Computational Community Interest and Comments Centric Analysis Ranking

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

Vadim von Brzeski
Yahoo! Inc.
701 First Ave, Sunnyvale CA 94086
vadimv@yahoo-inc.com

ABSTRACT

Ranking is an important subject in information retrieval, and a variety of techniques and algorithms have been developed to rank the retrieved documents and web pages for a given query. However, ranking is also a challenging task, since it is a dynamic problem, namely a user's interest toward each query changes from time to time and it is difficult to accurately extract user interest over time. In this paper, we propose an innovative method to extract and weight real time community interested topic for ranking. By generating *community interest vector* (CIV), we compute the probability score that community interests in specific document or web page in the search results based on daily or past few hours user-oriented data, and use this score for ranking.

Keywords

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

1. INTRODUCTION

Ranking is a key step in Information Retrieval (IR) system. All the ranking algorithms work to find the most "important" documents and show them to the users the top of the search results.

Generally, existing ranking algorithms measure the "importance" of the document in search result in different ways:

a. Similarity or distance score between query and document.

b. Probability that the document generated the query vice-versa.

c. The popularity of the document or web page from external context, such like citation or page link.

In our method, we used "*community interest*" to represent this "importance" score. Instead of employing a large number of users to make judgments, we used *user oriented text data* (such as daily blog postings and user comments) to represent users and compute the probability that community may interest in specific document in the search result for ranking.

2. RELATED WORK

A variety of techniques and algorithms have been developed for ranking retrieved documents and web pages for a given input query. The most important method is content based ranking, which is the similarity or probability of matching between query and target document. For recent web search engine, additional ranking information was provided by web elements, such as the hyperlinks between web pages, anchor text and the popularity of the page, as a result, we need to work for the new algorithm to better meet the special characters of web medium. Basically, there are two existing different approaches for document ranking:

Content based ranking

Content based ranking is a kind of relevance ranking. In the content based ranking, the ranking score of a document with respect to a query is determined by its "distance" to the query vector (Kobayashi and Takeda 2000). The most widely used ranking algorithm is vector space model (Salton and Yang 1973), which calculate the cosine similarity between query and document. In order to reduce the complexity of the vector, LSI method (Deerwester, Dumais et al. 1999) is used to project the high dimensional matrix into a lower one. Similarly, language model (Lafferty and Zhai 2001) and probabilistic model (Jones, Walker et al. 1976) calculate the probability of document generate query and the probability of relevance based on query and document respectively. And these score will be used for ranking.

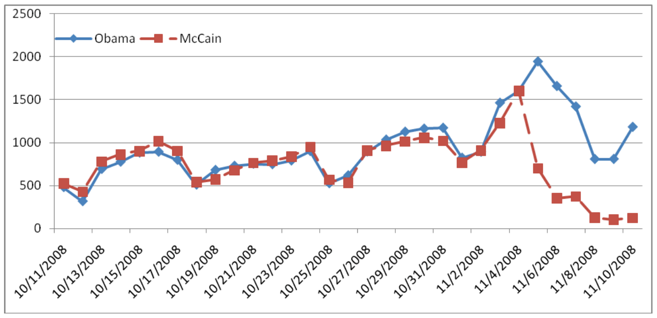
Linkage based ranking

In the WWW environment, the network structure of a hyperlinked network can be a rich source of information about the content of the pages, and provided us effective means to understand it. There are two famous algorithms: PageRank (Page, Brin et al. 1998) and HITS (Hypertext Induced Topic Selection) algorithm (Kleinberg 1999). The basic logics of both algorithms are pretty similar and both of them are based on traditional citation analysis and social network analysis.

Some more recent researches combined these two approaches together and used both content and hyperlinks to rank the search results, for instance (Haveliwala 2003) worked with topic-sensitive PageRank, which created a list of PageRank vectors by using a set of representative topics in order to capture the context of each hyperlink. Similarly, probabilistic model was used by (Richardson and Domingos 2002) to generate PageRank score for each possible query term.

3. COMMUNITY INTEREST RANKING

In Web 2.0 context, user may generate different kind text data, such as blogs or comments to express their opinion. As a result, it is logical to make the hypothesis that user oriented data can represent the overall opinion of the community. A simple example is the 2008 presidential election, as the following diagram shows, the number of blog postings from about "Obama" and "McCain" changed over time, from 2008-10-11 (before election) to 2008-11-10 (after election).



It is obvious that before election day (2008-11-4) the number of posting about two candidates are almost equal, but after the election, because of the election result, the gap between them significant enlarged, which represented the real world. The similar blog research about 2004 election can be found by (Adamic and Glance 2005).

In our model, there are three central questions to answer:

1. How can we accurately extract community interests through user oriented text data based on each protagonist (the main actor in the posting)?
2. If we get multiple interests (a community interest vector) of a protagonist, how can we weigh each user's interest to mirror real world community's requirement?
3. How do we use this computational community interest vector to rank (or re-rank) the documents in the search result?

In this section, we will describe our method and try to answer the abovementioned questions.

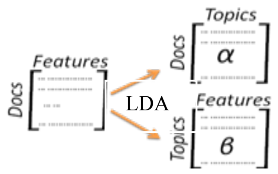
3.1 Community Interest Extraction

If we index user oriented text by protagonist, and each protagonist represents one or several similar hot queries identified from query log, then for each protagonist, we can collect a number of user oriented postings for certain period of time (e.g. today, or past few hours) to make the "Current Protagonist Collection" (CPC). When the number of postings increases, the representability of this collection (of community) also increased.

Our definition of community's interest toward each protagonist is a vector – Community Interest Vector (CIV), and each number in the vector represented a normalized community topic of the target protagonist. This interest vector may change in two different ways:

1. Vector space change – As each dimension in the vector represents an interest point of the target protagonist, the change of vector space demonstrate that either a new dimension (a brand new interest point) appears or an existing interest point fades out (an interesting point diminished).
2. Weight change only – It means community's interest points are stable, but the degree of interest (weight) changes over time. In other word, community's interest shift from one interest point to another.

We hypothesize that the postings in the CPC incorporate a fixed number of *latent topics* and we proceed to extract these topics. A topic in our model is a probability distribution over features (e.g. bag-of-words or entities). There are various techniques to perform this topic modeling step, and we chose an off-the-shelf public domain algorithm called Latent Dirichlet Allocation (LDA), (Blei, Ng et al. 2003). In a nutshell, LDA is similar to *probability Latent Semantic Analysis* (Hofmann 1999) in that it decomposes the posting-by-features matrix into a *posting-by-topics matrix*, α, and a *topics-by-features matrix*, β.



LDA is a generative probabilistic model in the hierarchical Bayesian framework. As the above diagram shows, traditional document indexing systems represent each document as a vector of features. By using LDA, the document-feature matrix can produce two different matrices: α contains the document (posting)-topic probability distributions, i.e. each row represents the probability of topic given the posting *P (topic | posting)*. β contains the topic-feature probability distributions, i.e. each row represents the probability of each feature given the topic *P (feature | topic)*.

An example of LDA result from user oriented text collection as following:

Swimming Topic		Russia War Topic		Gymnast Topic	
Wiki:Michael_Phelps	0.024279	georgia	0.008965	liukin	0.011639
Wiki:Phelps	0.017913	Wiki:Russia	0.007939	Wiki:Nastia_Liukin	0.011465
Swimming	0.011785	spanish	0.00598	Wiki:Gymnast	0.010945
Wiki:Gold_medal	0.011387	russia	0.005513	Wiki:Shawn_Johnson	0.009469
Wiki:Swimming	0.010114	war	0.005047	gymnastics	0.007647
200m	0.009398	Wiki:Georgia_U.S._state	0.003611	johnson	0.00565
Swim	0.009398	states	0.003368	age	0.005477
Record	0.008443	russian	0.002342	nastia	0.004869
Phelps	0.008443	georgian	0.002342	gymnast	0.004782
Meter	0.008045	bush	0.002248	born	0.003828
Freestyle	0.007409	Wiki:George_W._Bush	0.002062	young	0.003567
Wiki:World_record	0.006215	soviet	0.001968	http:	0.003567
Relay	0.005021	ning	0.001968	womens	0.003307
Water	0.004066	fight	0.001782	TVS:nastia	0.003307
LIVE:rebecca_soni	0.003668	did	0.001782	old	0.00322
100m	0.003509	Wiki:Spain	0.001782	father	0.00322
Swimmer	0.003509	iraq	0.001688	chinese_gymnasts	0.00322
Wiki:Medley_swim	0.003509	oil	0.001595	years	0.003133
Wiki:Ryan_Lochte	0.003509	russias	0.001595	Wiki:Uneven_bars	0.00295
Jones	0.003271	south	0.001502	Wiki:Gymnastics	0.002786

Above table shows three sample topics extracted from 2008-08-11 blog posting collection (N=1086 postings, num_of_topic = 30, protagonist = "Olympic"). Each topic is represented by features (bag-of-word + entity + Wikipedia ID, detailed description seen in next section), and the probability of the feature seen given topic *P (feature | topic)*.

Based on this probability distribution, we can use the LDA training model to infer the topic distribution in a new document. The inference result is a document-topic probability distribution vector, each dimension represented the probability that the document belongs to this topic:

$$TV(doc) = (P(topic_1 \mid doc), (topic_2 \mid doc)...(topic_n \mid doc))$$

TV(doc) is the topic vector of document, while *P(topicX|doc)* is the probability score that given document, the probability that the document talked about topic X.

3.2 Feature space

In any information retrieval and text mining system, features are important as the units which represent the document and corpus. However, compared with traditional retrieval system and web search engine, user-generated text's (such like blogs) quality is low, for instance, spell mistake, grammar mistake and spoken language expressions. Further, users tend to use different terms and phrases to express the same thing, which not only increase the dimensions of the feature space, but also misleadingly divided same feature into different ones. In order to solve this problem, we used context entity extraction solution (Brzeski, Irmak et al. 2005) to extract user-centric entities, which is based on user's query log and context.

In order to project different entities into the same semantic feature dimension, we used Wikipedia IDs. Our algorithm first computes the similarity between entity and Wikipedia title and them find a list of similar Wikipedia IDs and then compute the Google distance (Cilibrasi and Vitanyi 2007) between entity and Wikipedia IDs in the context of protagonist.

$$G\_dis(entity, wiki \mid P) = \frac{Max\{\log C(entity, P), \log C(wiki, P)\} - \log C(entity, wiki, P)}{\log M - Min\{\log C(entity_1, P), \log C(entity_2, P)\}}$$

P is the target protagonist; C is the count of results returns from Google general web search. M is the total number of web pages indexed by Google. G_dist is the normalized Google distance between Wikipedia ID and entity, ranged from 0 to 1 (0 means semantic same, 1 means no semantic relatedness).

The final feature space is the combination of Bag-of-word, entity and Wikipedia ID.

3.3 Comment Centric Analysis

Compared with user oriented text data, daily news is a much higher quality corpus. However, extracting community from daily news is risky, for instance:

1. *Obama government now focusing on economy…*
2. *Even with the pressure of economy, Obama have to worry about Iraq…*

The first sentence is about the "economy" topic and the second one talks about "Iraq". However, we may find the distribution distance between these two sentences (K-L divergence) is not large, as the second one mentions "economy" as the context of the central topic, which will negatively influence the performance of the topic extraction algorithm to compute community interest vector as well as weighing each topic in the CIV.

We proposed comment centric analysis to solve this problem. In most cases, a news comment provides reader's opinion based on specific part (or passage) in news posting. And we will use the comments to identify the most important (reader most interested) parts by changing the weight of those words and entities. We call this step *comment centric tagging*. Identification of boundaries of tagging is critically important for this method. There is method of an arbitrary text window in the document such as (Kaszkiel and Zobel 1997), which achieved promising result for information retrieval task. Alternatively, semantic boundaries can be inferred by shift of topic (Hearst and Plaunt 1993; Mittendorf and Schäuble 1994; Knaus, Mittendorf et al. 1998).

After we get the comment tagged news data, we will use the topic extraction algorithm (mentioned in 3.1) to compose CIV for ranking.

3.4 Ranking

When a query equals or similar to the protagonist, we can bias the ranking result by using current community interest vector. For any given retrieved document collection *R (doc₁, doc₂… docₖ)*, based on our training model (α, *topic-feature distribution*), we can inference and get the topic distribution of each document in the search result as:

$$TV(doc_x) = \{P(topic_1 \mid doc_x), P(topic_2 \mid doc_x) ......P(topic_n \mid doc_x)\}$$

Because the topic vector of each search result document is in the same dimensional space as CIV, we can compute the final document interest score by cosine vector similarity:

$$ranking\_score(doc_x) = Sim(CIV, TV(doc_x))$$
$$= \frac{\sum_{i=1}^{n} CIV(topic_i) \cdot TV(doc_x, topic_i)}{\sqrt{\sum_{i=1}^{n} CIV(topic_i)^2} \cdot \sqrt{\sum_{i=1}^{n} TV(doc_x, topic_i)^2}}$$

As CIV represent community's interest over each protagonist, the final ranking score can be viewed as user's oriented text voted rank result. So, the ranking score can represent *P(interst | doc)*, the probability that community interest given the document.

4. EXPERIMENT

In the experiment, we will compute the community interest vector through two different methods:

1. *Generate CIV with user oriented blog data*
2. *Generate CIV by comment centric tagged news data*

And we will use each CIV to inference and rank the retrieved documents from news web search engine. The baseline is the default ranking result from news web search engine.

5. EVALUATION

We will conduct a real world evaluation based on several protagonists over a period of 5 days.

Five real readers will evaluate the performance of three ranking sets: CIV-1 (by blog data) ranking, CIV-2 (by comment centric news data) ranking and the baseline ranking (from web search engine). The evaluation will be a blind evaluation where each day readers will be shown a protagonist and a list of 15 postings for that protagonist (top five ranked document in each set). Users did not know which postings came from which algorithm. Each user will be asked to rate the document as "Interesting and Hot", "Mildly interesting", "Not interesting or not relevant" or "Duplicate".

To evaluate each algorithm, we will compute the "Interesting and Hot" rate (percentage) and "Not interesting or not relevant" rate for each algorithm. If CIV algorithm(s) can increase the "Interesting and Hot" rate while decrease "Not interesting or not relevant" rate over baseline, the algorithm(s) will be judged as an improvement.

6. REFERENCES

[1] Adamic, L. A. and N. Glance (2005). "The political blogosphere and the 2004 U.S. election: divided they blog." Proceedings of the 3rd international workshop on Link discovery.

[2] Blei, D. M., A. Y. Ng, et al. (2003). "Latent Dirichlet Allocation." The Journal of Machine Learning Research 3.

[3] Brzeski, V. v., U. Irmak, et al. (2005). Leveraging context in user-centric entity detection systems. Conference on Information and Knowledge Management.

[4] Cilibrasi, R. L. and P. M. B. Vitanyi (2007). "The Google Similarity Distance." IEEE Transactions on Knowledge and Data Engineering 19(3).

[5] Deerwester, S., S. T. Dumais, et al. (1999). "Indexing by latent semantic analysis." Journal of the American Society for Information Science 41(6).

[6] Haveliwala, T. H. (2003). "Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search." IEEE Transactions on Knowledge and Data Engineering 15(4).

[7] Hearst, M. A. and C. Plaunt (1993). "Subtopic structuring for full-length document access." Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval.

[8] Hofmann, T. (1999). "Probabilistic latent semantic indexing." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval.

[9] Jones, K. S., S. Walker, et al. (1976). "A probabilistic model of information retrieval: development and status."

[10] Kaszkiel, M. and J. Zobel (1997). "Passage Retrieval Revisited." Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval.

[11] Kleinberg, J. M. (1999). "Authoritative sources in a hyperlinked environment." Journal of the ACM 46(5).

[12] Knaus, D., E. Mittendorf, et al. (1998). "Highlighting Relevant Passages for Users of the Interactive SPIDER Retrieval System." 4th Text Retrieval Conference.

[13] Kobayashi, M. and K. Takeda (2000). "Information retrieval on the web." ACM Computing Surveys (CSUR) 32(2).

[14] Lafferty, J. and C. Zhai (2001). "Document language models, query models, and risk minimization for information retrieval." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval.

[15] Mittendorf, E. and P. Schäuble (1994). "Document and passage retrieval based on hidden Markov models." Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval.

[16] Page, L., S. Brin, et al. (1998). "The PageRank Citation Ranking: Bringing Order to the Web."

[17] Richardson, M. and P. Domingos (2002). "The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank."

[18] Salton, G. and C. S. Yang (1973). "On the Specification of Term Values in Automatic Indexing." Journal of Documentation 29(4).