Search Result Clustering Using Label Language Model

Yeha Lee Seung-Hoon Na Jong-Hyeok Lee
Div. of Electrical and Computer Engineering
Pohang University of Science and Technology (POSTECH)
Advanced Information Technology Research Center (AITrc)
San 31, Hyoja-Dong, Pohang, Republic of Korea, 790-784
{sion,nsh1979,jhlee}@postech.ac.kr

Abstract

Search results clustering helps users to browse the search results and locate what they are looking for. In the search result clustering, the label selection which annotates a meaningful phrase for each cluster becomes the most fundamental issue. In this paper, we present a new method of using the language modeling approach over Dmoz for label selection, namely label language model. Experimental results show that our method is helpful to obtain meaningful clustering labels of search results.

1 Introduction

Most contemporary search engines generate a long flat list in response to a user query. This result can be ranked by using criteria such as PageRank (Brin and Page, 1998) or relevancy to the query. However, this long flat list is uncomfortable to users, since it forces users to examine each page one by one, and to spend significant time and effort for finding the really relevant information. Most users only look into top 10 web pages in the list (Kummamuru et al., 2004). Thus many other relevant information can be missed out as a result. Clustering method is proposed in order to remedy the problem. Instead of the flat list, it groups the search results to clusters, and annotates a label with a representative words or phrases to each cluster. Then, these labeled clusters of search results are presented to users. Users can benefit from labeled clusters because the size of information presented is significantly reduced.

Search result clustering has several specific requirements that may not be required by other cluster algorithms. First, search result clustering should allow fast clustering and fast generation of a label on the fly, since it is an online process. This requirement can be met by adopting "snippets" [1] rather than entire documents of a search result set. Second, labels annotated for clusters should be meaningful to users because they are presented to users as a general view of results. For this reason, recent search result clustering researches focus on selecting meaningful labels. This differs from general clustering which focuses on the similarity of documents. In Zamir and Etzioni (Zamir and Etzioni, 1998), a few other key requirements of search result clustering are presented.

In this paper, we present a language modeling approach with Dmoz for search result clustering. Dmoz [2] is an Open Directory Project, and contains manually tagged categories for web-sites. Since these categories are built by human, they provide a good basis to build labels for clusters. We can view the problem of label selection for clusters as a problem of label generation by Dmoz.

We define a language model for each Dmoz-category and select labels for clusters according to the probability that this language model would generate candidate labels.

Thus, our method can select more meaningful labels for clusters because we use labels generated by human-tagged categories of Dmoz. The selected la-

[1] The term "snippet" is used here to denote fragment of a Web page returned by certain search engines

[2] Open Directory Project, http://www.dmoz.com/

bels enable users to quickly identify the desired information.

The paper is organized as follows. The next section introduces related works. In Section 3, we formulate the problem and show the detail of our approach. The experiment results and evaluations are presented in Section 4. Finally, we conclude the paper and discuss future works in Section 5.

2 RELATED WORKS

Many approaches have been suggested for organizing search results to improve browsing effectiveness. Previous researches such as scatter/Gather (Hearst and Pedersen, 1996) and Leuski (Leuski and Allan, 2000), Leouski (Leouski and Croft, 1996), cluster documents using document-similarity, and generate representative terms or phrases as labels. However, these labels are often not meaningful, which complicates user relevance judgment. They are also slow in generating clusters and labels because they use entire document contents in the process. Thus it is difficult to apply these approaches to search engine applications.

Due to the problems mentioned above, research in search result clustering has focused on choosing meaningful labels which is not usually addressed in general document clustering. Zeng et al. presented salient phrase ranking problem for label selection, which ranks labels scored by a combination of some properties of labels and documents (Zeng et al., 2004). Kummamuru regarded label selection as a problem making a taxonomy of the search result, and proposed a label selecting criterion based on taxonomy likelihood (Kummamuru et al., 2004). Zamir presented a Suffix Tree Clustering (STC) which identifies sets of documents that share common phrases, and clusters according to these phrases (Zamir and Etzioni, 1998). Maarek et al. and Osinski presented a singular value decomposition of the term-document matrix for search result clustering (Maarek et al., 2000), (Osinski and Weiss, 2004). The problem of these methods is that SVD is extremely time-consuming when applied to a large number of snippets. Ferragina proposed a method for generating hierarchical labels by which entire search results are hierarchically clustered (Ferragina and Gulli, 2005). This method produces a hierarchy of labeled clusters by constructing a sequence of labeled and weighted bipartite graphs representing the individual snippets on one side and a set of labeled clusters on the other side.

3 LABEL LANGUAGE MODEL

The main purpose of Label Language Model(LLM) is to generate meaningful labels on-the-fly from search results, specifically snippets, for web-users. The generated labels provide a view of the search result to users, and allow the users to navigate through them for their search needs.

Our algorithm is composed of the four phases:

1. Search result fetching
2. Candidate Labels Generation
3. Label Score Calculation
4. Post-processing

Search result fetching. LLM operates as a meta-search engine on top of established search engines. Our engine first retrieves results from dedicated search engines in response to user queries. The search results are parsed through HTML parser, and snippets are obtained as a result. We assume that these snippets contain enough information to provide user-relevance judgment. Hence, we can generate meaningful labels using only those snippets rather than the entire document contents of the search result set.

Candidate Labels Generation. Candidate labels are generated using the snippets obtained by search result fetching. Snippets are processed by Porter's algorithm for stemming and stopword removing, then every n-grams becomes a candidate label. Each candidate label is tagged with a score calculated by the Label Language Model. Finally, top N candidate labels with highest scores are displayed to users as labels for clusters of search result.

Label Score Calculation. Our model utilizes Dmoz to select meaningful labels. Dmoz is the largest, most comprehensive human-edited directory of the Web and classifies more than 3,500,000 sites in more than 460,000 categories. It is used for ranking and retrieval by many search engines, such as

Google (Rerragina and Gulli, 2005).

Language model ranks documents according to the probability that the language model of each document would generate the user query.

Dmoz is a human-edited directory, which contains meaningful categories. We can use the probability that categories of Dmoz would generate candidate labels as criteria to rank labels.

In our approach, the user query and the document correspond to the candidate label and the Dmoz's category, respectively. We can obtain the probability that LLM of each category would generate a label by language model. We assume that the probability of certain candidate label being generated can be estimated by the maximum value of the probability that LLM of each category would generate the candidate label.

Let $label_i$ be i^{th} label, w_{ij} be j^{th} word of $label_i$, and C_k be k^{th} category of Dmoz, respectively. If we assume that the labels are drawn independently from the distribution, then we can express the probability that Dmoz generates labels as follows:

$$p(label_i|Dmoz) = \max_k p(label_i|C_k) \quad (1)$$

$$p(label_i|C_k) = \prod p(w_{ij}|C_k) \quad (2)$$

We use two smoothing methods, Jelinek-Mercer smoothing and Dirichlet Priors smoothing (Zhai and Lafferty, 2001), in order to handle unseen words. The score of $label_i$ is calculated as follows:

$$S_i = \max_k \sum_j \log\left(1 + \frac{\lambda p(w_{ij}|C_k)}{(1-\lambda)p(w_{ij}|C_{all})}\right) \quad (3)$$

$$S_i = \max_k \sum_j \log \frac{\#(w_{ij}^k) + \mu p(w_{ij}|C_{all})}{\#(C_k) + \mu} \quad (4)$$

To solve the equation, $p(w_{ij}|C_k)$ and $p(w_{ij}|C_{all})$ should be estimated. Let $\#(C_k)$ and $\#(C_{all})$[3] be the number of words in k^{th} category and the number of words in Dmoz. Further, let $\#(w_{ij}^k)$ and $\#(w_{ij}^{all})$ be the number of word, w_{ij}, in k^{th} category and the number of word, w_{ij}, in Dmoz. Then $p(w_{ij}|C_k)$ is estimated as $\frac{\#(w_{ij}^k)}{\#(C_k)}$, and $p(w_{ij}|C_{all})$ as $\frac{\#(w_{ij}^{all})}{\#(C_{all})}$.

[3] C_{all} denotes all categories of Dmoz

In Candidate Labels Generation phase, all candidate labels are scored. After post-processing, candidate labels are shown in a descending order.

Post-processing. In post processing phase, labels are refined through several rules. First, labels composed of only query words are removed because they do not provide better clues for users. Second, labels that are contained in another label are removed. Since every possible n gram is eligible for candidate labels, multiple labels that differ only at the either ends, i.e., one label contained in another, can be assigned a high score. In such cases, longer labels are more specific and meaningful than shorter ones, therefore shorter ones are removed. Users can benefit from a more specific and meaningful label that clarifies what a cluster contains. Finally, Top N Labels with highest scores produced by post processing are presented to users.

4 EXPERIMENTS

We conducted several experiments with varying smoothing parameter values, λ, μ. We investigated the influence of the smoothing parameter on the label selection procedure.

4.1 Experiment Setup

Despite heavy researches on search result clustering, a standard test-set or evaluation measurement does not exist. This paper adopts the methodology of (Zeng et al., 2004) in order to evaluate the expressiveness of selected label and LLM

4.1.1 Test Data Set

We obtained Google's search results that correspond to fifty queries. The fifty queries are comprised of top 25 queries to Google and 25 from (Zeng et al., 2004). For each query of the fifty, 200 snippets from Google are obtained. Table 1 summarizes the query used in our experiment.

Search results obtained from Google are parsed to remove html-tag and stopword, and stemming is applied to obtain the snippets. Every n-gram of the snippets, where $n \leq 3$, becomes candidate labels. Labels that do not occur more than 3 times are removed from candidate set in order to reduce noise.

Type	Queries
2005 Google Top query	Myspace, Ares, Baidu, orkut, iTumes, Sky News, World of Warcraft, Green Day, Leonardo da Vinci, Janet Jackson, Hurricane Katrina, tsunami, xbox 360, Brad Pitt, Michael Jackson, American Idol, Britney Spears, Angelina Jolie, Harry Potter, ipod, digital camera, psp, laptop, computer desk
(Zeng et al., 2004) query	jaguar, apple, saturn, jobs, jordan, tiger, trec, ups, quotes, matrix, susan dumais, clinton, iraq, dell, disney, world war 2, ford, health, yellow pages, maps, flower, music, chat, games, radio, jokes, graphic design, resume, time zones, travel

Table 1: Queries used in experiment

4.1.2 Answer Label Set for Evaluation

In order to evaluate LLM, we manually created labels for each query which are desired as outputs of our test, and we refer them as answer labels. There might be a case where an answer label and label selected by our model are semantically equivalent but lexically different; for example, car and automobile. To mitigate the problem, we used Wordnet to handle two different words with the same semantic. We explain the use of Wordnet further in section 4.1.3.

4.1.3 Evaluation Measure & Method

We used precision at top N labels to evaluate the model. Precision at top N is defined as $P@N = \frac{M@N}{N}$, where is $M@N$ is the number of relevant labels among the top N generated labels to the answer set. As explained in section 4.1.2, the labels generated by our model might not be equal to answer labels even when they have the same semantic meaning. It might be very time consuming for a human to manually compare the two label set where one set can vary due to the varying smoothing parameter if semantic meaning also has to be considered.

We used WordNet's synonyms and hypernyms relationships in order to mitigate the problem addressed above. We regard a test label to be equal to an answer label when WordNet's synonyms or hypernyms relationship allows them. Only the first listed sense in Wordnet is used to prevent overgeneration.

We evaluated the overall effectiveness of LLM with $P@N$ and the effect of smoothing parameter on $P@N$.

4.2 Experimental Result

We used $P@5$, $P@10$ and $P@20$ to evaluate the effectiveness of our model because most users disregard snippets beyond 20.

First, for each query, we obtained each label's MAP[4] for two smoothing methods. Figures 1 and 2 depicts MAP of Jelinke-Mercer smoothing and Dirichlet Priors smoothing.

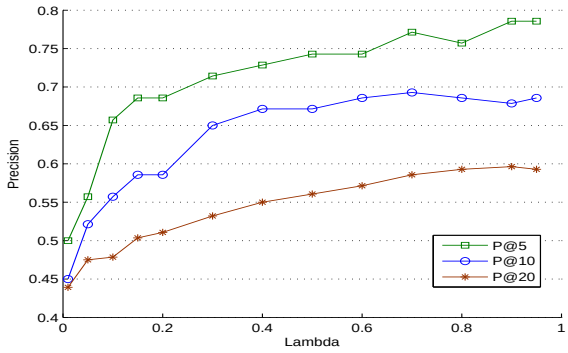

Figure 1: Jelinek-Mercer Smoothing

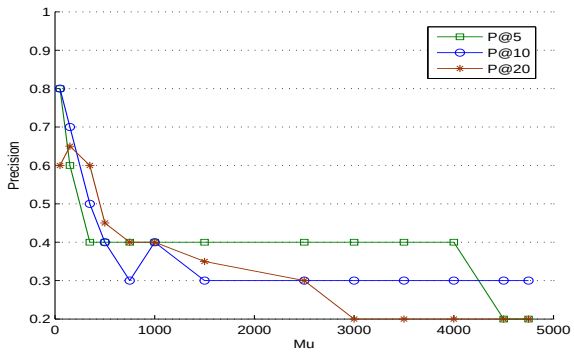

Figure 2: Dirichlet priors Smoothing

In figures 1 and 2, X-axis denotes smoothing parameter, and Y-axis denotes MAP. The figures show that the smaller the value of the smoothing is , the

[4]Mean Average Precision

higher the precision is. This indicates that a better label is selected when the probability that a specific category would generate the label is high. In our test result, when using Dirichlet smoothing, the precision of top 5 and 10 labels are 82% and 80%, thus users can benefit in browsing from our model using 5 or 10 labels. However, the precision rapidly drops to 60% at $P@20$. The low precision at $P@20$ shows the vulnerability of our model, indicating that our model needs a refinement.

Figure 3 shows individual precisions of labels for randomly selected five queries. The labels were generated by using Dirichlet priors smoothing.

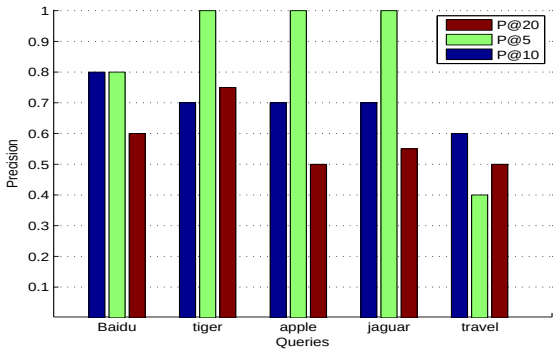

Figure 3: Using Dirichlet priors Smoothing

As shown in figure 1 and 2, the general order of result precisions is as follows: $P@20 \leq P@10 \leq P@5$. However, figure 3 shows that the precision for query "travel" is the lowest at $P@5$. This result indicates that words that appear many times in a specific category of Dmoz might have higher probability regardless of snippet's contents.

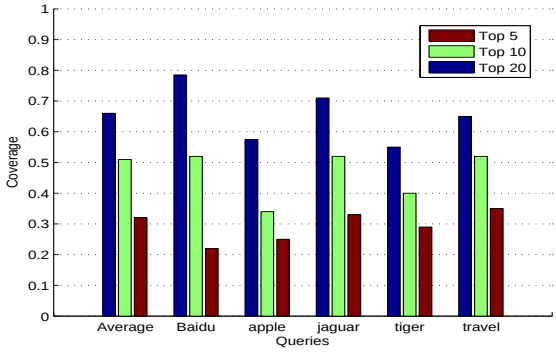

Figure 4: Coverage

Figure 4 shows the average coverage of labels generated by our model. The coverage of the labels is about 0.32%, 0.51% and 0.66% at top 5, 10 and 20 labels respectively. This means that the labels allow browsing over only 60% of the entire search results. The lack of coverage is another pitfall of our model, and further refining is needed.

Finally, in Table 2, we list top 10 labels for five queries.

5 CONCLUSION & FUTURE WORKS

We proposed a LLM for label selection of search results, and analyzed the smoothing parameter's effect on the label selection. Experimental results showed that LLM can pick up meaningful labels, and aid users in browsing web search results. Experimental results also validated our assumption that the high probability that Dmoz categories generate a label indicates meaningful labels. Further research directions remain as future works.

Our model is sensitive to Dmoz because we use the language model based on Dmoz. Our model may result in poor performance for labels that are not represented or over-represented in Dmoz. Therefore, it is meaningful to study how sensitive to Dmoz the performance of the LLM is, and how to mitigate sensitivity. We used Google's search results as an input to our system. However, multiple engines offer a better coverage of the web because of the low overlap of current search engines (Bharat and Broder, 1998). Further work can utilize multiple engines to generate input to our system. In our test, snippet's title and content were assigned the same weight, and titles and descriptions of Dmoz's category were also assigned the same weight. Future work might benefit from varying the weights to them. We did not utilize the information buried in the documents, such as $tf \cdot idf$, but used only knowledge provided by the external system, Dmoz. We believe that this also affected LLM's poor performance on over-represented terms. Future work will benefit from incorporating the information derivable from the documents.

6 ACKNOWLEDGMENTS

This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the Advanced Information Technology Research Center

Queries	Labels
Baidu	language search set, Chinese search engine, search engine company, Baidu.com, MP3 Search, Baidu engine, Japanese Search Engine, IPO, search market, Mobile
apple	Mac OS X, iPod, Apple Macintosh, Apple products, language character set, Music Store, Apple develops, Apple Support, information, San Francisco
jaguar	Mac OS X, Jaguar Cars, Land Rover, Jaguar XJ, Jaguar XK, largest cat, Leopard, Photos tagged jaguar, Jaguar dealer, Jaguar Clubs
tiger	Mac OS X, Tiger Woods, Tiger Cats, Detroit Tigers, Security tool, Parts PC Components, Paper Tiger, Adventure Tour, National Zoo, Tiger Beat
travel	Car Rental, airline tickets, discount hotels, Plan trip, Airfares, package holidays, Visa, Travel Cheap, Destination guides, Travel news

Table 2: Queries used in experiment

(AITrc), also in part by the BK 21 Project and MIC & IITA through IT Leading R&D Support Project in 2007.

References

P. Rerragina and A. Gulli. 2005. A personalized search engine based on web-snippet hierarchical clustering. In *Special Interest Tracks and Poster Proceedings of WWW-05, International Conference on the World Wide Web*, 801-810

H. Zeng, Q. He, Z. Chen, W. Ma and J. Ma 2004. Learning to cluster web search results. In *Proceedings of the 27th ACM SIGIR Conference on Research and Development of Information Retrieval*

M. A. Hearst and J. O. Pedersen. 1996. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of 19th ACM SIGIR Conference on Research and Development in Information Retrieval*, 76-84

K. Kummamuru, R. Lotlikar, S. Roy, K. Signal and R. Krishnapuram 2004. A hierarchical monothetic document clustering algorithm for summarization browsing search results. In *Proceedings of 13th International Conference on World Wide Web*, 658-665

A. Leuski and J. Allan. 2000. Improving Interactive Retrieval by Combining Ranked List and Clustering. In *Proceedings of RIAOI, College de France*, 665-681

A. V. Leouski and W. B. Croft. 1996. An Evaluation of Techniques for Clustering Search Results. In *Technical Report IR-76*, Department of Computer Science, University of Massachusetts, Amherst

O. Zamir and O. Etzioni. 1998. Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the 21th ACM SIGIR Conference on Research and Development of Information Retrieval*, 46-54

Y. Maarek, R. Fagin, I. Ben-Shaul and D. Pelleg. 2000. Ephemeral document clustering for Web applications. *Technical Report RJ 10186, IBM, San Jose, US*

S. Osinski and D. Weiss. 2004. Conceptual clustering using Lingo algorithm: Evaluation on Open Directory Project data In *Proceedings of IIPWM-04, 5th Conference on Intelligent Information Processing and Web Mining*, 369-377

P. Ferragina and A. Gulli. 2005. A personalized search engine based on Web-snippet hierarchical clustering. In *Special Interest Tracks and Poster Proceedings of WWW-05, International conference on the World Wide Web*, 801-810

S. Brin and L. Page 1998. The anatomy of a large-scale hypertextual(Web) Search Engine. In *Proceedings of the 7th International Conference on World Wide Web*, 107-117

C. Zhai and J. Lafferty 2001. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th ACM SIGIR Conference on Research and Development of Information Retrieval*, 334-342

K. Bharat and A. Broder. 1998. A technique for measuring the relative size and overlap of public web search engines. In *Proceedings of the 7th International Conference on World Wide Web*