Learning to Generate Labels for Organizing Search Results from a Domain-Specified Corpus

Jing Zhao, Jing He
Dept. of Computer Science, School of EECS, Peking Univ., Beijing, 100871, P.R.China
jingzhao@pku.edu.cn, hj@net.pku.edu.cn

Abstract

Organizing Web search results into labeled categories is a difficult but very useful task. The idea is to group the many results that each user query generates into well-labeled categories, so that users can find it much easier to browse these results. In the past, clustering-based methods have been applied to solve the search-result organization problem, but it has been difficult to extract the human-readable descriptions for these clusters. An alternative solution to this problem is to generate a series of labels from search results firstly, and then assign documents to relevant labels to form labeled categories. In this approach, a major task is how to generate the labels for the documents. In this paper, we propose a novel label generation method: Firstly, we extract some phrases as candidates of labels based on the search results, and adopt a binary classifier as our learning model to classify these label candidates into useful or meaningless label category. Then, the candidates in the useful label category form the final results. As our method is applied on the search results which are retrieved from a domain-specified corpus instead of general corpus, there're some special features of the labels for classification. Experimental results show that the accuracy of our system is nearly 10% higher than using the mutual information criterion, which is an unsupervised method for solving this problem, to do the label selection.

1. Introduction

As Web search increases its popularity, many successful commercial search engines, such as Google, Yahoo, and MSN serve the needs of many users everyday. With these search engines, users can get a ranked list of Web pages after sending their queries. Ambiguous queries can also generate many different types of results, overloading for users at the same time. In fact, only a small fraction of the search results is useful from the users' perspective. Some results are relevant for one user, and others are relevant for other users. Thus, a user may spend much effort in browsing for the correct result.

Many researches have been done on this issue. One solution is to organize search results into some labeled groups, and give description (also called *labels*) to each group. Users discriminate the labels, and pick the group which has the favored label to look into. A commercial online system available is vivisimo[10]. This liberates users from going through the whole list. Traditional clustering algorithms can construct the labeled group by clustering all the result pages into a hierarchical structure. However, it is difficult to provide highly readable labels to these groups.

An alternative method is to first generate some phrases or sentences as labels based on search results, and then assign documents to related labels to form labeled groups. These label-based approaches supply readable labels of groups for ease browsing compared to clustering method. The critical step of these approaches is how to generate these labels. In this work, we focus on this issue.

Our key contribution is on label generation from search result documents. First, we extract a list of label candidates from search results, and then classify them into two categories: one for useful labels, and the other for meaningless labels by adopting a binary classifier. Finally, useful labels are selected to be present to the users. An important component is to select features to describe label candidates for the classification, just as in other classification tasks. Thus, we develop a collection of features for label candidates. We also adopt two classical feature selection measures, information gain and CHI-square, to evaluate the quality of these features.

In the past, several researches as [4] have been conducted. The approach in [4] adopts regression model to perform supervised learning for label selection. On the contrary, ours uses classification instead. As such, we have a much broader range of choices in classification methods. Also, their target is to organize search results from a general search engine. In our work, we concentrate on a domain-specific corpus that makes our features effective.

This paper is organized as follows. Related works are introduced in section 2. System description is given in section 3. In section 4, we illustrate the experiment results of our system. Finally, we conclude the paper and propose some future works in section 5.

2. Related Works

Many studies for organizing search results have been done, and can be categorized into three classes:

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00 © 2006 IEEE

IEEE COMPUTER SOCIETY

1. Traditional clustering method based on content.
2. Clustering using other information such as link analysis.
3. Label based method.

Some algorithms such as [2][3][12][13] belong to the first one class. They apply traditional clustering methods, and mostly use titles and snippets instead of full pages as content for the consideration of efficiency. The common problem of these methods is difficult to give human-readable and reasonable label for each cluster.

Other algorithms such as [5] notice some special characteristics of web pages that not exist in common documents - the hyperlinks, and introduce link analysis to clustering search results. System in [5] organizes search results by traditional clustering method such as k-means, but constructs similarity function based on comparison of links (may include in and out links) between pages.

Label based methods first extract some labels which can be phrases, terms or sentences, then assign documents to related labels to form labeled groups. [1][4][6][7][14] are classical algorithms of this class. Our system also belongs to this one.

3. System Description

As figure 1 shows, our system contains three main components: 1)label candidates extractor is used to generate label candidates based on search results and query term; 2)representation of label candidates by some features is done by feature value calculator; 3)label selector implemented by a binary classifier selects final labels from label candidates.

We give three sub sections below to describe label candidates extraction, features of label candidates and classifier as learning model separately.

3.1. Label Candidates Extraction

Our system first extracts some phrases based on search results using the method in [9]. Then it employs mutual information between query term and phrase as criterion to select some of these phrases as our label candidates.

3.2. Features of Label Candidates

In our application, we discover that there are three main classes of characteristics of label candidates: 1)some indicators about co-occurrence of candidate and query term in domain-specified corpus; 2)general characteristics of label candidates independent to domain-specified corpus; 3)the difference between dependence of query term and label candidate in global corpus and that in domain-specified corpus.

We use a news search engine as our IR system, considering the news corpus as a domain-specified corpus, and propose five features for representation of label candidate: distance, mutual information and likelihood ratio belong to the first class, part of speech of label candidate belongs to the second class, and difference mutual information between global corpus and in domain-specified corpus belongs to the third class.

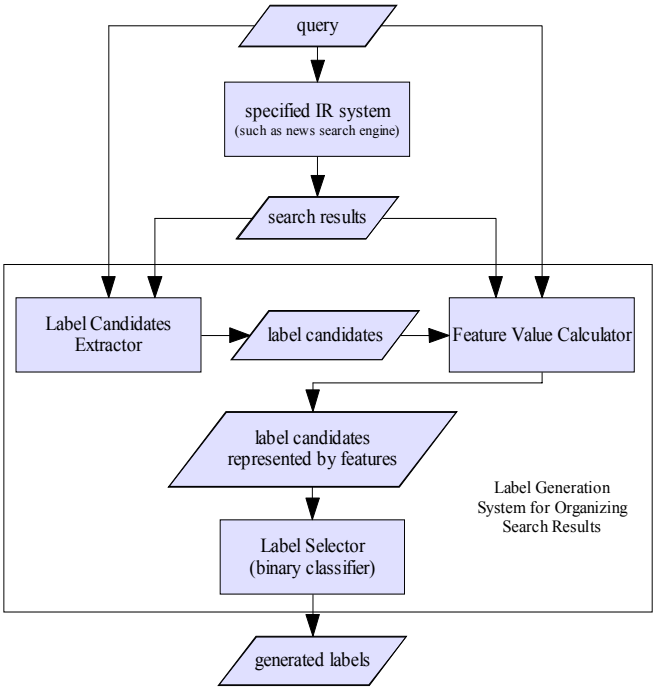

Figure 1: Architecture of Label Generation System

We adopt two classical measures for feature selection, information gain and CHI-square to examine the effect of each feature, as this two measures are proved to be most effective as feature selection measure in the study in[18].
Details for each feature are given in the following:

Distance between Query Term and Label Candidate

We define the distance between query term and label candidate as the number of terms between them when they appear together in one field (where a field can be a part of document, such as title, body, description, anchor text and so on).If label is behind the query term, the distance is negative, otherwise, it is positive.

Distance information has been studied in works [15] and [16]. Their works focus on extracting collocation in text, both of which employ variance of distance as indicator. In our application, we get distances between a given query term and label candidate in the domain-specified corpus, and calculate both mean value and variance of distances.

Proceedings of the 2006 IEEE/WIC/ACM International Conference
on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)
0-7695-2747-7/06 $20.00 © 2006 IEEE
IEEE COMPUTER SOCIETY

Mutual Information between Label Candidate and Query Term

Mutual information is used as an indicator of local dependence between label candidate and query term. We give the definition as:

$$MI = \log_2 \frac{p(query, label)}{p(query) * p(label)}$$

$p(query, label)$ indicates the possibility of query term and label candidate occurring together in one field. Definition of $p(query)$ and $p(label)$ is similar.

Mutual information is generally used in finding the collocation of terms for speech recognition and machine translation in NLP and co-occurrence of terms for query expansion in IR. However, it can not process the sparse data well, so we introduce likelihood ratio as another feature to complement the limitation.

Likelihood Ratio between Query Term and Label Candidate

Likelihood ratio is used in [8] for text analysis. In our application, it is adopted to indicate the dependence between query term and label candidate.

We use w_1 and w_2 to indicate query term and label candidate. N is the total number of fields (as before, a field can be a part of document, such as title, body, description, anchor text and so on). $c(w_1)$ (or $c(w_2)$) is the number of times that w_1 (or w_2) appears in one field. $c(w_1 w_2)$ is the number of times that w_1 and w_2 appear together in one field. As [11], there are two hypotheses as follow:

Hypothesis 1: $P(w_2 \mid w_1) = p = P(w_2 \mid \neg w_1)$

Hypothesis 2: $P(w_2 \mid w_1) = p_1 \neq p_2 = P(w_2 \mid \neg w_1)$

Hypothesis 1 implies independence between w_1 and w_2, while hypothesis 2 shows their dependence which is the pattern we are interested in. Under hypothesis 1, $p = \frac{c(w_2)}{N}$, while under hypothesis 2, $p_1 = \frac{c(w_1 w_2)}{c(w_1)}$ and $p_2 = \frac{c(w_2) - c(w_1 w_2)}{N - c(w_1)}$.

Assume that p, p_1 and p_2 is under binominal distribution: $b(k; n, x) = \binom{n}{k} x^k (1-x)^{(n-k)}$

Then the likelihood is defined as:

$$L(H_1) = b(c(w_1 w_2); c(w_1), p) * b(c(w_2) - c(w_1 w_2); N - c(w_1), p)$$

$$L(H_2) = b(c(w_1 w_2); c(w_1), p_1) * b(c(w_2) - c(w_1 w_2); N - c(w_1), p_2)$$

Then we can calculate the likelihood ratio as:

$$\log \lambda = \log \frac{L(H_1)}{L(H_2)}$$

This value indicates the dependence between query term and label candidate. As likelihood ratio is an approximate form of Chi-square test, it works well in processing sparse data[11]. So likelihood ratio and mutual information have distinct conditions to fit in.

Part of Speech of Label Candidate

Intuitively, noun phrase is always more likely to be a good label. So we consider part of speech(POS) of label candidate as a feature. Part of speech is a general feature which is independent of the domain-specified corpus.

We first use a lexical analytical system to annotate POS of candidates. Then we do some emendation of the annotation manually (as lexical analytical systems cannot do annotation well for some phrase etc).

Difference between Mutual Information in Global and domain-specified corpus

As the dependence between query term and label candidate may be different in global corpus (general corpus) and in domain-specified corpus (local corpus), we assume that the difference may reveal some characteristics of the label candidate. So we define the difference as following:

$$diff\_MI = \frac{global\_MI}{local\_MI}$$

At last, five features are selected for latter classification:

1. distance between query term and label candidate;
2. mutual information between label candidate and query term;
3. likelihood ratio between query term and label candidate;
4. part of speech of label candidate;
5. difference between mutual information in global and in domain-specified corpus

3.3. Learning Model

A binary classifier is used as our learning model. Each new-coming label candidate represented as a vector x = (distance, MI, likelihood ratio, part of speech, MI_Diff) is classified into useful label category or meaningless label category to be decided whether it is a valid label or not.

In the training data, each label candidate for a given query term is judged manually, and be given an annotation of 1 or 0 to show whether it is a good label or not. The binary classifier is trained by these data.

IEEE COMPUTER SOCIETY

We use four different classifiers including support vector machine, decision tree, KNN (k nearest neighbor) and ANN (a multiple-layer perceptron) in experiment.

4. Experiments

We prepare training data based on a news search engine[17], and validate the effectiveness of our proposed features using classical feature selection measures. Then evaluate our learning model by some methods in Information Retrieval.

4.1. Experiment Setup

4.1.1.Training Data. We give 100 classical queries to a news search engine[17]. There are ambiguous queries, specified ones, and meaningless ones to broader the applicability.

For each query, we pick up top n articles (n=100 in the experiment), then extract label candidates based on these articles, and calculate the value of each feature for each label candidate. At last each label candidate is annotated manually to indicate whether it is a valid label. So a vector as following is an entry in the training set:

x = (distance, MI, likelihood ratio, part of speech, MI_Diff, manual-annotation).

4.1.2.Evaluation Measure. The evaluation of our system adopts classical evaluation measure in Information Retrieval.

Precision is defined as:

$$precision = \frac{|L \cap C|}{|L|}$$

where L is the set of labels, and C is the set of manually tagged correct labels.

As listing all potential labels according to search results manually is laborious and subjective which leads unavailable of all valid labels, we assume the recall label candidates extraction is 100% meaning all potential labels are included in the candidates.

We define recall of our binary classifier as:

$$recall = \frac{|S \cap C|}{|C|}$$

Where S is the set of labels which are selected as valid labels by our classifier, C is the set of manually tagged correct labels.

4.2. Result for Label Candidates Extraction

According to manually tagged correct labels, precision is 75% in this step, and recall is 100% as assumption.

4.3. Feature Examination

We use the methods of information gain and CHI-square to do feature examination for each feature we proposed before. Here are the results:

Table 1: Comparison of Features

	CHI square	Inform gain
Phrase POS	134.98 (2.37)	0.1 (0.002)
Body MI	91.63 (4.64)	0.055 (0.004)
Title MI	0	0
Body LR	0	0
Title LR	31.03 (1.08)	0.026 (0.001)
Distance Mean	29.47 (1.42)	0.02 (0.001)
Distance variance	0	0
Title MI-diff	0	0
Body MI-diff	79.9 (2.0)	0.049 (0.001)

We adopt cross validation to give both mean value and variance of CHI-square and information gain to verify effect of different features.

Phrase POS denotes POS of label candidates. *Body MI* and *Title MI* mean MI values calculated on news body and news title. *Body LR* and *Title LR* indicate likelihood ratios on news body and news title. *Distance Mean* and *Distance variance* denote mean value and variance of distances between query and label candidate. *Title MI-diff* and *Body MI-diff* represent the difference of global and local MI values on title and body.

Observing results in Table 1, we find *MI* measure is a better indicator applied on news body than on news title. On the contrary, *Likelihood ratio* measure works well on news title, but fails on news body. We ascribe these phenomena to the fact that *MI* measure works well on dense data while *Likelihood ratio* measure is adept for sparse data. Obviously, data on news body is much denser than on news title. Distance Variance is a good indicator to discover collocation, but our target is to indicate relevance between query and label candidate, so mean value of distance may be more appropriate.

As results listed in Table 1, five features including Phrase POS, Body MI, Title LR, Distance Mean, Body MI-diff are selected to represent label candidate.

4.3. Learning Result

Using four classical classification methods (including kNN, decision tree, support vector machine, ANN) as our learning model, we get the recall and precision of them as:

IEEE COMPUTER SOCIETY

In order to compare our learning model results with unsupervised method, we use mutual information between query term and label candidate as a criterion for label selection also, and compare its performance with four classifiers, and show the results in figure 2.

Table 2.Comparative Results of Four Classifiers

Classifier	precision	recall
kNN	0.84	0.95
decision tree	0.84	0.94
SVM	0.825	0.99
ANN	0.855	0.94

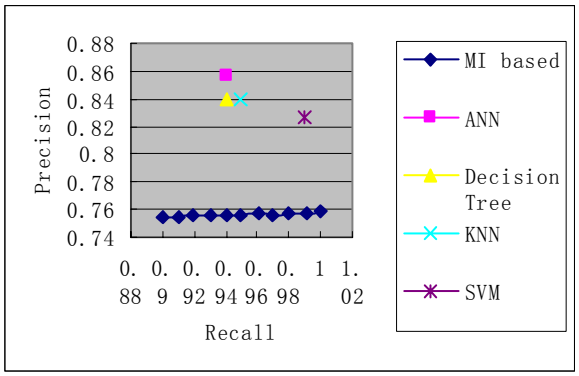

Figure 2

5. Conclusions and Future Work

Our application focuses on label generation for organizing search results from a specified corpus (so not from general search engine), and we select five features to give representation to label candidates, and validate their efficiency of discriminating between useful and meaningless labels. A binary classifier is used as learning model.

We will further study this problem, and try to solve the problems:1) Investigate more features that can describe the label candidates well; 2) The results of our system is a flat structure for browsing, we will try some methods to get a hierarchical structure which will be more convenient for users.

6. References

[1]Stanislaw Osinski, Jerzy Stefanowski and Dawid Weiss, "Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition". In Advances in Soft Computing, Intelligent Information Processing and Web Mining, Proceedings of the International IIS: IIPWM′04 Conference, Zakopane, Poland, 2004, pp. 359-368

[2] D. Cutting, D. Karger, J. Pedersen, and J. Tukey, "Scatter-gather: A cluster-based approach to browsing large document collections," in Proceedings of SIGIR'92, 1992

[3] Hearst, M., and Pedersen.J, "Reexamining the Cluster Hypothesis: scatter/gather on Retrieval Results", Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development Information Retrieval(SIGIR'96), Zurich, 1996

[4] Hua-Jun Zeng and Qi-Cai He et al, "Learning to Cluster Web Search Results". Proceeding of ACM SIGIR'04, UK, 2004

[5] Yitong Wang, Masaru Kitsuregawa, "Use link-based Clustering to Improve web search results", Proceedings of the Second International Conference on Web Information Systems Engineering(WISE'01), 2001, Volume1

[6] Krishna Kummamuru and Rohit Lotilkar et al, "A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results", WWW2004, New York, 2004

[7] Hiroyuki Toda and Ryoji Kataoka, "A Search Result Clustering Method using Informatively Named Entities", processing of WIDM'05, Bremen, 2005

[8] Dunning, Ted, "Accurate Methods for the Statistics of Surprise and Coincidence", Computational Linguistics 19: 61-74, 1993

[9] Lee-Feng Chien, "PAT-Tree-Based Adaptive Keyphrase Extraction for Intelligent Chinese Information Retrieval", In Proceedings of the 20th Annual International ACM/SIGIR conference on Research and Development in Information Retrieval(SIGIR'97), Philadelphia, 1997

[10] Vivisimo clustering engine, http://vivisimo.com, 2006

[11] Christopher D. Manning and Hinrich Schutze, "Foundations of Statistical Natural Language Processing", Massachusetts Institute and Technology, 1999.

[12] Leuski A. and Allan J., "Improving Interactive Retrieval by Combining Ranked List and Clustering", Proceedings of RIAO, College de France, 2000

[13] Leuski A.V. and Croft W.B., "An Evaluation of Techniques for Clustering Search Results", Technical Report IR-76, Department of Computer Science, University of Massachusetts, Amherst, 1996

[14] Zamir O. and Etzioni O. , "Web Document Clustering: A Feasibility Demonstration" , Proceedings of the 19th International ACM SIGIR Conference on Research and Development of Information Retrieval(SIGIR'98), 1998

[15] Smadja Frank, "Retrieving Collocations from Text: Xtract", Computational Linguistics 19:143-177, 1993

[16] Smadja, Frank A., and Kathleen R.McKeown, "Automatically Extracting and Representing Collocations for Language Generation", In ACL 28, pp.252-259, 1990

[17] http://news.baidu.com

[18] Yiming Yang and Jan O. Pederson: "A Comparative Study On Feature Selection in Text Categorization", Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), 1997, pp412-420

IEEE COMPUTER SOCIETY