREC 2002, pp1-16

[4] D. Hawking and N. Craswell, Overview of the TREC-2001 Web Track, TREC 2001, pp61-67

[5] T. Westerveld, et al, Retrieving Web pages using Content, Links, URLs and Anchors, TREC 2001

[6] N. Craswell, et al, Effective Site Finding using Link Anchor Information; SIGIR, 2001

[7] W. Kraaij, et al, The Importance of Prior Probabilities for Entry Page Search, SIGIR2002

[8] W. Xi, E. A. Fox, Machine Learning Approach for Home page Finding Task, TREC 2001, pp686-697

[9] Min Zhang, et al, THU at TREC2002: Novelty, Web and Filtering, TREC 2002, pp29-42

[10] MacFarlane, A. MacFarlane, Pliers at Trec 2002, page 311, TREC 2002, pp311-313

[11] D. Boley, et al, Partitioning-based Clustering for Web Document Categorization, in: Decision Support System 27, 329-341, 1999

[12] O. Zamir and O. Etzioni. Web document clustering: A feasibility demonstration. SIGIR'98, 1998

[13] O. Zamir, O., Etzioni, Grouper: a dynamic clustering interface to Web search results, in Computer Networks, 31(11), pp1361-1374, 1999

[14] M. Steinbach, G. Karypis, V. Kumar, A comparison of document clustering techniques. Text Mining Workshop, KDD, 2000.

[15] G. Salton, M. J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, NY, 1983

[16] K. Yang, Combining Text-, Link-, and Classification-based Retrieval Methods to Enhance Information Discovery on the Web, Ph D. thesis, UNC-CH., 2002

[17] J. Picard, J. Savoy, Using Probabilistic Argumentation Systems to Search and Classify Web Sites, 24(3), pp33-41, Data Engineering Bulletin,2001

[18] A.K. Jain, M. N. Murty, and P.J. Flynn, Data clustering: A review, ACM Computing Surveys, 31(3), pp264-323, 1999