Generating True Relevance Labels in Chinese Search Engine Using Clickthrough Data

Hengjie Song[1, 2], Chunyan Miao[1] and Zhiqi Shen[3]

[1]Emerging Research Lab, School of Computer Engineering, Nanyang Technological University
{HJSONG, ASCYMIAO}@ntu.edu.sg

[2] Baidu.com, Inc., songhengjie@baidu.com

[3] ICIS, School of Electrical and Electronic Engineering, Nanyang Technological University
ZQSHEN@ntu.edu.sg

Abstract

In current search engines, ranking functions are learned from a large number of labeled <query, URL> pairs in which the labels are assigned by human judges, describing how well the URLs match the different queries. However in commercial search engines, collecting high quality labels is time-consuming and labor-intensive. To tackle this issue, this paper studies how to produce the true relevance labels for <query, URL> pairs using clickthrough data. By analyzing the correlations between query frequency, true relevance labels and users' behaviors, we demonstrate that the users who search the queries with similar frequency have similar search intents and behavioral characteristics. Based on such properties, we propose an efficient discriminative parameter estimation in a multiple instance learning algorithm (MIL) to automatically produce true relevance labels for <query, URL> pairs. Furthermore, we test our approach using a set of real world data extracted from a Chinese commercial search engine. Experimental results not only validate the effectiveness of the proposed approach, but also indicate that our approach is more likely to agree with the aggregation of the multiple judgments when strong disagreements exist in the panel of judges. In the event that the panel of judges is consensus, our approach provides more accurate automatic label results. In contrast with other models, our approach effectively improves the correlation between automatic labels and manual labels.

Introduction

For a given query and the retrieved URLs, search engines order URLs via a ranking function that produces a score for each URL, indicating how well the URL matches the query. By training ranking function with training data and then evaluating its performance with test data, learning to rank plays an important role in commercial search engine. Usually, the typical training data includes the following triples <query, URL, label>, where the label is assigned by human judges (editors), indicating the relevance of a URL to a query from highly relevant to not relevant, e.g., Perfect, Excellent, Good, Fair or Bad. As previous works have shown (Xun et al. 2010; Sheng et al. 2008), the retrieval accuracy of a ranking function depends both on the quality of the training labels and on the number of training examples. Consequently, increasing the number of training examples is the most common method to improve the accuracy of a ranking function.

However, researchers found that the retrieval accuracy of a ranking function stops after the number of training examples reaches a certain threshold (Sheng et al. 2008). When more training examples are not able to further improve the retrieval accuracy, improving the quality of labels is a promising solution: the quality of the training labels heavily influences the quality of a ranking function (Yang et al. 2010; Agrawal et al. 2009).

In commercial search engines, collecting high quality labels is time-consuming, labor-intensive and costly. Since the labels of training data are collected using human judges, label quality depends both on the expertise of editors and on the number of editors. The manually generated labels may contain a personal bias since it is very hard for an editor to capture all the intents of a query, and hence create unreliable labels (Joachims et al. 2002; Bailey et al. 2008; Sheng et al. 2008). To alleviate such errors, a panel of judges are used to obtain multiple judgments for the same

<query, URL> pair. The final label of the pair is then derived by aggregating the multiple judgments. However, commercial search engines require a large number of training data, which requests a relatively larger number of repeated labels for each sample. Accordingly, the high cost makes this approach impractical. In this case, relevance labels are usually conducted by a few, even one judgment. Such kind of relevance labels is prone to contain errors.

Therefore, there is a pressing need to automate the labeling process as much as possible in commercial search engines (Agrawal et al. 2009). Especially for Chinese search engines, as far as we know, few previous works investigate how to automatically produce the true relevance labels for <query, URL> pairs. To tackle this issue, clickthrough data is utilized in this paper to perform automatic labeling process—where the task is to build a classifier to predict whether a URL is 'Perfect', 'Excellent', 'Good', 'Fair' or 'Bad' with respect to a given query.

Unlike previous works, we first analyze the correlations between query frequencies, true relevance labels, and the users' behavior characteristics (e.g., the ratio of turning to the search results in the 2nd page, the ratio of clicking the 1st, 2nd, 3rd or other URLs). Following this analysis, we propose an efficient discriminative parameter estimation by a multiple instance learning algorithm (MIL) to automatically generate the true relevance labels of <query, URL> pairs. In particular, the proposed approach is performed so that: 1) it utilizes the correlations between query frequency, users' behavioral characteristics and true relevance labels; and 2) unlike other methods (Agrawal et al. 2009; Cao et al. 2010), our approach focuses on automatically producing the true relevance labels, rather than the pairwise preferences. By doing so, the proposed approach may be more applicable for several ranking functions which are directly derived from the training data with true relevance labels (Burges et al. 2006; Michael et al. 2008). Furthermore, we test our approach using a set of real world data extracted from a Chinese commercial search engine. Manually labeled data is used as the ground truth to evaluate the precision of the proposed approach. Experimental results not only validate the effectiveness of the proposed approach, but also indicate that our approach is more likely to agree with the aggregation of the multiple judgments when strong disagreements exist in the panel of judges. In the event that the panel of judges is consensus, our approach provides more accurate label accuracy. In contrast with sequential dependency model (SDM) and full dependency model (FDM) (Xun et al. 2010), our approach effectively improve the correlation between automatic label results and manual label results.

Related Work

As a powerful signal about users' preference and latest tendency on search results, the terabytes of users' clickthrough data can be collected at very low cost in a commercial search engine. Clickthrough data contains a large amount of valuable information about users' feedback, which could be considered as complementary information to describe the relevance of URLs with respect to a given query: clicked URLs are most likely relevant to the users' intent, while skipped URLs are most likely not.

Recently, many studies have attempted to automatically generating labels from click-through data (Joachims et al. 2007; Agrawal et al. 2009; Bailey et al. 2008; Cao et al. 2010). For these approaches, a common assumption is that the relative order of the retrieved URLs in terms of the performance obtained with training labels is quite stable even if remarkable disagreement exists among human judgments. Accordingly, these approaches mainly focus on generating the pairwise preferences to train ranking function, instead of predicting the true relevance labels (Xun et al. 2010). When applying these approaches, many contradicting pairwise preference must be reconciled to obtain a consistent labeling.

The work most related to ours (Carterette et al. 2007; Xun et al. 2010) tried to model the relationship between true relevance labels and clickthrough data. Differing from the previous work, the method in (Carterette et al. 2007) revealed the important impact of true relevance labels on ranking functions. The true relevance label of an URL with respect to a given query is then defined as the regression function of probability distribution of the query, the true relevance labels of other retrieved URLs and their respective clickthrough rates.

More recently, the approach proposed in (Xun et al. 2010) proves that errors in training labels can significantly degrade the performance of ranking functions. Furthermore, Xun et al. proposes two new discriminative models, SDM and FDM, to detect and correct the errors in relevance labels using click-through data. As conditionally dependent models, SDM and FDM assume that the relevance label of a search result (e.g., URL, documents) is conditionally dependent on the relevance labels of other search results. The basic assumption means that the search intent of a query is explicit and the contents of the related URLs are comparable. And only thus can the appropriate label be given to a <query, URL> pair after editors compare the retrieved search results.

This assumption seems to be too strong for the queries in which users' intent is ambiguous and very general. For example, given a query 'machine learning', it is very hard for editors to judge what the real intention is, to buy a book, to submit a paper or to browse relevant knowledge. In this case, the URLs content with respect to different search in-

Eq. (12) embeds the model parameters through the likelihood ratios defined in Eqs. (6)-(11). Therefore, discriminative training can be used to estimate the parameters, which is described in the next section.

Estimation of parameters

Assuming the parameter set Λ_{i} at the t-th iteration is $\Lambda_{\mathrm{i}}(\mathrm{t})$, then Λ_{i} at the $(\mathrm{t}+1)$-th iteration $\Lambda_{\mathrm{i}}(\mathrm{t}+1)$ is updated as,

$$\Lambda_{\mathrm{i}}(t+1) = \Lambda_{\mathrm{i}}(t) + \mathrm{r} \cdot \nabla L|_{\Lambda_{\mathrm{i}}(t)} \quad (13)$$

where r is learning rate; $\nabla L|_{\Lambda_{\mathrm{i}}(t)}$ indicates the gradients over all parameters which are described as,

$$\nabla L|_{\Lambda_{\mathrm{i}}(t)} = \alpha\big(\textstyle\sum_{\mathrm{i}=1}^{7}(1-\mathrm{P}(l=+|B_i^+,\Lambda_i(t)))\cdot\nabla P(B_i^+|\Lambda_i(t))|_{\Lambda_i(t)} - \sum_{\mathrm{i}=1}^{7}(1-\mathrm{P}(l=-|B_i^-,\Lambda_i^-(t)))\cdot\nabla P(B_i^-|\Lambda_i(t))|_{\Lambda_i(t)}\big) \quad (14)$$

In Eq. (14), $\nabla P(B_i^c|\Lambda_i(t))|_{\Lambda_i(t)}$ is described as,

$$\nabla P\left(B_i^c|\Lambda_i(t)\right)|_{\Lambda_i(t)} = \sum_{\mathrm{j}=1}^{\mathrm{n_i^c}} \frac{\exp\left(\eta\cdot p\left(x_{ij}^c|\Lambda_i\right)\right)}{\sum_{\mathrm{j}=1}^{\mathrm{n_i^c}}\exp\left(\eta\cdot p\left(x_{ij}^c|\Lambda_i\right)\right)}(\nabla logg^+\left(x_{ij}^c|\Lambda_i(t)\right)|_{\Lambda_i(t)} - \nabla logg^-\left(x_{ij}^c|\Lambda_i(t)\right)|_{\Lambda_i(t)}) \quad (15)$$

In particular, we derive the gradients of positive model with respect to its parameters $(\nabla logg^+(x_{ij}^c|\Lambda_i(t))|_{\Lambda_i(t)})$,

$$\nabla logg^+\left(x_{ij}^c|\Lambda_i(t)\right)|_{w_i^+(t)} = N(x_{ij}^c|w_i^+(t),\mu_i^+(t),\textstyle\sum_i^+(t) \quad (16)$$

$$\nabla logg^+\left(x_{ij}^c|\Lambda_i(t)\right)|_{\mu_i^+(t)} = \beta(t)\cdot\textstyle\sum_i^+(t)^{-1}\cdot(x_{ij}-\mu_i^+(t)) \quad (17)$$

$$\nabla logg^+\left(x_{ij}^c|\Lambda_i(t)\right)|_{\sum_i^+(t)} = -\frac{1}{2}\beta(t)\cdot(\textstyle\sum_i^+(t)^{-1}-\sum_i^+(t)^{-1}\cdot A\cdot\sum_i^+(t)^{-1}) \quad (18)$$

In Eq. (17) and (18),

$$\beta(t) = \frac{w_i^+(t)N(x_{ij}^c|w_i^+(t),\mu_i^+(t),\sum_i^+(t))}{g^+(x_{ij}^c|w_i^+(t),\mu_i^+(t),\sum_i^+(t))} \quad (19)$$

$$A = (x_{ij}^c-\mu_i^+(t))(x_{ij}^c-\mu_i^+(t))^T \quad (20)$$

Similarly, $\nabla logg^-(x_{ij}^c|\Lambda_i(t))|_{\Lambda_i(t)}$ is also computed by the similar formulations as Eqs. (16)-(20)

Experiments

Dataset and Evaluation Metrics

All the experiments are based on the clickthrough data collected from Baidu.com in October 2010. Baidu.com is one of the largest commercial search engines in Chinese Wed environment.

We sampled 4,723 unique queries and 12,3474 unique URLs randomly. Considering the related clickthrough data are absolutely necessary for our approach, we filter out the <query, URL> pairs without click activity (2,3717 <query, URL> pairs) by parsing the one-month click logs. Finally, each refined <query, URL> pair was manually labeled by 3 well-trained editors for relevance. On the average, 21.12 URLs per query are labeled. The ordinal judgments are converted into numeric values by assigning the scores 1-5 to the labels 'Bad'-'Perfect' respectively. If there was a disagreement, a consensus was made by a group discussion. The detailed information about the dataset is summarized in Table 1.

Table 1: Detailed Information about the Dataset

	Data Set	Query Frequency						
		1	**2~8**	**9~64**	**65~512**	**513~4096**	**4097~32768**	**≥32769**
Number (rate%)	99757	16379	15731	21258	16080	13177	10285	6847
Perfect (rate%)	11916	1296(7.92%)	1481(9.42%)	2882(13.56%)	2117(13.17%)	2272(17.24%)	1303(12.67%)	561(8.2%)
Excellent (rate%)	20882	3302(20.17%)	2873(18.27%)	3715(17.48%)	3414(21.24%)	3468(26.32%)	2753(26.77%)	1353(19.76%)
Good (rate%)	22957	3237(19.77%)	3199(20.34%)	4714(22.18%)	4157(25.85%)	3272(24.83%)	2381(23.15%)	1994(29.11%)
Fair (rate%)	23448	4066(24.83%)	4452(28.31%)	5534(26.04%)	3868(24.06%)	1626(12.34%)	2020(19.64%)	1878(27.44%)
Bad (rate%)	20536	4456(27.21%)	3721(23.66%)	4408(20.74%)	2521(15.68%)	2539(19.27%)	1828(17.77%)	1060(15.49%)

We note that the distribution of query frequency in the sample set is different from that in population (discussed in Section 3) slightly. This difference mainly results from filtering out the <query, URL> pairs without click activity. In our experiments, we don't consider the effect imposed by the difference between the sample set and population. Finally, we randomly divide the dataset into training set and test set, and perform 10-fold cross validation.

Due to the lack of publicly available datasets, manually labeled data is used as the ground truth to evaluate the *label accuracy* (Cao et al, 2010) of the proposed approach which is defined as,

$$Acc = \frac{\sum_{i=1}^{5} N_{acc}^{(i)}}{\sum_{i=1}^{5} N^{(i)}} \quad (21)$$

where $N^{(i)}$ is the number of <query, URL> pairs with label i. $N_{acc}^{(i)}$ is the number of correctly predicted relevance labels.

To facilitate the comparison with model SDM and FDM (Jingfang et al. 2010), we also adopted the *correlation* (Carterette et al. 2007) between predicted and actual relevance labels, which is defined as,

$$\mathrm{Corr} = \frac{n\sum_{i=1}^{n}x_iy_i - \sum_{i=1}^{n}x_i\cdot\sum_{i=1}^{n}y_i}{\sqrt{n\sum_{i=1}^{n}x_i{}^2-\left(\sum_{i=1}^{n}x_i\right)^2}\cdot\sqrt{n\sum_{i=1}^{n}y_i{}^2-\left(\sum_{i=1}^{n}y_i\right)^2}} \quad (22)$$

where n is the number of <query, URL> pairs. x_i and y_i are predicted and actual relevance label of the i^{th} <query, URL> pair respectively.

Features

Table 2 lists a part of features that are used to describe <query, URL> pairs in our experiments. These features are categorized into three classes: Query, URLs and Feature Mixture, which capture the characteristics of queries, URLs and the relationships between a query and the associated URL respectively. It is worth mentioning that some statistical variables, which are less common in other models, are used in our experiments, such as skewness of query occurrences (representing the query timeliness), etc.

In our opinion, the broad range of features enables us to capture many aspects of aggregated user behaviors. These features are generated for each <query, URL> pair and used to construct the aforementioned classifier.

Experimental Results

First, we summarize the label accuracy of our approach and compare it with three editors' judgments in Table 3.

Table 3 shows that the individual editor's judgments differ from the consensus of all judges involved. This result is concordant with the previous work proposed in (Yang et al. 2010). But for the dataset extracted from the Chinese search engine, the individual editor's judgments on top ($\text{Interval}_{QF} \geq$ 32,769 times/day) and tail queries (Interval_{QF}=1 time/day) have greater consistency to the consensus of all judges. This difference about consistency mainly results from the fact that the rules are much clearer and easier to be performed for editors when labeling the URLs associated with top or tail queries.

Table 2: Feature List

Query	
QueryWordLength	Number of words in a query
RatioClickedFreq	Ratio of the query click number to its occurrence number
AvgClickPos	Average click position for the query
SkewQuery	Skewness of query occurrences
CVQuery	Coefficient of variation of query occurrences
URLs	
URLFirstEntropy	For a given query, click entropy of the URL as the first choice
URLLastEntropy	For a given query, click entropy of the URL as the last choice
RatioURLDwells	For a given query, ratio of the average dwelling time on a URL to the average dwelling time on all clicked URLs associated with the query
SkewURL	Skewness of URL clicks
CVURL	Coefficient of variation of a URL clicked
Feature Mixture	
ClickEntropy	Entropy of click number of a URL to the click number of all URLs associated with the query
RatioSkewURLQuery	Ratio of skewness of a URL clicks to the skewness of the query clicks

Table 3: Comparison with the manually generated labels

#1, #2 and #3 represent three editors respectively; #P represents the proposed approach

Query Frequency	**Consistency to the consensus**				**Accuracy**			
	#1	**#2**	**#3**	**#P**	**#1**	**#2**	**#3**	**#P**
1	75.4%	77.3%	72.7%	***61.2%***	81.1%	82.8%	77.4%	***68.7%***
2~8	67.5%	63.2%	67.4%	***68.3%***	73.7%	69.7%	68.4%	***71.5%***
9~64	48.8%	48.6%	51.2%	***52.7%***	56.2%	56.4%	58.3%	***60.3%***
65~512	54.2%	56.9%	55.3%	***57.4%***	63.5%	67.2%	67.6%	***68.1%***
513~4096	42.3%	43.8%	47.2%	***61.3%***	58.5%	56.1%	61.4%	***75.7%***
4097~32768	48.5%	51.3%	55.2%	***63.8%***	56.1%	58.7%	63.5%	***68.1%***
≥32769	73.1%	71.4%	74.2%	***75.1%***	80.4%	79.6%	81.1%	***80.7%***

For tail queries, editors' judgments perform better than our approach, both on consistency and accuracy. It is mainly because that the click data of the URLs associated with tail queries is insufficient for the proposed approach to reveal the relationship between click activity and actual relevance labels.

However, for non-tail queries (amounting for 83.7% of the dataset), our approach provides comparable or better consistency and accuracy than human judges. This result not only validates the effectiveness of our approach but also indicates that our approach more likely agrees with the aggregated multiple judgments when strong disagreements exist in the panel of judges. When the panel of judges is consensus, our approach improves the label accuracy.

Furthermore, we calculate the correlations between the predicted labels and the actual labels. Figure 2 shows the correlations with respect to different query frequency intervals. Experiments #1, #2 and #3 respectively perform 1000, 1250 and 1500 iterations with the randomly initialized parameter set Λ_i. Table 4 summarizes the other parameters involved in the training process and compares the performance of SDM, FDM and our approach in terms of the correlation.

In statistics, the closer the correlation is to 1, the stronger relationship between the predicted labels and the actual labels. From this perspective, Table 4 indicates that the proposed approach outperforms the baseline model SDM and FDM.

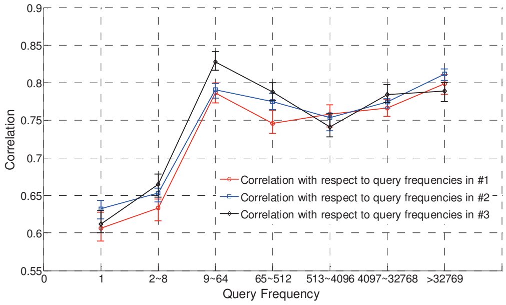

Figure 2: Correlations with different iterations

Table4: Comparison of Correlation

	Parameter	**Correlation**
SDM†	NA	0.69
FDM†	NA	0.75
#1	θ=0.182; η=2; r=0.25;	0.72±0.021
#2		0.75±0.014
#3		0.78±0.023

Summary and Future Work

In this paper, we study how to automatically produce true relevance labels, instead of pairwise preferences for <query, URL> pairs using clickthrough data. By analyzing the correlations between query frequency, user behavioral characteristics and true relevance labels, we demonstrate that the users who search the queries with similar frequency most likely have similar search intents and behavioral characteristics. Based on this property, an effective method to estimate the parameters in a MIL is proposed for automatic labeling process. Experiments on real world data validate the effectiveness of our approach.

The future challenge for this approach is to improve the effectiveness on tail queries, which is a common problem for data mining. In addition, improving the performance of ranking functions, to some extent, requires the automatic label results with high quality. To tackle this issue, it is maybe useful to score automatic labels by the confidence. Such a scoring could be used to determine which automatic labels to use while training ranking functions.

References

Xun, J. F., Chen, C. L., Xu, G., Li, H., and Elbio, A. 2010. Improving quality of training data for learning to rank using click-through data. In *Proceedings of the 3th ACM international conference on Web Search and Data Mining (WSDM'10)*, 171-180. New York, US: ACM.

Sheng, V. S., Provost, F., and Lpeirotis, P. G. 2008. Get another label? Improving data quality and data mining using multiple noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery & Data Mining (SIGKDD'08)*, 614-622. Nevada, US: ACM.

Yang, H., Anton, M., and Krysta, M. S. Collecting high quality overlapping labels at low cost. In *SIGIR'10*, 459-466. Geneva, Switzerland: ACM.

Agrawal, R., Kenthapadi, K., Mishra, H., and Tsaparas, P. 2009. Generating labels from clicks. In *WSDM'09,* 172-181. Barcelona, Spain: ACM.

Joachims, T., Granka, L. A., Pan, B., Hembrooke, H., and Gay, G. 2007. Evaluating the accuracy of implicit feedback from clicks and query reformulation in web search. *ACM Trans. Inf. Syst.*, 25(2):7-33.

Joachims, T., Granka, L. A., Pan, B., Hembrooke, H., and Filip, R., 2002. Optimizing search engines using click-through data. In *SIGKDD'02*, 133-142. Edmonton, Canada: ACM.

Carterette, B., and Jones, R. 2007. Evaluating search engines by modeling the relationship between relevance and clicks. In *Advances in Neural Information Processing Systems (NIPS'07)*, 20:217–224. Whistler: Canada.

Cao, B., Shen, D., Wang, K. S, and Yang. Q. 2010. Click-through log analysis by collaborative ranking. *Association for the Advancement of Artificial Intelligence*. 224-229. Atlanta, Georgia: AAAI Press.

Bailey, P., Craswell, N., Soboroff, I., Thomas, P., Vries, A. P. and Yilmaz, E. 2008. Relevance assessment: are judges exchangeable and does it matter. In *SIGIR'08*, 666-674.: ACM.

Burges, C. J. C., Ragno, R., and Le, Q.V. 2006. Learning to rank with nonsmooth cost functions, In *NIPS'06*, 193-200. Vancouver, Canada.

Michael, J. T., John, G., Stephen, R., and Tom, M. 2008. Softrank: optimizing non-smooth rank metrics. In *WSDM'08*, 77-86. California, US: ACM.

Dietterich, T.G., Lathrop, R. H., and Lozano, P. T. 1997. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31-71.

Ray, S., and Craven, M. 2005, Supervised versus multiple instance learning: an empirical comparison. In *ICML'05*, 697-704.

Viola, P., Chen, X., and Gao, W. 2006. Multiple instance boosting for object detection. In *NIPS' 05*, 18: 1417-1426.

Maron, O., and Perez, T. L. 1998. A framework for multiple instance learning. In *NIPS'98*, 570-576.

Zhang, Q., Goldman, S. A. 2002. EM-DD: an improved multiple-instance learning technique, In *NIPS'02*, 11: 1073-1080.