UCSC at Relevance Feedback Track

Lanbo Zhang, Yi Zhang
School of Engineering
UC Santa Cruz
Santa Cruz, CA, USA
{lanbo,yiz}@soe.ucsc.edu

Jadiel de Arma*
Computer Science
Arizona State University
Phoenix, AZ, USA
jadielam@gmail.com

Kai Yu
NEC Laboratories America
Cupertino, CA, USA
kyu@sv.nec-labs.com

ABSTRACT

The relevance feedback track in TREC 2009 focuses on two sub tasks: actively selecting good documents for users to provide relevance feedback and retrieving documents based on user relevance feedback. For the first task, we tried a clustering based method and the Transductive Experimental Design (TED) method proposed by Yu et al. [5]. For clustering based method, we use the K-means algorithm to cluster the top retrieved documents and choose the most representative document of each cluster. The TED method aims to find documents that are hard-to-predict and representative of the unlabeled documents. For the second task, we did query expansion based on a relevance model learned on the relevant documents.

1. INTRODUCTION

We assume good documents for relevance feedback should firstly be relevant, or near relevant, since prior research has shown that relevant documents are more informative than non-relevant ones for commonly used relevance feedback algorithms. Thus a simple idea of selecting documents for relevance feedback is just to choose the top retrieved documents. However, this approach may return similar documents and thus suffers from two problems: diminishing of return and all eggs in one basket. Let's consider one case where the top 5 documents are very similar to each other (or near duplicates). Knowing all of them are relevant won't be more valuable than knowing just one of them is relevant, which is the so-called "diminishing of return" problem. Meanwhile, if one of them is non-relevant, usually all of them are non-relevant, which is the so-called "all eggs in one basket" problem. To avoid these problems, we hope to diversify the selected document set so that we have more chances of finding relevant documents and learning more relevant factors or terms for query expansion. Motivated by the above intuitions, we tried two approaches to find good documents for relevance feedback in our experiments. One approach is based on document clustering, and the other approach is based on Transductive Experimental Design [5].

For the retrieval task, we adapt a language modeling approach based on the Indri (part of lemur) information retrieval toolkit. The language modeling approaches are widely used on several standard information retrieval benchmark data sets, and we want to see how they perform on the new web data ClueWeb09. In our experiments, we go beyond the commonly used "bag-of-words" models by incorporating phrase and text window match in our retrieval model.

2. CHOOSING DOCUMENTS FOR RELEVANCE FEEDBACK BASED ON DOCUMENT CLUSTERING

There are two motivations for us to choose the clustering approach. First, two similar documents sharing most of the common words won't be more informative than only one of them, and a diverse set of relevant documents can help us to find a more diverse set of relevant terms/factors for query expansion. Second, for an ambiguous query, a group of documents representing different possible user intentions will increase the probability that at least one of them is relevant.

We use the K-means algorithm to cluster the top 50 retrieved documents into 5 clusters. A document from each cluster C is chosen as follows:

$$d_{\mathrm{C}} = \arg\max_{d_i \in \mathrm{C}} \sum_{d_j \in \mathrm{C}} similarity(d_i, d_j) \tag{1}$$

Each document is represented as a vector, where each dimension is the TFIDF score of the corresponding term m:

$$d(m) = \frac{(k_1+1)tf_{m,d}}{k_1(1-b+b\frac{L_d}{L_{avg}}) + tf_{m,d}} \log \frac{N - df_m + 0.5}{df_m + 0.5} \tag{2}$$

where $tf_{m,d}$ is the term frequency of term m in document d, L_d is the length of d, L_{avg} is the average document length, N is the total number of documents in the corpus, df_m is the document frequency of term m, and k_1, b are two parameters that need to be set manually. Cosine similarity is used to measure the distance between two documents:

$$similarity(d_i, d_j) = \frac{d_i \cdot d_j}{||d_i||\,||d_j||} \tag{3}$$

3. CHOOSING DOCUMENTS FOR RELEVANCE FEEDBACK BASED ON TRANSDUCTIVE EXPERIMENTAL DESIGN (TED)

Transductive Experimental Design [5] aims to select instances that are hard-to-predict and representative of unlabeled instances. More specifically, TED tries to solve the following optimization problem:

$$\min_{\mathrm{A}, \alpha_i \in \mathrm{R}^K} \sum_{x_i \in \mathrm{X_P}} \{||\, x_i - \mathrm{X}_{\mathrm{A}}^{\mathrm{T}} \alpha_i \,||^2 + \mu \,||\, \alpha_i \,||^2\} \tag{4}$$
$$subject\ to\ |\mathrm{A}| = K, \mathrm{A} \in \mathrm{C}$$

[*] The work is done while the author is a visiting student at UCSC.

Algorithm 1 Sequential Algorithm

```
Input: $X_C, X_P, \mu > 0, K$
Initialization: $\alpha_{i,j} \leftarrow x_i \cdot x_j / ||x_i|| \, ||x_j||, i \in P, j \in C$
repeat:
    $j \leftarrow \arg\max_{j \in C} \sum_{i \in P} \alpha_{i,j}^2 / (\alpha_{j,j} + \mu)$
    $A \leftarrow A \cup \{j\}$
    for $i \in P, i' \in C$ $\alpha_{i,i'} \leftarrow \alpha_{i,i'} - \alpha_{i,j}\alpha_{i',j} / (\alpha_{j,j} + \mu)$
until $|A| = K$
return A
```

where P is the set of all instances, C is the set of candidate instances (i.e. documents), A is the set of selected instances. X_P, X_C and X_A are the the matrix representations of all instances in P, C, A, and K is the number of instances we want to select.

TED aims to find the optimal set of examples A to approximate each instance x_i of X_P. The approximation can be seen as the regularized projection of x_i onto the linear subspace spanned by X_A. Therefore, TED has a geometric interpretation that it tends to find the representative data set X_A that span a linear subspace that retains most of the information of the whole set of instances X_P. Therefore, given a sufficiently large set X_P, TED actually explores the information about the distribution of unlabeled data.

The optimization problem is NP-hard, and we used the sequential algorithm introduced in [6] to find a suboptimal solution (Algorithm 1).

4. RETRIEVAL MODELS

In the second phase, we tried a learning to rank approach (Multiple Addictive Regression Tree) and language models. We use .GOV data set to tune the parameters and evaluate different approaches. Because the settings of the TREC experiments are very different from the standard learning to rank scenario, MART does not perform well on the .GOV data set, thus we didn't submit the corresponding results. Now we briefly discuss the language modeling approach used for generating the submitted run.

The language modeling approach has been successfully used in standard non web retrieval tasks. We are interested to see whether it can do well on the relevance feedback task with web data. Indri [1] (in Lemur toolkit) is a standard open source search engine based on language models. It is designed to handle large datasets, and supports a very flexible query language. Given the limited time, we used Indri as our primary search tool throughout our experiments and implemented our retrieval models based on the Indri query language.

4.1 Baseline Model

In the traditional "bag-of-words" models, the word position information is ignored. However, word position information might be of great value in some cases. For example, for query "the music man", which is a musical name, what the user really wants is information about the musical. Thus a relevant document must contain exactly the phrase "the music man". Besides, intuitively all query words are expected to occur closely in relevant documents, so that their combination makes the document relevant. These motivate us to use both phrase match and text window match in our retrieval models.

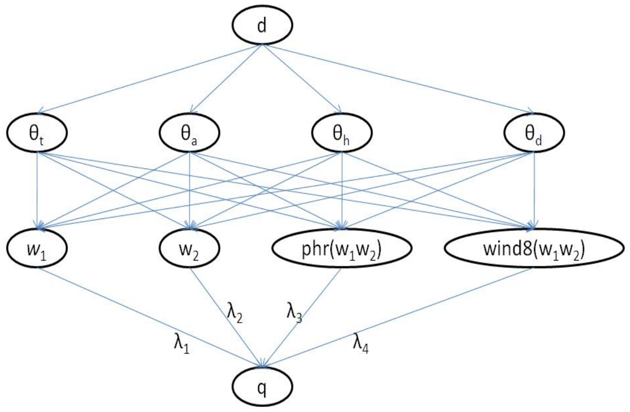

Figure 1: Baseline retrieval method

Our retrieval models are based on the multiple Bernoulli model [3](model B). In this model, a binary random variable is defined for each feature (which could be a word, a phrase, or a text window) to indicate whether this feature f_i is present($f_i = 1$) or absent($f_i = 0$) in a document d. We treat each position of this document as a sample from the model where some features have shown up and other features are absent. For example, in a document with content "A B C", the features present in the first position include the word "A", the phrase "AB" (phr(AB)), the phrase "ABA" (phr(ABA)), the text window of length 2 and containing both "A" and "B" ($wind2$(AB)), the text window of length 3 and containing both "A" and "C" ($wind3$(AC)), and the text window of length 3 and containing "A", "B", and "C" ($wind3$(ABC)). The MLE of $p(f_i = 1|d)$ is calculated based on the frequency of feature f_i occurring in d.

In many cases, the title field of a web page includes very important information about this page. Prior research also found that the anchor texts of web pages are useful for retrieval, possibly because they are usually not written by the page author and thus unlikely to be biased. To take advantages of the prior findings, we tried a mixture model with different document representations in our baseline retrieval. The fields we used are: (t)itle, (a)nchor, (h)eading (text with h1 and h2 tags), and the whole (d)ocument (which includes all the other fields). The mixture model is:

$$\begin{aligned} P(f_i = 1|d) = & \frac{1}{\lambda_t + \lambda_a + \lambda_h + \lambda_d}\{\lambda_t P(f_i = 1|\theta_t) + \lambda_a P(f_i = 1|\theta_a) \\ & + \lambda_h P(f_i = 1|\theta_h) + \lambda_d P(f_i = 1|\theta_d)\} \end{aligned} \tag{5}$$

where f_i could be a word, a phrase, or a text window.

Figure 1 shows the baseline retrieval methods with a query example "w_1 w_2". We assign particular weights to the word match (λ_1 for w_1 and λ_2 for w_2, which are chosen to be equal in our experiment), phrase match (λ_3), and text window match (λ_4) respectively, then documents could be ranked according to:

$$\begin{aligned} S_0(q,d) = & \frac{1}{\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4}(\lambda_1 * logP(w_1 = 1|d) \\ & + \lambda_2 * logP(w_2 = 1|d) \\ & + \lambda_3 * logP(phr(w_1w_2) = 1|d) \\ & + \lambda_4 * logP(wind8(w_1w_2) = 1|d)) \end{aligned} \tag{6}$$

4.2 Relevance Model

A relevance model is estimated based on the relevant documents.

$$P(w = 1|R) = \frac{P(w = 1, R)}{P(R)} = \frac{\sum_{d \in \mathrm{D}^+} P(w = 1|d)P(R|d)P(d)}{P(R)} \tag{7}$$

D^+ is the set of relevant documents of the query, $P(d)$ is assumed to be uniform over all documents, $P(R|d)$ is the probability of relevance of document d, which is given. According to this model, we select the top N words with largest probabilities $P(w = 1|R)$ for query expansion.

The relevance model is shown in figure 2. The weight of each new word is set according to $P(w = 1|R)$. Based on the relevance model, we calculate:

$$S_1(q, d) = \frac{\sum_{w_i \in \mathrm{W}_r} P(w_i = 1|R) log P(w_i = 1|d)}{\sum_{w_i \in \mathrm{W}_r} P(w_i = 1|R)}$$

where W_r is the set of words that are selected for query expansion.

Combining the original query model with the relevance model, we rank the documents based on:

$$S(q_{exp}, d) = (1 - \lambda_{fb})S_0(q, d) + \lambda_{fb} S_1(q, d) \tag{8}$$

where λ_{fb} is the combination weight that reflects how much we believe the relevance model.

5. EXPERIMENTAL SETUP

The large web collection Clueweb09 is used this year. It contains around 1 billion web pages. We chose to work on its subset B, which contains about fifty million documents. We index the documents using the Indri toolkit [2]. A stop word list is used when creating the index. We also chose to create an index on different document fields, including title, heading, anchor, and the full document (all the other fields).

Indri Query Language that considers phrase match and text window match is used in our experiment. [4]. Take the query "obama family tree" as an example, the indri query string would be:

$\#weight($
λ_1 obama
λ_1 family
λ_1 tree
λ_2 #1(obama family)
λ_2 #1(family tree)
λ_2 #1(obama family tree)
λ_3 $\#uw8$(obama family)
λ_3 $\#uw8$(obama tree)
λ_3 $\#uw8$(family tree)
λ_3 $\#uw12$(obama family tree)
)

For the mixture model of different document representations, let's look at the word "obama" as an example. In this case, the indri query string is:

$\#wsum($
λ_t obama.$(title)$ λ_a obama.$(anchor)$
λ_h obama.$(heading)$ λ_d obama.$(document)$
)

In the clustering experiment, we used K-means to cluster the top 50 retrieved documents into 5 clusters. The reason why we chose a small number (50 in our case) as the cut line is that we hope the chosen documents have high probabilities of being relevant. Equation (2) is used to calculate the document vectors. Cosine similarity measure is used to calculate the distance. When calculating the document vectors, k_1 and b are set as 1.2 and 0.75 respectively.

In the TED experiments, we kept the top 1000 retrieved documents, which consist of the set P (see section 3). The candidate set C is the top 50 documents. The sequential algorithm introduced in section 3 is used to select 5 best documents from the top 50 based on the top 1000 documents. Equation (2) is used to calculate the document vectors as well.

In phase 2, all parameters introduced in Section 4 are trained on last year's relevance feedback data set. The Indri query string of our relevance feedback model is created as follows:

$$\#weight((1 - \lambda_{fb})QS_{org} \quad \lambda_{fb} QS_{rel})$$

where QS_{org} is the original query. QS_{rel} is the following query string created based on the relevance model:

$$\#weight(P(w_1|R)\ w_1 \quad P(w_2|R)\ w_2 \quad \cdots \quad P(w_N|R)\ w_N)$$

6. EXPERIMENTAL RESULTS

6.1 Which documents are good for relevance feedback

We submitted two runs in phase 1: UCSC.1 (generated by TED) and UCSC.2 (generated by clustering). Based on the evaluation metric[1] used for phase 1, UCSC.1 is ranked higher than UCSC.2, which means that the TED method performs better than the clustering method in general. Table 1 compares the document sets for UCSC.1 and UCSC.2. We notice that the TED method returns more relevant documents (126) than the clustering method (116). This is probably why UCSC.1 performs better than UCSC.2. However, UCSC.2 has fewer topics with zero relevant documents (9) than UCSC.1 (12). This is consistent with what we expected: for ambiguous queries, clustering method tends to choose a diversified set of documents, so it is more likely that some relevant documents are covered.

Table 1: Clustering v.s. TED

doc set	method	# of rel docs returned	# of topics with no rel doc
UCSC.2	Clustering	116	9
UCSC.1	TED	126	12

[1] See track overview paper for details

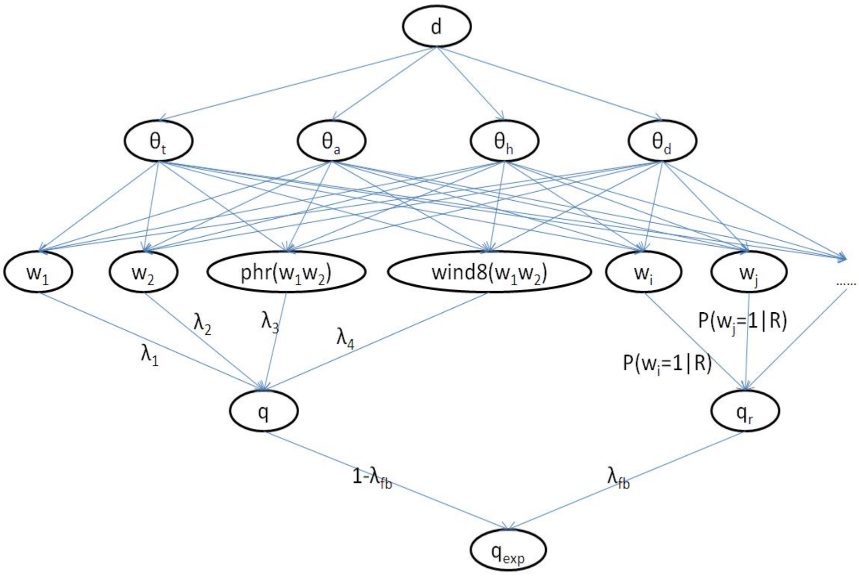

Figure 2: Relevance feedback method. $w_i, w_j...$ are words selected for query expansion

6.2 Performance of relevance feedback algorithm

Our submissions in phase 2 are ranked the 2nd among all participants. This indicates the effectiveness of our relevance feedback algorithm. Also, our RF algorithm performs similarly well on each of the eight document sets, which shows the robustness of our algorithm.

The topic-level performances of our RF algorithm are shown in table 2. All the results are averaged over 8 document sets. The average retrieval performance (stAP) over all topics is improved by 34.9% over the baseline case where no RF is used.

We also notice that relevance feedback may hurt some topics. Among the 50 topics, the retrieval performances get worse on 13 ones (which is more than 1/4 of all topics) when using relevance feedback. Table 3 shows these topics. Further failure analysis shows that relevance feedback actually exacerbates the focus on some aspects of these topics and neglects some other key aspects.

7. CONCLUSIONS

We tried two methods in finding good documents for relevance feedback. We found that the TED method performs better than the clustering method mainly because it returns more relevant documents. Our baseline retrieval model goes beyond the commonly used "bag-of-words" approach, and our relevance feedback model improves retrieval performance by 35% over a baseline without RF.

8. ACKNOWLEDGEMENT

This research was funded by National Science Foundation IIS-0713111, AFOSR/AFRL and ISSDM at University of California Santa Cruz. Any opinions, findings, conclusions or recommendations expressed in this paper are the authors', and do not necessarily reflect those of the sponsors.

9. REFERENCES

[1] Indri: Language modeling meets inference networks. http://sifter.org/ simon/journal/20061211.html.

[2] D. Metzler and W. Croft. Combining the language model and inference network approaches to retrieval. In *Information Processing and Management Special Issue on Bayesian Networks and Information Retrieval*, 2004.

[3] D. Metzler, V. Lavrenko, and W. B. Croft. Formal multiple-bernoulli models for language modeling. In *Proceedings of ACM SIGIR 2004*, 2004.

[4] S. T. T. H. Metzler, D. and W. Croft. Indri at trec 2004: Terabyte track. In *Online Proceedings of 2004 Text REtrieval Conference (TREC 2004)*, 2004.

[5] K. Yu, J. Bi, and V. Tresp. Active learning via transductive experimental design. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, 2006.

[6] K. Yu, S. Zhu, W. Xu, and Y. Gong. Non-greedy active learning for text categorization using convex transductive experimental design. In *Proceedings of the 31st Annual International ACM SIGIR Conference (SIGIR 08)*, 2008.

Table 2: Performances of relevance feedback on each topic

topic id	stAP of baseline	stAP with RF	improve	average # of rel docs
overall	0.1718	0.2318	34.9%	1.97
rf09-1	0.4021	0.6860	70.6%	1.9
rf09-2	0.3383	0.2670	-21.1%	3.4
rf09-3	0.0042	0.0124	194.6%	2.0
rf09-4	0.0465	0.1344	189.1%	0.8
rf09-5	0.0878	0.1049	19.4%	0.5
rf09-6	0.0872	0.3701	324.3%	0.9
rf09-7	0.0369	0.0262	-29.0%	3.5
rf09-8	0.0024	0.0319	1226.5%	0.6
rf09-9	0.0365	0.0292	-20.1%	3.4
rf09-10	0.0768	0.3783	392.9%	1.6
rf09-11	0.1303	0.1752	34.4%	3.3
rf09-12	0.2155	0.1961	-9.0%	2.9
rf09-13	0.0008	0.0006	-21.0%	0.3
rf09-14	0.0101	0.0265	162.1%	1.3
rf09-15	0.2886	0.3559	23.3%	4.0
rf09-16	0.3092	0.4999	61.7%	2.8
rf09-17	0.0972	0.0952	-2.0%	1.1
rf09-18	0.1449	0.2234	54.2%	2.0
rf09-19	0.0016	0.0019	17.7%	0.0
rf09-21	0.2960	0.3878	31.0%	3.6
rf09-22	0.4201	0.4864	15.8%	4.4
rf09-23	0.0280	0.0306	9.4%	0.9
rf09-24	0.0213	0.0520	144.6%	2.1
rf09-25	0.2458	0.4280	74.1%	1.4
rf09-26	0.1858	0.2061	10.9%	2.5
rf09-27	0.2033	0.2166	6.5%	1.8
rf09-28	0.5131	0.4535	-11.6%	3.0
rf09-29	0.0483	0.0532	10.1%	0.1
rf09-30	0.2775	0.2478	-10.7%	3.4
rf09-31	0.1331	0.4079	206.6%	4.3
rf09-32	0.0125	0.0078	-37.9%	2.9
rf09-33	0.4390	0.3959	-9.8%	3.4
rf09-34	0.0274	0.0736	168.5%	0.6
rf09-35	0.3079	0.3423	11.2%	4.0
rf09-36	0.0789	0.1482	88.0%	1.3
rf09-37	0.0516	0.0566	9.6%	0.3
rf09-38	0.1793	0.1029	-42.6%	2.3
rf09-39	0.1436	0.2709	88.6%	2.3
rf09-40	0.1519	0.2286	50.5%	1.5
rf09-41	0.2137	0.2251	5.3%	2.8
rf09-42	0.0504	0.1396	177.0%	0.3
rf09-43	0.2324	0.4128	77.6%	0.9
rf09-44	0.0123	0.0592	381.4%	0.9
rf09-45	0.1854	0.1880	1.4%	2.8
rf09-46	0.6565	0.6366	-3.0%	3.4
rf09-47	0.4392	0.6092	38.7%	2.6
rf09-48	0.1618	0.2698	66.7%	1.3
rf09-49	0.2591	0.4948	90.9%	1.0
rf09-50	0.1258	0.1092	-13.2%	1.0

Table 3: Topic examples where relevance feedback (RF) hurts

id	topic	stAP without RF	stAP improvement with RF
rf09-38	dogs for adoption	0.1793	-42.6%
rf09-32	website design hosting	0.0125	-37.9%
rf09-7	air travel information	0.0369	-29.0%
rf09-2	french lick resort and casino	0.3383	-21.1%
rf09-13	map	0.0008	-21.0%
rf09-9	used car parts	0.0365	-20.1%
rf09-50	dog heat	0.1258	-13.2%
rf09-28	inuyasha	0.5131	-11.6%
rf09-30	diabetes education	0.2775	-10.7%
rf09-33	elliptical trainer	0.4390	-9.8%
rf09-12	djs	0.2155	-9.0%
rf09-46	alexian brothers hospital	0.6565	-3.0%
rf09-17	poker tournaments	0.0972	-2.0%