Community Interest Language Model for Ranking

Xiaozhong Liu
School of Information Studies
Syracuse University, Syracuse NY 13210
xliu12@syr.edu

Miao Chen
School of Information Studies
Syracuse University, Syracuse NY 13210
xliu12@syr.edu

ABSTRACT

Ranking documents in response to users' information needs is a challenging task, due, in part, to the dynamic nature of users' interests with respect to a query or similar queries. We hypothesize that the interests of a given user could be similar to the interests of the broader community of which she is a part at the given time and propose an innovative method that uses social media to characterize and model the interests of the community and use this dynamic characterization to improve future rankings. By generating community interest language model (CILM) for a given query, we use community interest to compute the ranking score of individual documents retrieved by the query. The CILM is based on a continuously updated set of recent (daily or past few hours) user-oriented text data while smoothed by historical community interest. The user-oriented data can be user blogs or user generated textual data.

General Terms

Algorithms, Human Factors, Experimentation

Keywords

Information Retrieval, Ranking, Topic, User, Blog, Community Interest, LDA, Ranking

1. INTRODUCTION

Ranking is a key step in Information Retrieval (IR) systems. Existing ranking algorithms use different approaches to increase performance based on similarity computation, social link analysis, user behavior data, or personalization (user profiling). Ranking is a dynamic problem, namely, user judgments with respect to a query may change dramatically over time. We hypothesize that the ranking score for each retrieved document in the search result should depend on current community interests, for instance, as the following formula shows, (ranked by) the probability that the community (for the target query) interested in document at a given time.

$$Score(doc) = P_{interest}(doc|Community_{query}, time)$$

In this paper, we use "*community interest*" to determine the ranking score, and we compute the interest level of the global (or local) community in a specific document for a given query at a given time. Instead of employing user judgments about what is interesting and what is not, we will use *user oriented (real-time) text data* (such as blog postings, news comments or user selected news text) to represent users' interests. By using a topic-modeling algorithm, *topics* of the real-time community interest in the user text data are identified as probability distributions over words. Each word or topic is then weighted by historical text data from the community. At last, the community interest language model (CILM) is constructed as a language model for each query to represent the current interests of the community. For each document in the search results, we also infer a score (using the precomputed probability topic models) that is proportional to the level of community interest in this specific document given the query. This score is then used for ranking the entire set of retrieved results.

2. COMMUNITY INTEREST RANKING

In the Web 2.0 context, users may generate different kinds of text data, such as blogs, selected news, and comments that reflect their interests. In this paper, we use time sensitive blog data (from blog search engine) to represent users, and we also extract dynamic computational community interest from language model perspective.

For each popular query (from query log), a list of real-time blogs is collected. Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) is used to extract the topics within the collection, and each topic is a probability distribution over words. The next step is to model the community interest based on the extracted topics for ranking.

2.1 Community Interest Language Model

We define community interest (toward each query) as a dynamic probability distribution of each candidate topic over each query, and each number in this distribution represents the current community interest probability of a specific topic given the target query. From language model perspective, the final ranking score could be the (retrieved) document likelihood given the dynamic topic probability distribution for the target query.

$$Score(doc) = P_{interest}(doc|\theta_{query,topic-interest})$$

As mentioned earlier, the candidate topics are extracted from most recent blogs generated by users (in the community). In the preliminary experiment, we find there are three different kinds of topics:

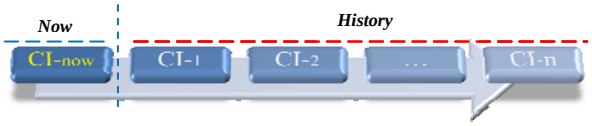
1. *Background topic (stoptopic): the topic covers the very basic background features of the query. Those words could be judged as a query specific stopword list.*
2. *Hot topic: there are two types of hot topics for the community; first, a topic in which the community is continuously and*

increasingly focused, and second, a topic related to breaking news surrounding the query, which is of great interest in the community.

3. *Diminishing topic: the topic is no longer popular for community; and the community's interest is shifting to other topic(s).*

In order to detect each abovementioned topic category and build language model to mirror the current real-world community interest, we will use historical community interest as the smoothing factor.

First, the collected user generated blog data will be separated into different segments based on their generate timestamp (as the following diagram shows, from *CI-now* to *CI-n* ordered by time). Each segment can be viewed as a snapshot of the community interest (about the query) at a given time. Second, the most recent LDA topic model will be used to infer each of the historical CI to get the probability of each CI-k interested in topic-k, *P (topic-z | CI-k).*



In this paper, from language model perspective, retrieved document likelihood given real-time community interest distribution θ_{CI} estimated by real-time blogs will be used to rank the result collection. As the following formula shows:

$$Score(doc) = P(doc \mid \theta_{CI})$$

CI is the real time community interest

As mentioned earlier, two different parts compose community interest (*CI*): *CI-now* (the most current *CI* snapshot) and *CI-history* (historical *CI* snapshots). According to (Zhai & Lafferty, 2004), smoothing could be used to model the noisy word (like IDF effects) and improve the accuracy of the estimated language model. In this research, we face the same problem, as we need to model the real-time community interest by identifying the popular topic distribution, while filtering the noisy information (background and diminishing topics). In order to solve this problem, we used history community interest distribution as the smoothing factor as following:

$$\log P(doc|\theta_{CI}) = \log \frac{P(doc|\theta_{CI-now})}{\alpha \cdot P(doc|\theta_{CI-history})} + n \cdot \log\alpha$$

In the above formula, the probability that *CI* generate the document could be computed by *CI-now* generate document probability divided by *CI-history* generate document probability. So, if a document gets a high ranking score, it should have a high current interest probability score (document topics interest current community) with a low historical interest probability score (document topics not interest historical community).

As we know, *CI-now* is the most current LDA model based topic distribution. So, the problem left is how to model *CI-history*. Intuitively, historical community interest should be a decay function, as the more recent community snapshots should have a larger contribution to the interest model compared with old snapshots. Based on this hypothesis, the following recursive function will be employed to define to *CI-history* with user interest decay parameter β.

$$\theta_{CI-n} = \theta_{n-topic}\ ;\ \theta_{CI-k} = \frac{\theta_{k-topic} + \beta \cdot \theta_{CI-(k-1)}}{1+\beta}$$

In the formulas, θ_{CI-n} is the oldest community interest snapshot, and it is estimated by the inferred topic probability distribution. For any θ_{CI-k}, it is defined (normalized) by θ_{n-1} and CI_n with an interest decay parameter β. Based on this definition, finally, the *CI-history* will be:

$$\theta_{History} = \theta_{CI-1} = \frac{\theta_{1-topic} + \beta \cdot \theta_{CI-2}}{1+\beta}$$

So, CI_1 is composed by $CI_1, CI_2, CI_3 \ldots CI_n$ (all the historical *CI* snapshots) and decay parameters will be $1, \beta^2, \beta^3 \ldots \beta^{n-1}$ ($0 < \beta < 1$).

Finally, the retrieved documents will be ranking by the real time community interest generate probability scores.

From July to December 2008, I implemented the CIV approach with blog training at Yahoo! Search. And in the next academic year, I will be focusing on CILM approach development and experiment with blog training as well as comment tagged news training.

3. EVALUTION

The evaluation of a ranking algorithm is difficult, especially for the real-time ranking task, which cannot employ existing test collections such as TREC. Precision-at-document-n (Anh & Moffat, 2002) is currently a good measure for the web, as most users will be focusing on only the very first page of n results. And Normalized Discount Cumulative Gain (NDCG) (Järvelin & Kekäläinen, 2002) works when user graded relevance data is available.

As a real-time ranking algorithm proposed in this thesis, human judgment will be used for evaluation. The CIV and CILM generated from blog and news will be compared to the popular ranking algorithms and existing web search results. The top ranked documents will be rated as "interesting", "just ok", or "not relevant" by user. For NDCG, each kind of user judgment will be assigned a numeric score, such as 2, 1 and 0. In preliminary evaluation, experiment with Yahoo collection, the CIV algorithm can increase the number of "interesting" ratings by 16.74% while decreasing the "not relevant" rating by 20.59% compared with popular news search engine ranking result (over 9 queries, top 5 search results by 5 users for 5 days). NDCG shows that CIV can significantly (t-test $p<0.05$) increase the ranking performance.

In the next academic year, I will launch a much more comprehensive user evaluation based on Amazon Turk and propose an innovative dynamic ranking evaluation method. While traditional evaluations are based on static relevance judgments, the new ranking evaluation will be based on a user's real-time preferences. As time is introduced as a new parameter in the evaluation matrix, a user's judgment about the same query-document pair could be different depending on the moment the decision is made. The great challenge is to develop an evaluation framework that allows results from the new ranking algorithms to be compared to those obtained using existing ranking algorithms and to popular search engine ranking results.

4. REFERENCES

[1] Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. The Journal of Machine Learning Research, 3.

[2] Liu, X., & Brzeski, V. v. (2009). Computational community interest for ranking. Paper presented at the Conference on Information and Knowledge Managemen

[3] Anh, V. N., & Moffat, A. (2002). Improved retrieval effectiveness through impact transformation. Paper presented at the ACM International Conference Proceeding Series.

[4] Järvelin K. & Kekäläinen J., Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems (TOIS), v.20 n.4, p.422-446, October 2002

[5] Zhai, C., & Lafferty, J. (2004). A Study of Smoothing Methods for Language Models Applied to Information Retrieval. ACM Transactions on Information Systems (TOIS), 22(2), 179-214.