Clustering E-Commerce Search Engines

Qian Peng, Weiyi Meng, Hai He
Department of Computer Science
SUNY at Binghamton
Binghamton, NY 13902, USA
meng@cs.binghamton.edu

Clement Yu
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607, USA
yu@cs.uic.edu

ABSTRACT
In this paper, we sketch a method for clustering e-commerce search engines by the type of products/services they sell. This method utilizes the special features of interface pages of such search engines. We also provide an analysis of different types of ESE interface pages.

Categories & Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – Clustering; H.3.5: Online Information Services – Commercial Services, Web-based Services.

General Terms
Algorithms, Performance, Design, Experimentation.

Keywords
Search engine categorization, document clustering

1. INTRODUCTION
A large number of databases are Web accessible through form-based search interfaces and many of these Web sources are E-commerce Search Engines (ESEs). Providing a unified access to multiple ESEs selling similar products is of great importance in allowing users to search and compare products from multiple sites with ease. To enable a unified access to multiple ESEs, we need to collect ESEs from the Web and cluster them into different groups such that the ESEs in the same group sell the same type of products (i.e., in the same domain) so that a comparison-shopping site can be built on top of the ESEs in each group. In this paper, we provide an overview of a technique to cluster ESEs into different groups according to their product domains.

ESE clustering is related to search engine categorization [1, 2]. The methods described in [1, 2] are query-submission based and they consider document search engines, not ESEs. In contrast, our method uses only the Web pages that contain the search forms of the considered ESEs to perform clustering (no queries are submitted to these ESEs). ESE clustering is also related to document/Web page clustering. However, existing approaches for the latter do not utilize the special features (such as search forms) that only ESE search interfaces have.

In this paper, we sketch a new method for clustering ESEs and the method utilizes the special features of ESE interface pages. We also provide an analysis of different types of ESE interface pages.

2. DIFFERENT TYPES OF INTERFACES
While most ESEs are dedicated to one category of products or services, a significant fraction of ESEs cover multiple categories of products. Based on our analysis of 270 ESE sites, we identified the following 6 types of ESE interfaces.

1. *Devoted Type.* In such an interface, only one ESE search form exists and it searches only one category of products. About 83% of the ESE sites we studied belong to this type.
2. *Divided Type.* Such an interface can search multiple categories of products but it also has separate child ESE search interfaces for different category of products. In this study, the child ESEs of the same site are treated as separate ESE interfaces for clustering purpose.
3. *Co-existing Type.* In this case, multiple ESEs coexist on a single Web page. For example, the airfare.com interface page has 4 search forms that can search flight, car reservation, travel package and hotels separately. In our approach, search forms on the same interface page are first separated and then treated as separate ESE interfaces during clustering.
4. *Merged Type.* In this case, a single search form can search multiple categories of products and the search fields of different types of products co-exist in the search form. Moreover, only one submission button is available. For example, *alldirect.com* sells books, music and movies, and its search form contains book attributes, music attributes and movie attribute. In our approach, such attributes are separated into different *logical interfaces* such that each logical interface covers only one category of products.
5. *Shared Type.* In this case, the ESE site has multiple child pages for different types of products but these pages share the same search form (i.e., the same search form is available on all child interface pages) that can search multiple categories of products. Note that each child page contains more information about a specific category of products. In our approach, we treat each such child page as a separate ESE for clustering purpose.
6. *Multi-page Type*. An ESE of this type requires a user to submit a sequence of pages to complete a query. This occurs, for example, when a user needs to obtain an insurance quote. As our clustering approach does not submit queries, for an ESE of the multi-page type, only the first page of the ESE is used for clustering.

3. ESE INTERFACE FEATURES
Our method utilizes the following features on ESE interfaces.

1. *The number of links/images*. Our observation and analysis of the 270 ESE sites indicate that this feature is very useful for differentiating ESEs that sell tangible products from those selling intangible products. Tangible products are those that have a physical presence such as books and music CDs while intangible products have no physical presence such as insurance policy. The interfaces of ESEs that sell tangible products usually have more images/links than those that sell intangible products.
2. *Price values*. For online shopping customers, price information is very important for their purchase decisions.

WWW 2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

Therefore, to attract consumers, bargaining products or top-selling products are frequently advertised with prices on the interface pages of ESEs that sell tangible products. Products of the same category usually have similar prices. As such, the range of price values can be useful to differentiate different types of products. To facilitate meaningful comparison, we convert each actual price to a *representative price* that will then be used for comparison. Each representative price represents all prices within a certain price range.

3. *Form terms*. A typical ESE search form has an interface schema, which contains some labels (descriptive text) and a number of form control elements such as text boxes, selection lists, radio buttons or checkboxes to allow users to specify complex queries rather than just keywords. The labels usually specify the real meaning of its associated elements. These attributes and their values can strongly indicate the contents of the ESE. Therefore, they are very useful for ESE clustering. In our approach, form terms include both labels and values associated with form elements.
4. *Regular terms.* These are the terms that appear in an ESE Web page but are not price terms or form terms.

To simplify notations in the following sections, for a given ESE interface, we use ***N*** to represent the total number of images and hyperlinks, ***P*** to represent the vector of price terms with weights, ***FT*** to represent the vector of weighted form terms and ***RT*** to represent the vector of weighted regular terms. The terms in each category (**P**, **FT**, or **RT)** are organized into a term vector with weights computed using the standard *tf***idf* formula [3]. For a given term, formula is an increasing function of the term's frequency in the term category on a page and a decreasing function of the term's document frequency (i.e., the number of ESE interfaces that have the term).

4. CLUSTERING ALGORITHM

Our clustering algorithm consists of two phases. In the first phase, all ESEs are clustered into two groups, one for selling tangible products and the other for selling intangible products. This is done by comparing the number of images/links (N) in an ESE with a threshold *T*. If $N \geq T$, the ESE is placed into the tangible group; otherwise, it is placed into the intangible group. In the second phase, ESEs in each group are further clustered using other features (i.e., P, FT and RT for the tangible group, and FT and RT for the intangible group). There is no fundamental difference between clustering the ESEs in the tangible group and those in the intangible group except that the latter does not have the price vector P. In both cases, the basic idea is to use features to cluster similar ESEs together based on their feature similarity. The similarity between two ESEs is defined to be the weighted sum of the similarities between the price vectors (for the tangible group only), the form term vectors and the regular term vectors of the two ESEs. In our current implementation, the similarity between two vectors is computed using the Cosine similarity function [3]. Below, we describe the clustering algorithm for the second phase in more detail. This phase itself consists of two steps.

Preliminary Clustering step: In this step, a simple single-pass clustering algorithm is applied to obtain a preliminary clustering of the ESEs in each group. The basic idea of this algorithm is as follows. First, starting from an arbitrary order of all the input ESEs, take the first ESE from the list and use it to form a cluster. Next, for each of the remaining ESEs, say A, compute its similarity with each existing cluster. Let C be the cluster that has the maximum similarity with A. If sim(A, C) is greater than a threshold, which is to be determined experimentally, then add A to C; otherwise, form a new cluster based on A. Function sim(A, C) is defined to be the average of the similarities between A and all ESEs in C. This single-pass algorithm is efficient as it considers each input ESE once. However, it is order sensitive and t is possible that an ESE is not included in the most suitable cluster due to the order it is considered.

Refining step: This step tries to remedy the weakness of the previous step. The idea is to move potentially unfitting ESEs from their current clusters to more suitable ones. This is carried out as follows. First, we compute the average similarity AS of each cluster C, which is the average of the similarities between all ESE pairs in cluster C. Second, identify every ESE A in C whose similarity(A, C) is less than AS. These ESEs are considered to be potentially unfitting. Third, for each ESE A obtained in the second step, we compute its similarities with all current clusters (including the cluster that contains A) and then move it to the cluster with the highest similarity. The above refining process is repeated until there is no improvement (increase) on the sum of the similarities of all clusters.

5. EXPERIMENTS

270 ESE interface pages from 8 categories in Yahoo directory *"Business and Economy → Shopping and services"* are obtained, 7 categories contain ESEs selling tangible product while one selling intangible products. After child/logical ESE interfaces are identified, 294 ESEs are obtained. Before evaluation, all ESEs are manually grouped based on what products they sell. Clusters obtained by the manual clustering are deemed correct and are used as the basis to evaluate our clustering algorithm. The following criteria are used to measure the performance of clustering: *precision*: the ratio of the number of ESEs that are correctly clustered over the number of all ESEs; *recall*: for a given cluster, recall is the ratio of the number of ESEs that are correctly clustered over the number of ESEs that should have been clustered, and the overall recall for all clusters is the average of the recalls for all clusters weighted by the size of each cluster.

As discussed in Section 4, our clustering algorithm has two phases. In Phase 1, ESEs are clustered into two big groups: the *tangible group* and the *intangible group* based on the number of images/links on each ESE interface. The recall and prevision of the Phase 1 clustering when *T* is 30 are all above 95%. For the Phase 2 clustering, the performance is as follows: precision is ***94%*** and recall is ***93%***. Our experiments indicate that form terms are critically important for accurately clustering ESEs. When form terms are not used, both recall and precision decrease by about 35%.

6. ACKNOWLEDGEMENTS

This work is supported in part by the grants IIS-0208574, IIS-0208434 from the National Science Foundation.

7. REFERENCES

[1] P. Ipeirotis, L. Gravano, M. Sahami. *Probe, Count and Classify: Categorizing Hidden-web Databases*. ACM SIGMOD Conference, May 2001, Santa Barbara, California.

[2] W. Meng, W. Wang, H. Sun, C. Yu. *Concept Hierarchy Based Text Database Categorization*. Journal of Knowledge and Information Systems, 4(2), 2002, 132-150.

[3] G. Salton and M. McGill. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.